

 **software**^{AG}

Alfabet-Schnittstelle für RESTful Web-Services

Alfabet-Referenzhandbuch

Dokumentationsversion Alfabet 10.13.1

Urheberrechtlich geschützt © 2013 - 22 Software AG, Darmstadt, Deutschland und/oder Software AG USA Inc., Reston VA, USA und/oder ihre Tochtergesellschaften und/oder ihre Lizenzgeber.



Der Name Software AG und die Namen der Software AG Produkte sind Marken der Software AG und/oder Software AG USA Inc., einer ihrer Tochtergesellschaften oder ihrer Lizenzgeber. Namen anderer Gesellschaften oder Produkte können Marken ihrer jeweiligen Schutzrechtsinhaber sein. Genaue Informationen über die geschützten Marken und Patente der Software AG und ihrer Tochtergesellschaften sind veröffentlicht unter <http://softwareag.com/licenses>.

Die Nutzung dieser Software unterliegt den Lizenzbedingungen der Software AG. Diese Bedingungen sind Bestandteil der Produktdokumentation und befinden sich unter <http://softwareag.com/licenses> und/oder im Wurzelverzeichnis des lizenzierten Produkts.

Diese Software kann Teile von Software-Produkten Dritter enthalten. Urheberrechtshinweise, Lizenzbestimmungen sowie zusätzliche Rechte und Einschränkungen dieser Drittprodukte können dem Abschnitt "License Texts, Copyright Notices and Disclaimers of Third Party Products" entnommen werden. Diese Dokumente enthalten den von den betreffenden Lizenzgebern oder den Lizenzen wörtlich vorgegebenen Wortlaut und werden daher in der jeweiligen Ursprungssprache wiedergegeben. Für einzelne, spezifische Lizenzbeschränkungen von Drittprodukten siehe PART E der Legal Notices abrufbar unter dem Abschnitt „License Terms and Conditions for Use of Software AG Products / Copyrights and Trademark Notices of Software AG Products“. Diese Dokumente sind Teil der Produktdokumentation, die unter <http://softwareag.com/licenses> oder im Verzeichnis der lizenzierten Produkte zu finden ist.

Die Produkte der Software AG stellen Funktionalität zur Verfügung, die für die Verarbeitung persönlicher Daten entsprechend der EU-Datenschutz-Grundverordnung (DSGVO) genutzt werden kann. Die Beschreibungen zur Nutzung dieser Funktionalität finden Sie in der Administrationsdokumentation des jeweiligen Produkts.

Konventionen für die Dokumentation

Konvention	Bedeutung
Fett	<p>Wird für alle Elemente verwendet, die auf der Benutzeroberfläche dargestellt werden, wie zum Beispiel Menüelemente, Schaltflächen, Registerkarten, Dialogfelder, Titel von Ansichtsseiten und Kommandos.</p> <p>Beispiel: Klicken Sie nach Beenden des Setups auf Fertigstellen.</p>
<i>Kursiv</i>	<p>Wird für Hervorhebungen und Verweise auf Dokumententitel und Kapitelüberschriften verwendet. Wird im Code für Variablen verwendet</p> <p>Beispiel: Informationen hierzu finden Sie im Referenzhandbuch <i>Administration</i>.</p> <p>Beispiel: <code><XmlElement XmlAttribute="Anwendername"/></code></p>
Anführungszeichen oben	<p>Kennzeichnet einzugebende Werte und feststehende Namen im Text.</p> <p>Beispiel: Wenn der Objektstatus "Aktiv" ist, dann...</p>
Begriffe komplett in Großbuchstaben	<p>Tastaturtasten</p> <p>Beispiel: STRG+UMSCHALT</p>
Datei > Öffnen	<p>Wird für Menüaktionen verwendet, die vom Anwender durchzuführen sind.</p> <p>Beispiel: Um die Applikation zu schließen, wählen Sie Datei > Beenden</p>
< >	<p>Steht für Variablen, die vom Anwender eingegeben werden.</p> <p>Beispiel: Erzeugen Sie einen neuen Anwender und geben Sie <Anwendername> ein. (Ersetzen Sie den Begriff inklusive Klammern mit dem jeweiligen aktuellen Wert.)</p>
	<p>Dies ist ein Hinweis, der Zusatzinformationen gibt.</p>
	<p>Dies ist ein Hinweis, der Prozessinformationen gibt.</p>
	<p>Dies ist ein Beispiel.</p>
	<p>Dies ist eine Warnung.</p>

Inhaltsverzeichnis

Kapitel 1:	Einleitung	6
Kapitel 2:	Erforderliche Lizenzen	8
Kapitel 3:	Aktivieren des Alfabet RESTful API auf Serverseite	9
	Aktivieren Sie das Alfabet RESTful API in den web.config-Dateien der Alfabet-Webapplikation.	9
	Konfigurieren des Web-Servers, auf dem die Alfabet-Webapplikation gehostet wird, zur Aktivierung der Alfabet RESTful API	11
	Deaktivieren Sie das WebDAV-Modul der Internetinformationsdienste	13
	Einstellen der erforderlichen Autorisierung für den api-Ordner	13
	Konfigurieren des Server-Alias der Alfabet-Webapplikation zum Aktivieren der REST-API-Aufrufe	14
Kapitel 4:	Autorisierung	17
	Erforderliche Konfiguration auf Serverseite	18
	Generieren eines REST-API-Passworts für einen Anwender	18
	Aktivieren von Berichten, ADIF-Schemata und Workflow-Vorlagen, die über RESTful-Serviceaufrufe ausgeführt werden sollen	21
	Aktivieren des Zugriffs auf Ordner in der internen Dokumentenauswahl	22
	Konfigurieren nach Objektklassenberechtigungen für das Lesen oder Beschreiben der Alfabet-Datenbank	23
	Zugriff über Mandanten steuern	26
	Zugriff pro Objekt über Zugriffsberechtigungskonzepte steuern	28
	Erforderliche Implementierung auf Clientseite	29
	Übersicht über für jeden Endpunkt erforderliche Zugriffsberechtigungen	30
Kapitel 5:	Konfigurieren der Handhabung von Datums-, Uhrzeit- und Zahlenformaten für API-Aufrufe	35
	Konfigurieren einer neuen API-Landeseinstellung	35
	Eine API-Landeseinstellung in einem Serviceaufruf verwenden	36
Kapitel 6:	Serviceaufrufe und Rückgabewerte	38
	Exportieren von Informationen für das gesamte Alfabet -Klassenmodell, einschließlich Aufzählungen und Landeseinstellungen	41
	Exportieren von Informationen zu allen oder mehreren Klassen des Alfabet -Klassenmodells	46
	Exportieren von Informationen zu allen oder mehreren Aufzählungen im Alfabet -Klassenmodell	50
	Exportieren von Informationen zu in der Alfabet-Datenbank gespeicherten Objektdaten	53
	Exportieren von Daten zu Objekten ohne definierte REFSTR-Werte	54
	Daten über Objekte einer definierten Objektklasse exportieren, die einer Filterdefinition entsprechen	58
	Exportieren von Informationen zu Objekten, die von einem konfigurierten Bericht gefunden werden	65
	Löschen von Objekten aus der Alfabet-Datenbank	81
	Erstellen und Aktualisieren von Objektdaten in der Alfabet-Datenbank	83
	Erzeugen eines neuen Objekts in der Alfabet-Datenbank	86
	Ändern der Eigenschaften eines vorhandenen Objekts in der Alfabet-Datenbank	89
	Erstellen oder Aktualisieren einer Beziehung zwischen Objekten in der Alfabet-Datenbank	93
	Archivieren von Objekten aus der Alfabet-Datenbank	94
	Erneutes Generieren des Passworts eines Alfabet -Anwenders	97
	Anonymisieren von Daten für bestimmte Anwender	100
	Starten eines Workflows über einen RESTful-Serviceaufruf	101

ADIF-Import über einen RESTful-Serviceaufruf starten	102
Auslösen des ADIF-Imports aus einer externen Datenbank oder einem Dokument in der Alfabet-Datenbank	104
Auslösen des ADIF-Imports aus einem Dateistrom im Serviceaufruf	107
Überprüfen des Ergebnisstatus der ADIF-Ausführung	109
Herunterladen der Protokolldatei zur ADIF-Ausführung	111
ADIF-Export über einen RESTful-Serviceaufruf starten	112
Voraussetzungen für das Ausführen eines ADIF-Exports	112
Synchrone Ausführung eines ADIF-Exports	113
Asynchrone Ausführung eines ADIF-Exports	116
Exportieren von Informationen über den Inhalt der internen Dokumentenauswahl	123
Herunterladen von Dokumenten aus der internen Dokumentenauswahl	125
Hochladen von Dokumenten in die interne Dokumentenauswahl	126
Es wird geprüft, ob die Alfabet-Komponenten ausgeführt werden	128
Aktualisierung des Metamodells	129
Kapitel 7: Zugreifen auf die Alfabet -Benutzeroberfläche von externen Anwendungen	132
Kapitel 8: Testen der Alfabet RESTful API	133
Testen der Alfabet RESTful API	133
Konfigurationen, die für die Verwendung eines Swagger-Editors zum Testen erforderlich sind	133
Kapitel 9: Überprüfen des Erfolgs von Serviceaufrufen an die Alfabet-RESTful-API	135
Index	136

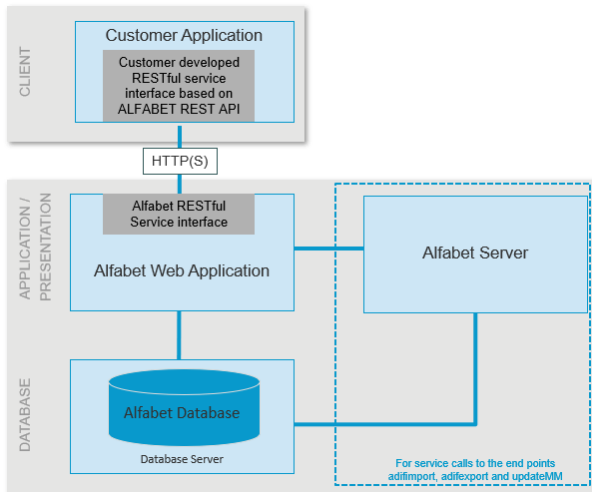
Kapitel 1: Einleitung

Ein RESTful API steht für die Alfabet -Anwendung zur Verfügung. Es ermöglicht einen einfachen Zugriff auf die Inhalte der Alfabet-Datenbank. Das API ist als eine Webdienstarchitektur entworfen worden, die auf dem Softwarearchitekturtyp "Representational State Transfer (REST)" basiert. Sitzungscookie werden nicht verwendet.

Die API kann für Folgendes verwendet werden:

- zum Abrufen von Informationen über das Objektklassenmodell,
- zum Abrufen von Informationen über Objekte, die in der Alfabet-Datenbank gespeichert sind,
- zum Erstellen, Aktualisieren und Löschen von Objekten und Beziehungen in der Alfabet-Datenbank,
- zum Archivieren von Objekten in der Alfabet-Datenbank,
- um die Ausführung von ADIF-Jobs, die auf einem ADIF-Schema in der Alfabet-Datenbank basieren, auszulösen,
- um den Start eines Workflows über die Alfabet-Webapplikation auszulösen, der auf einer Workflow-Vorlage in der Alfabet-Datenbank basiert,
- zum Wiederherstellen von Anwenderpasswörtern,
- zum Anonymisieren von Daten für bestimmte Anwender,
- zum Herunterladen von Dokumenten aus der **Internen Dokumentenauswahl**, zum Hochladen von Dokumenten in die **Interne Dokumentenauswahl** und zum Generieren einer Liste mit Dokumenten in der **Internen Dokumentenauswahl**,
- zum Aktualisieren der Metamodellkonfiguration der Alfabet-Datenbank mit der in einer AMM-Datei gespeicherten Metamodellkonfiguration,
- zum Überprüfen der Verfügbarkeit der Alfabet -Komponenten.

In diesem Dokument werden die verfügbaren Datenendpunkte beschrieben, einschließlich der erforderlichen Aufrufe und Rückgabewerte. Mit diesen Informationen können Kunden Schnittstellen entwickeln, über die sie mittels einer HTTP-Anfrage an die Alfabet RESTful API einer laufenden Alfabet-Webapplikation auf die Alfabet-Datenbank zugreifen können. Die Antworten werden im JSON-Format gesendet.



Kapitel 2: Erforderliche Lizenzen

Für die Verwendung des Alfabet RESTful API muss eine Lizenz für das Alfabet Data Integration Framework (ADIF) aktiv sein.

Kapitel 3: Aktivieren des Alfabet RESTful API auf Serverseite

Service-Aufrufe an das Alfabet RESTful API der Alfabet-Webapplikation werden von der Alfabet-Webapplikation nur dann verarbeitet, wenn eine aktive Lizenz für das Alfabet Data Integration Framework (ADIF) verfügbar ist und auf Serverseite die folgenden Konfigurationen vorgenommen werden:

- [Aktivieren Sie das Alfabet RESTful API in den web.config-Dateien der Alfabet-Webapplikation.](#)
- [Konfigurieren des Web-Servers, auf dem die Alfabet-Webapplikation gehostet wird, zur Aktivierung der Alfabet RESTful API](#)
- [Deaktivieren Sie das WebDAV-Modul der Internetinformationsdienste](#)
- [Einstellen der erforderlichen Autorisierung für den api-Ordner](#)
- [Konfigurieren des Server-Alias der Alfabet-Webapplikation zum Aktivieren der REST-API-Aufrufe](#)

Aktivieren Sie das Alfabet RESTful API in den web.config-Dateien der Alfabet-Webapplikation.

Die web.config-Konfigurationsdatei befindet sich im Alfabet-Webapplikation -Verzeichnis des Installationsverzeichnisses für die Alfabet-Komponenten. Die web.config-Datei kann mithilfe eines Standard-Texteditors bearbeitet werden.

Die erforderlichen Einstellungen für das Alfabet RESTful API müssen dem handlers-Element der web.config-Datei der Alfabet-Webapplikation hinzugefügt werden. Das Handler-Element muss neben den untergeordneten Elementen, die bereits für andere Prozesse enthalten sind, die folgenden untergeordneten Elemente umfassen:

```
<remove name="ExtensionlessUrlHandler-Integrated-4.0" />
<add name="AlfaRest1" type="AlfabetWeb5.api.v1.AlfaRestService, AlfabetWeb5"
verb="*" path="api/v1" />
<add name="ExtensionlessUrlHandler-Integrated-4.0" path="*."
verb="GET,HEAD,POST,DEBUG,PUT,DELETE"
type="System.Web.Handlers.TransferRequestHandler" resourceType="Unspecified"
requireAccess="Script" precondition="integratedMode, runtimeVersionv4.0"
responseBufferLimit="0" />
```

Zusätzlich muss der folgende Code in der web.config-Datei verfügbar sein, um die korrekte Version der erforderlichen Drittanbieterkomponente Newtonsoft.Json zu integrieren:

```
<configuration>
...
<runtime>
  <assemblyBinding xmlns="urn:schemas-microsoft-com:asm.v1">
    <dependentAssembly>
      <assemblyIdentity name="Newtonsoft.Json"
publicKeyToken="30ad4fe6b2a6aeed" culture="neutral" />
      <bindingRedirect oldVersion="0.0.0.0-12.0.0.0"
newVersion="12.0.0.0" />
    
```

```

</dependentAssembly>
<dependentAssembly>
  <assemblyIdentity name="System.Net.Http.Formatting"
    publicKeyToken="31bf3856ad364e35" culture="neutral" />
  <bindingRedirect oldVersion="0.0.0.0-5.2.7.0"
    newVersion="5.2.7.0" />
</dependentAssembly>
</assemblyBinding>
...
</runtime>
</configuration>

```




Aufgrund von Änderungen an der Version der eingebetteten Komponente unterliegt dieser Abschnitt möglicherweise Änderungen während eines Upgrades von einer Alfabet -Version auf eine andere. Es wird empfohlen, die im Release enthaltene `web.config`-Beispieldatei zu verwenden und an die aktuelle Umgebung anzupassen, um sicherzustellen, dass der oben genannte Code und alle anderen erforderlichen Einstellungen in der `web.config`-Datei den Anforderungen des jeweiligen Release entsprechen.



Wenn SAML oder andere Authentifizierungsmechanismen für einmaliges Anmelden für Alfabet verwendet werden, können diese Einstellungen die für die Authentifizierungsmethode erforderlichen Einstellungen beeinträchtigen. Bitte wenden Sie sich bei Problemen mit den REST-API-Einstellungen in der `web.config`-Datei an den Software AG Support.

Optional kann die Sicherheit der Datenübertragung durch die Angabe des JSON Web Token (JWT) für das Senden von JSON-Objekten über die Alfabet RESTful Service-API erweitert werden. Standardmäßig ist das JWT hartcodiert und daher für alle Alfabet -Installationen gleich. Wenn das JWT für eine Installation geändert werden soll, muss ein einzelnes JWT mit einer Mindestlänge von 128 Bit Base64-verschlüsselt werden, und die verschlüsselte Version muss in die Datei `alfabet.config` der Alfabet-Webapplikation eingegeben werden:

- 1) Öffnen Sie den Alfabet Administrator.
- 2) Klicken Sie im Explorer auf den Knoten **Alfabet-Aliase**. Ein Arbeitsbereich mit einer Symbolleiste wird geöffnet.
- 3) Klicken Sie in der Symbolleiste auf **Tools > alfabet.config konfigurieren**. Ein Editor wird geöffnet.
- 4) Klicken Sie auf die Schaltfläche **Durchsuchen**  rechts neben dem Feld **Webordner**, und wählen Sie im Verzeichnisbrowser das Hauptverzeichnis der Alfabet-Webapplikation aus. Die Datei `alfabet.config` im Unterverzeichnis **config** des ausgewählten Verzeichnisses wird im Editor geöffnet.
- 5) Fügen Sie folgenden Code dem XML-Element `alfaSection` als untergeordnetes Element hinzu:

```
<add key="ApiJwtBase64Key" value="{Base 64 Encrypted key}"/>
```

- 6) Klicken Sie auf **Speichern**. Die Änderung wird gespeichert und der Editor geschlossen.

Wenn Ihr RESTful-Client eine hohe Anzahl von Serviceaufrufen pro Sekunde an die RESTful-Service-API von Alfabet sendet, muss möglicherweise die maximal zulässige Anzahl von Anfragen pro Sekunde erhöht werden. Die Verarbeitung eingehender RESTful Serviceaufrufe ist standardmäßig auf 300 Anfragen pro Sekunde beschränkt. Der Grenzwert kann in der Datei „`alfabet.config`“ von Alfabet-Webapplikation geändert werden:

- 1) Öffnen Sie den Alfabet Administrator.
- 2) Klicken Sie im Explorer auf den Knoten **Alfabet-Aliase**. Ein Arbeitsbereich mit einer Symbolleiste wird geöffnet.
- 3) Klicken Sie in der Symbolleiste auf **Tools > alfabet.config konfigurieren**. Ein Editor wird geöffnet.
- 4) Klicken Sie auf die Schaltfläche **Durchsuchen**  rechts neben dem Feld **Webordner**, und wählen Sie im Verzeichnisbrowser das Hauptverzeichnis von Alfabet-Webapplikation aus. Die Datei `alfabet.config` im Unterverzeichnis **config** des ausgewählten Verzeichnisses wird im Editor geöffnet.
- 5) Suchen Sie nach dem XML-Element `add`, für das das XML-Attribut `key` auf `max_api_requests_per_second` festgelegt ist, und ändern Sie das XML-Attribut `value` auf die erforderliche Anzahl von Anfragen pro Sekunde. Die Standardeinstellung lautet „300“.

```
<add key="max_api_requests_per_second" value="300"/>
```
- 6) Klicken Sie auf **Speichern**. Die Änderung wird gespeichert und der Editor geschlossen.



Bitte beachten Sie, dass die Anfragebegrenzungen, die außerhalb der Alfabet-Webapplikation für die maximale Größe einer Anfrage, die maximale URL-Länge und die maximale Länge einer Abfragezeichenfolge für den Webserver konfiguriert wurden, für alle Aufrufe der Alfabet RESTful-Services gelten.

Konfigurieren des Web-Servers, auf dem die Alfabet-Webapplikation gehostet wird, zur Aktivierung der Alfabet RESTful API

Folgende Einstellungen sind für den Webserver, auf dem die Alfabet-Webapplikation installiert ist, erforderlich, um das Alfabet RESTful API zu aktivieren:

- Das WebDAV-Modul der Internetinformationsdienste®, auf denen die Alfabet-Webapplikation läuft, ist nicht installiert.
- Das Alfabet RESTful API der Alfabet-Webapplikation unterstützt Windows Sign-On nicht. Für den `api`-Unterordner der Alfabet-Webapplikation auf dem Webserver muss Windows Sign-On deaktiviert und der anonyme Zugriff aktiviert sein.






Wenn die Alfabet-Webapplikation so konfiguriert ist, dass Anwender über die Portalauthentifizierung authentifiziert werden, kann die RESTful API für diese Alfabet-Webapplikation überhaupt nicht implementiert werden.

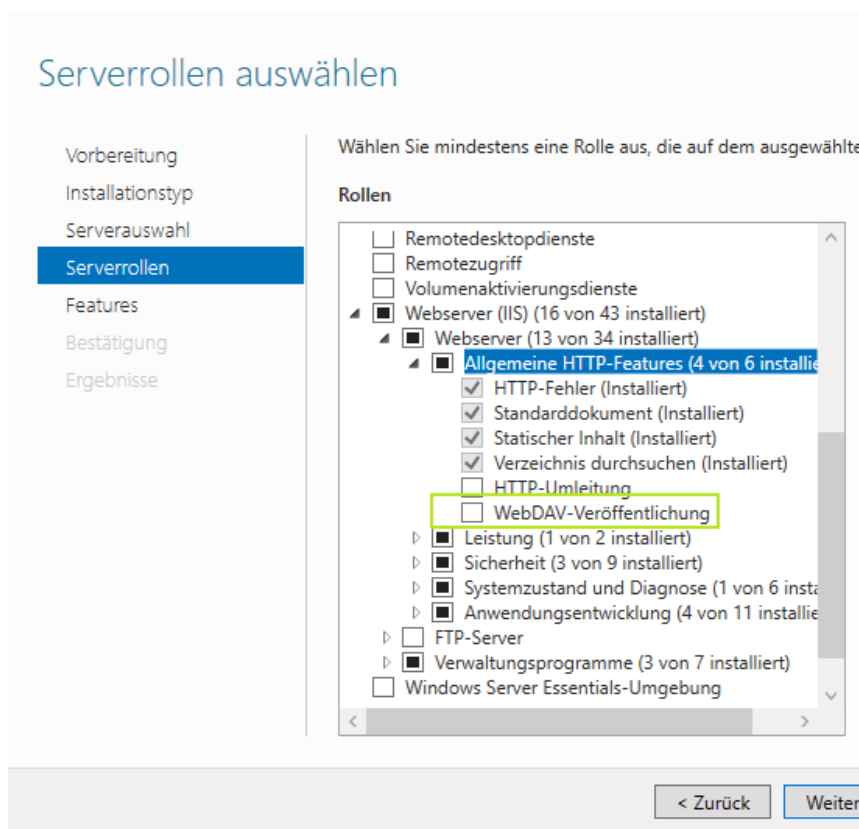
Um die Alfabet RESTful API in Kombination mit der Portalauthentifizierung zu verwenden, muss eine zweite Alfabet-Webapplikation konfiguriert werden, die auf die gleiche Alfabet-Datenbank zugreift. Dieses zusätzliche Alfabet-Webapplikation muss mit einem anderen Server-Alias ausgeführt werden, der für die Verwendung der Standardauthentifizierung über Anwendername und Passwort für den Anwenderzugriff auf die Alfabet-Datenbank konfiguriert ist. Die Alfabet RESTful API wird durch diese zusätzliche Alfabet-Webapplikation bereitgestellt. Alle in diesem Handbuch beschriebenen Konfigurationsschritte und URL-Angaben beziehen sich auf diese zusätzliche Alfabet-Webapplikation.

Deaktivieren Sie das WebDAV-Modul der Internetinformationsdienste

Das WebDAV-Modul kann in den Serverrollen deaktiviert werden:


- 1) Klicken Sie auf dem Webserverhost auf das Symbol **Start** , das angezeigt wird, wenn Sie den Mauszeiger in die linke untere Ecke verschieben, und klicken Sie auf das Symbol **Server-Manager** , um den Server-Manager zu öffnen.
- 2) Wählen Sie im Menü in der oberen rechten Ecke des Server-Managers **Verwalten > Rollen und Features hinzufügen** aus.
- 3) Wählen Sie auf den ersten drei Seiten des **Assistenten zum Hinzufügen von Rollen und Features** den Installationstyp und den Web-Serverhost aus, und wechseln Sie mit der Schaltfläche **Weiter** des Assistenten zur Seite **Serverrollen**.
- 4) Erweitern Sie auf der Seite **Serverrollen** die Option **Webserver (IIS)**.
- 5) Stellen Sie sicher, dass die Option **WebDAV-Publikation** nicht aktiviert ist:

 Assistent zum Hinzufügen von Rollen und Features



Einstellen der erforderlichen Autorisierung für den api-Ordner

Unabhängig von der für die Alfabet-Webapplikation ausgewählte Authentifizierung muss die Authentifizierung des `api`-Unterverzeichnisses der Alfabet-Webapplikation auf **Anonyme Authentifizierung** gesetzt werden:

- 1) Klicken Sie auf dem Webserverhost auf das Symbol **Start** , das angezeigt wird, wenn Sie den Mauszeiger in die linke untere Ecke verschieben, und klicken Sie auf das Symbol **Server-**



Manager, um den Server-Manager zu öffnen.

- 2) Wählen Sie im Menü in der oberen rechten Ecke des Server-Managers **Werkzeuge > Internetinformationsdienste (IIS)-Manager** aus.
- 3) Erweitern Sie im Explorer den Knoten der Alfabet-Webapplikation und klicken Sie auf den Knoten des `api`-Ordners.
- 4) Doppelklicken Sie im Bereich **IIS** im mittleren Bereich auf **Authentifizierung**. Der Status der verfügbaren Authentifizierungsmodi wird angezeigt.

Name	Status	Antworttyp
Anonyme Authentifizierung	Aktiviert	
ASP.NET-Identitätswechsel	Deaktiviert	
Formularauthentifizierung	Deaktiviert	HTTP 302 - Anmeldung.
Standardauthentifizierung	Deaktiviert	HTTP 401 - Abfrage
Windows-Authentifizierung	Deaktiviert	HTTP 401 - Abfrage


- 5) Setzen Sie die Option **Anonyme Authentifizierung** auf **Aktiviert** und die Option **Windows-Authentifizierung** auf **Deaktiviert**. Ändern der Einstellungen erfolgt durch Auswählen der Option in der Liste und Klicken auf **Deaktivieren** bzw. **Aktivieren** im Bereich **Aktionen** auf der rechten Seite.

Konfigurieren des Server-Alias der Alfabet-Webapplikation zum Aktivieren der REST-API-Aufrufe

Das Alfabet -API für RESTful Web-Services muss im Server-Alias der Alfabet-Webapplikation aktiviert werden. Die Konfiguration erfolgt im Konfigurationstool Alfabet Administrator:



Informationen über das Zugreifen auf und das Verwenden von Alfabet Administrator finden Sie unter *Arbeiten mit dem Alfabet Administrator* im Referenzhandbuch *Systemadministration*.

- 1) Klicken Sie im Alfabet Administrator -Explorer auf den Knoten **Alfabet Aliases**.
- 2) Klicken Sie in der Tabelle auf den Serveralias, den Sie konfigurieren möchten.
- 3) Klicken Sie in der Symbolleiste auf die Schaltfläche **Bearbeiten** . Ein Editor wird geöffnet.
- 4) Wechseln Sie zur Registerkarte **Server-Einstellungen > REST API**.
- 5) Setzen Sie das Häkchen für die Option **REST-API v2 aktivieren**.
- 6) Aktivieren Sie das Häkchen der RESTful-Service-Endpunkte, die in der Alfabet-Webapplikation aktiviert werden sollen, im Feld **API-Zugriffsoptionen**:
 - **Hat Zugriff auf Metamodell** : Aktiviert den Zugriff auf die Endpunkte `metamodel`, `classes` und `enums`, um Informationen über die Struktur des Alfabet -Klassenmodells einschließlich Aufzählungen und Landeseinstellungen lesen zu können.

- **Hat Zugriff auf GetObjectsByRefs (Objekte durch Referenzen erhalten)** : Aktiviert den Lesezugriff auf den `objects`-Endpunkt für Informationen zu den gespeicherten Daten für Objekte in der Alfabet -Datenbank, die über die Spezifikation des `REFSTR` des Objekts in der REST-API-Anfrage gefunden werden. Die Zugriffsoption **Hat Zugriff auf GetObjectsByRefs** ist auch für den Zugriff auf den Endpunkt `archiveobject` erforderlich, um Informationen zum Objekt in eine ZIP-Archivdatei zu exportieren, die relevante Ansichten als HTML-Dateien enthält.
- **Hat Zugriff auf GetObjectsByReport (Objekte durch Bericht erhalten)** : Aktiviert den Zugriff auf den `objects`-Endpunkt für Informationen zu den gespeicherten Daten für Objekte in der Alfabet -Datenbank, die über einen konfigurierten Bericht gefunden werden.
- **Hat den GetObjectsByFilter-Zugriff** : Aktiviert den Zugriff auf den `objects`-Endpunkt für Informationen zu den gespeicherten Daten für Objekte in der Alfabet -Datenbank, die über die Spezifikation der Objektklasse und der Filterbedingungen in der Anfrage gefunden werden.
- **Hat Zugriff auf CreateObjects (Objekte erstellen)** : Aktiviert den Zugriff auf den `update`-Endpunkt zum Erzeugen neuer Objekte in der Alfabet -Datenbank.
- **Hat Zugriff auf UpdateObjects (Objekte aktualisieren)** : Aktiviert den Zugriff auf den `update`-Endpunkt zum Aktualisieren der Daten für vorhandene Objekte in der Alfabet -Datenbank.

Darüber hinaus ist diese Zugriffsoption für den Zugriff auf den `regeneratepassword`-Endpunkt erforderlich. Beachten Sie, dass die Zugriffsoption **Hat PasswordAPI-Zugriff** ebenfalls ausgewählt sein muss, um den `regeneratepassword`-Endpunkt zu aktivieren.

- **Hat Zugriff auf DeleteObjects (Objekte löschen)** : Aktiviert den Zugriff auf den `delete`-Endpunkt zum Löschen vorhandener Objekte in der Alfabet -Datenbank. Die Berechtigung **Hat Zugriff auf GetObjectsByRefs** ist für den Zugriff auf den Endpunkt `archiveobject` ebenfalls erforderlich, um Informationen zum Objekt in eine ZIP-Archivdatei zu exportieren, die relevante Ansichten als HTML-Dateien enthält, auch wenn die Option, das Objekt nach dem Export des Archivs zu löschen, ausgewählt ist.
- **Hat den AnonymizeUser-Zugriff** : Aktiviert den Zugriff auf den `anonymizeuser`-Endpunkt zum Anonymisieren von Anwendern, die über die Spezifikation des `REFSTR` des Anwenders in der REST-API-Anfrage in der Alfabet-Datenbank gefunden werden.
- **Hat PasswordAPI-Zugriff** : Aktiviert den Zugriff auf den `regeneratepassword`-Endpunkt. Beachten Sie, dass die Option **Hat Zugriff auf UpdateObjects** ebenfalls ausgewählt sein muss, um den Endpunkt `regeneratepassword` verwenden zu können.
- **Hat den ADIFAPIInvocation-Zugriff** : Aktiviert den Zugriff auf die `adifimport`- und `adifexport`-Endpunkte, um die Ausführung von ADIF-Jobs, die auf einem ADIF-Schema in der Alfabet-Datenbank basieren, auszulösen, Hinweis: Diese Zugriffsoption ist auch zur asynchronen Ausführung des ADIF-Schemas über die Funktionalität **ADIF-Prozessverwaltung** und zur Ausführung von ADIF-Schemata über die Funktionalität **Prozesszeitplan** erforderlich.
- **Hat den WorkflowAPIInvocation-Zugriff** : Aktiviert den Zugriff auf den `workflow`-Endpunkt zum Starten von Workflows, die auf einer Workflow-Vorlage in der Alfabet-Datenbank basieren.
- **Hat Zugriff auf Überwachungs-API** : Aktiviert den Zugriff auf den `monitor`-Endpunkt, um zu prüfen, ob die Alfabet-Komponenten verfügbar sind.

- **Hat IDoc-Zugriff zum Hochladen** : Aktiviert den Zugriff auf den `idocupload`-Endpunkt, um Dokumente in die interne Dokumentenauswahl der Alfabet-Datenbank hochzuladen.
 - **Hat IDoc-Zugriff zum Herunterladen** : Aktiviert den Zugriff auf den `idocdownload`-Endpunkt, um Dokumente aus der internen Dokumentenauswahl der Alfabet-Datenbank herunterzuladen.
 - **Hat Zugriff auf IDoc-Dateiliste** : Aktiviert den Zugriff auf den `idocfilelist`-Endpunkt, um Informationen zum Inhalt der internen Dokumentenauswahl in der Alfabet-Datenbank zu exportieren.
 - **Hat Zugriff auf Batch-Hilfsprogramme-API**: Aktiviert den Zugriff auf die RESTful-Service-Funktionalität, die benötigt wird, um Batch-Prozesse über die Funktionalität **Prozesszeitplan** und zum asynchronen Exportieren oder Importieren von Datenerfassungsvorlagen auszuführen.
 - **Hat Zugriff auf Metamodell-Aktualisierung** : Aktiviert den Zugriff auf den `updateMM`-Endpunkt, um das Metamodell der Zieldatenbank mit der in einer AMM-Datei gespeicherten Metamodellkonfiguration zu aktualisieren.
 - **Hat Zugriff auf Ansichtsmomentaufnahmen**: Aktiviert den Zugriff auf den Endpunkt, der von der Microsoft Teams®-Integrationsfunktion verwendet wird, um Momentaufnahmen von Alfabet-Ansichten in den Microsoft Teams-Kanal einzuschließen. Diese Option sollte aktiviert werden, wenn die Microsoft Teams-Integration implementiert ist.
- 7) Klicken Sie auf **OK**, um Ihre Änderungen zu speichern.

Kapitel 4: Autorisierung

Die Autorisierung für das Alfabet Data Integration Framework erfolgt pro Benutzer, pro Klasse und pro Objekt:

- Der Zugriff auf das Alfabet RESTful API muss einem benannten Alfabet -Benutzer in der Benutzerkonfiguration in der Alfabet-Datenbank ausdrücklich gewährt werden. Die Zugriffsrechte können auf einen Teil der verfügbaren Endpunkte beschränkt werden.

Vor jedem Aufruf zum Zwecke einer Datenanfrage muss ein Autorisierungsaufruf erfolgen. Im Autorisierungsaufruf werden Anwendername und Merkmal an die Alfabet RESTful API der Alfabet-Webapplikation übermittelt, und es wird ein eindeutiger Autorisierungsschlüssel für Datenanforderungsaufrufe zurückgesendet. Dieser eindeutige Autorisierungsschlüssel kann für einen konfigurierbaren Zeitraum nach dem Autorisierungsaufruf für Aufrufe zur Datenanfrage genutzt werden. Wenn der Schlüssel abläuft, muss ein neuer Autorisierungsaufruf durchgeführt werden.

- Konfigurationsobjekte wie ADIF-Schemata, Workflow-Vorlagen und konfigurierte Berichte sind nur über die REST-API ausführbar, wenn die Attributeinstellungen des Konfigurationsobjekts die Berechtigung zur Ausführung über die REST-API enthalten.
- Daten über eine Objektklasse können über das Alfabet RESTful API nur dann gelesen, erstellt oder geändert werden, wenn die Klasseneinstellungen für die Objektklasse so konfiguriert sind, dass die entsprechenden Zugriffsrechte erteilt sind. Eine Objektklasse kann mehrere Klasseneinstellungen aufweisen. Die Klasseneinstellungen werden Ansichtsschemata zugeordnet. Ein Ansichtsschema wird dann einem Anwenderprofil zugeordnet. Dabei können einem benannten Anwender ein oder mehrere Anwenderprofile zugeordnet werden.

Parallel zur Anmeldung bei der Alfabet -Benutzeroberfläche werden REST-API-Aufrufe auch für einen bestimmten Anwender mit einem bestimmten Anwenderprofil durchgeführt. Das Anwenderprofil ist entweder in der REST-API-Anfrage definiert, oder es wird ein Standard aus den Anwenderprofilen ausgewählt, die dem Anwender zugeordnet sind. Ob der Anwender berechtigt ist, Daten für eine Objektklasse zu lesen, zu erstellen oder zu bearbeiten, hängt von den für das Anwenderprofil relevanten Klasseneinstellungen ab.

- Wenn die Mandantenfunktionalität in Ihrem Unternehmen implementiert ist, hat der Anwender nur dann Lesezugriff auf Daten zu einem Objekt, wenn die Mandanteneinstellungen sowohl für den Anwender als auch das Objekt Zugriff auf das Objekt gewähren. Sind dem Anwender mehrere Mandanten zugeordnet, kann der für den REST-API-Aufruf gültige Mandant in der Anfrage angegeben werden. Ist der in der Anfrage angegebene Mandant dem Anwender nicht zugeordnet, wird ein Fehler ausgegeben.



Einen Überblick über die Mandantenfunktionalität und das Konfigurieren von Mandanten finden Sie unter *Implementieren der Mandantenfähigkeit für eine föderative Architektur* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

- Ob ein Anwender Daten zu einem Objekt bearbeiten kann, hängt von den in Ihrer Alfabet -Lösung implementierten Zugriffsberechtigungskonzepten ab, wie z. B. Berechtigungsregeln, die Zugehörigkeit zu autorisierten Anwendergruppen oder die Zuweisung von Aufgaben zu einem Objekt im Zusammenhang mit Workflows oder Zuordnungen.



Einen vollständigen Überblick über die Objektberechtigungskonzepte in Alfabet finden Sie im Kapitel *Konfigurieren von Zugriffsberechtigungen für Alfabet* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

Erforderliche Konfiguration auf Serverseite

Vor dem Senden von Service-Aufrufen an das Alfabet RESTful API muss die Alfabet -Lösung so konfiguriert werden, dass alle erforderlichen Zugriffsrechte gewährt sind:

- [Generieren eines REST-API-Passworts für einen Anwender](#)
- [Aktivieren von Berichten, ADIF-Schemata und Workflow-Vorlagen, die über RESTful-Serviceaufrufe ausgeführt werden sollen](#)
- [Aktivieren des Zugriffs auf Ordner in der internen Dokumentenauswahl](#)
- [Konfigurieren nach Objektklassenberechtigungen für das Lesen oder Beschreiben der Alfabet-Datenbank](#)
- [Zugriff über Mandanten steuern](#)
- [Zugriff pro Objekt über Zugriffsberechtigungskonzepte steuern](#)

Generieren eines REST-API-Passworts für einen Anwender

Die Autorisierung erfolgt pro Anwender und setzt voraus, dass der Zugriff auf das Alfabet RESTful API explizit für einen benannten Alfabet -Anwender in der Anwenderkonfiguration in der Alfabet-Datenbank erteilt wurde. Die für die Konfiguration der Autorisierung erforderlichen Editor-Felder sind im Anwender-Editor nur verfügbar, wenn für ADIF eine gültige Lizenz vorhanden ist.

Die Anwenderkonfiguration erfolgt in der Funktionalität **Anwenderverwaltung** auf der Benutzeroberfläche. Diese steht über das `Admin`-Anwenderprofil oder die Funktionalität **Anwenderverwaltung** zur Verfügung, auf die wiederum über den verbundenen Server-Alias-Knoten in Alfabet Administrator zugegriffen werden kann.




Detaillierte Informationen über das Erstellen benannter Alfabet -Anwender finden Sie im Kapitel *Definieren und Verwalten von Anwendern* im Referenzhandbuch *Anwender- und Lösungsverwaltung* oder *Verwalten neuer und vorhandener Anwender* im Referenzhandbuch *Systemadministration*.

Berücksichtigen Sie bei der Gewährung von REST-API-Zugriffsberechtigungen für einen benannten Anwender aus Sicherheitsgründen und zur Vermeidung technischer Konflikte das Folgende:

- Der Anwender sollte ausschließlich für Anfragen an das Alfabet RESTful API und nicht parallel für den Zugriff auf die Alfabet -Anwendung über die Alfabet -Benutzeroberfläche oder für den Zugriff auf die Alfabet-Datenbank über andere von Alfabet bereitgestellte Schnittstellen, wie z. B. ADIF oder ARIS/Alfabet-Interoperabilitätsschnittstelle, verwendet werden.
- Es können mehrere Anwender mit Zugriff auf einen Teil der Endpunkte des Alfabet RESTful API definiert werden, die entweder Lesezugriff oder Lese-/Schreibzugriff haben. Wenn mehrere Client-Anwendungen so konfiguriert sind, dass sie Anfragen an das Alfabet RESTful API senden, sollten für jede Client-Anwendung separate Anwender verwendet werden, deren Zugriffsberechtigungen genau den jeweiligen Anforderungen entsprechen.
- Einer der Anwender, die für den Zugriff auf die Alfabet -RESTful Services konfiguriert sind, sollte für die Ausführung von selbstreflektierenden Ereignissen definiert werden. Viele Alfabet -Prozesse basieren auf der Ausführung von RESTful Service-Aufrufen im Hintergrund. Dieses Prozesse

werden mithilfe des Anwenders autorisiert, der zur Verwendung für selbstreflektierende Ereignisse definiert ist.

Nach dem Erstellen eines benannten Anwenders müssen die folgenden Einstellungen durchgeführt werden, um Zugriff auf das Alfabet RESTful API zu gewähren:

- 1) Wählen Sie in der Tabelle der Funktionalität **Anwenderverwaltung** den Anwender aus, der zum Versenden von Anfragen über das Alfabet RESTful API verwendet werden soll.
- 2) Klicken Sie in der Symbolleiste auf die Schaltfläche **Bearbeiten** . Der Editor **Anwender** wird angezeigt.
- 3) Wechseln Sie im Editor zur Registerkarte **Berechtigungen**.
- 4) Markieren Sie das Kontrollkästchen **Hat Zugriff auf API V2**.
- 5) Geben Sie die Details über die Zugriffsberechtigungen mit den folgenden Attributen an:
 - **API Token-Dauer (Minuten):** Die RESTful-Serviceschnittstelle auf Clientseite muss so implementiert werden, dass vor der Übermittlung einer Datenanfrage eine Anfrage für den Autorisierungscode gesendet wird. Der Autorisierungscode, der in der Antwort der Autorisierungsanfrage enthalten ist, kann in Aufrufen zur Alfabet RESTful API für die Datenanfrage verwendet werden, die innerhalb einer begrenzten Zeit nach Erhalt des Autorisierungscode übermitteln werden. Geben Sie an, wie viele Minuten der Autorisierungscode gültig sein soll. Der Autorisierungscode kann standardmäßig 20 Minuten lang verwendet werden.
 - **API-Zugriffsoptionen:** Standardmäßig sind die meisten Optionen in diesem Feld aktiviert, um dem Anwender Zugriff auf alle zentralen Funktionen zu gewähren, die für die Alfabet RESTful Services bereitgestellt werden. Deaktivieren Sie alle Optionen, zu deren Ausführung der Anwender keine Berechtigung erhalten soll:
 - **Hat Zugriff auf Metamodell :** Wenn diese Option ausgewählt ist, hat der Anwender Zugriff auf die Endpunkte `metamodel`, `classes` und `enums`, um Informationen über die Struktur des Alfabet -Klassenmodells einschließlich Aufzählungen und Landeseinstellungen lesen zu können.
 - **Hat Zugriff auf GetObjectsByRefs (Objekte durch Referenzen erhalten) :** Wenn diese Option ausgewählt ist, hat der Anwender Lesezugriff auf den `objects`-Endpunkt für Informationen zu den gespeicherten Daten für Objekte in der Alfabet -Datenbank, die über die Spezifikation des `REFSTR` des Objekts in der REST-API-Anfrage gefunden werden. Die Berechtigung **Hat Zugriff auf GetObjectsByRefs** ist auch für den Zugriff auf den Endpunkt `archiveobject` erforderlich, um Informationen zum Objekt in eine ZIP-Archivdatei zu exportieren, die relevante Ansichten als HTML-Dateien enthält.
 - **Hat Zugriff auf GetObjectsByReport (Objekte durch Bericht erhalten) :** Wenn diese Option ausgewählt ist, hat der Anwender Lesezugriff auf den `objects`-Endpunkt für Informationen zu den gespeicherten Daten für Objekte in der Alfabet -Datenbank, die über einen konfigurierten Bericht gefunden werden.
 - **Hat den GetObjectsByFilter-Zugriff :** Wenn diese Option ausgewählt ist, hat der Anwender Lesezugriff auf den `objects`-Endpunkt für Informationen zu den gespeicherten Daten für Objekte in der Alfabet -Datenbank, die über die Spezifikation der Objektklasse und der Filterbedingungen in der Anfrage gefunden werden.

- **Hat Zugriff auf CreateObjects (Objekte erstellen)** : Wenn diese Option ausgewählt ist, hat der Anwender Zugriff auf den `update`-Endpunkt zum Erzeugen neuer Objekte in der Alfabet -Datenbank.
- **Hat Zugriff auf UpdateObjects (Objekte aktualisieren)** : Wenn diese Option ausgewählt ist, hat der Anwender Zugriff auf den `update`-Endpunkt zum Aktualisieren der Daten für vorhandene Objekte in der Alfabet -Datenbank.

Darüber hinaus ist diese Berechtigung für den Zugriff auf den `regeneratepassword`-Endpunkt erforderlich. Beachten Sie, dass die Berechtigung **Hat PasswordAPI-Zugriff** ebenfalls ausgewählt sein muss, um den Endpunkt `regeneratepassword` verwenden zu können.

- **Hat Zugriff auf DeleteObjects (Objekte löschen)** : Wenn diese Option ausgewählt ist, hat der Anwender Zugriff auf den `delete`-Endpunkt zum Löschen vorhandener Objekte in der Alfabet -Datenbank. Die Berechtigung **Hat Zugriff auf DeleteObjects** ist für den Zugriff auf den `archiveobject`-Endpunkt ebenfalls erforderlich, um Informationen zum Objekt in eine ZIP-Archivdatei zu exportieren, die relevante Ansichten als HTML-Dateien enthält, auch wenn die Option, das Objekt nach dem Export des Archivs zu löschen, ausgewählt ist.
- **Hat den AnonymizeUser-Zugriff** : Wenn diese Option ausgewählt ist, hat der Anwender Zugriff auf den `anonymizeuser`-Endpunkt zum Anonymisieren von Anwendern, die über die Spezifikation des `REFSTR` des Anwenders in der REST-API-Anfrage in der Alfabet-Datenbank gefunden werden.
- **Hat PasswordAPI-Zugriff** : Wenn diese Option ausgewählt ist, hat der Anwender Zugriff auf den `regeneratepassword`-Endpunkt. Beachten Sie, dass die Berechtigung **Hat Zugriff auf UpdateObjects** ebenfalls ausgewählt sein muss, um den Endpunkt `regeneratepassword` verwenden zu können.
- **Hat den ADIFAPIInvocation-Zugriff** : Wenn diese Option ausgewählt ist, hat der Anwender Zugriff auf die `adifimport`- und `adifexport`-Endpunkte, um die Ausführung von ADIF-Jobs, die auf einem ADIF-Schema in der Alfabet-Datenbank basieren, auszulösen,
- **Hat den WorkflowAPIInvocation-Zugriff** : Wenn diese Option ausgewählt ist, hat der Anwender Zugriff auf den REST-Endpunkt `workflow` zum Starten von Workflows, die auf einer Workflow-Vorlage in der Alfabet-Datenbank basieren.
- **Hat Zugriff auf Überwachungs-API** : Wenn diese Option ausgewählt ist, hat der Anwender Zugriff auf den `monitor`-Endpunkt, um zu prüfen, ob die Alfabet-Komponenten verfügbar sind.
- **Hat IDoc-Zugriff zum Hochladen** : Wenn diese Option ausgewählt ist, hat der Anwender Zugriff auf den `idocupload`-Endpunkt, um Dokumente in die **interne Dokumentenauswahl** der Alfabet-Datenbank hochzuladen.
- **Hat IDoc-Zugriff zum Herunterladen** : Wenn diese Option ausgewählt ist, hat der Anwender Zugriff auf den `idocdownload`-Endpunkt, um Dokumente aus der **internen Dokumentenauswahl** der Alfabet-Datenbank herunterzuladen.
- **Hat Zugriff auf IDoc-Dateiliste** : Wenn diese Option ausgewählt ist, hat der Anwender Zugriff auf den `idocfilelist`-Endpunkt, um Informationen zum Inhalt der internen Dokumentenauswahl in der Alfabet-Datenbank zu exportieren.

- **Hat Zugriff auf Batch-Hilfsprogramme-API:** Wenn diese Option ausgewählt ist, hat der Anwender Zugriff auf die RESTful-Service-Funktionalität, die benötigt wird, um Batch-Prozesse über die Funktionalität **Prozesszeitplan** und zum asynchronen Exportieren oder Importieren von Datenerfassungsvorlagen auszuführen.
- **Hat Zugriff auf Metamodell-Aktualisierung :** Wenn diese Option ausgewählt ist, hat der Anwender Zugriff auf den `updateMM`-Endpunkt, um das Metamodell der Zieldatenbank mit der in einer AMM-Datei gespeicherten Metamodellkonfiguration zu aktualisieren.
- **Hat Zugriff auf Ansichtsmomentaufnahmen:** Aktiviert den Zugriff auf den Endpunkt, der von der Microsoft Teams@-Integrationsfunktion verwendet wird, um Momentaufnahmen von Alfabet-Ansichten in den Microsoft Teams-Kanal einzuschließen. Diese Option sollte für den Anwender, der für die Ausführung selbstreflektierender Ereignisse konfiguriert ist, aktiviert werden, wenn die Microsoft Teams-Integration implementiert ist.



Für den Zugriff auf Objektdaten, ADIF-Schemata, Workflows, Berichte und die Interne Dokumentenauswahl sind zusätzliche Berechtigungseinstellungen erforderlich, die in den folgenden Abschnitten erläutert werden. Eine Übersicht aller Einstellungen, die für individuelle Endpunkte erforderlich sind, finden Sie im Abschnitt [Übersicht über für jeden Endpunkt erforderliche Zugriffsberechtigungen](#).

- 6) Klicken Sie auf die Schaltfläche **API-Passwort generieren**. Es wird ein Code generiert und in der Alfabet-Datenbank gespeichert.
- 7) Kopieren Sie den Code, und speichern Sie die Informationen zum Anwendernamen und Code für die Verwendung auf Clientseite.
- 8) Wenn der Anwender für die Ausführung von Ereignissen des Typs `SelfReflective` verwendet werden soll, wählen Sie den Anwender in der Tabelle aus, und wählen Sie in der Symbolleiste **Aktion > Anwender als berechtigt für „Selbstreflektierende Ereignisse ausführen“ festlegen** aus.



Es kann nur ein Anwender zur Ausführung von Ereignissen des Typs `SelfReflective` ausgewählt werden. Wenn Sie diese Funktion einem Anwender zuweisen und bereits ein anderer Anwender zur Ausführung von Selbstreflexionsereignissen ausgewählt wurde, wird die Einstellung von diesem Anwender entfernt, wenn sie für jenen Anwender festgelegt wird, dem Sie die Einstellung aktuell zuweisen.

Der generierte Code ist für die Autorisierung des Clients bei Anfragen zu verwenden, die zusammen mit dem Anwendernamen an das Alfabet RESTful API der Alfabet-Webapplikation gesendet werden.



Ein Passwort kann auch dann für einen Anwender generiert werden, wenn das Kontrollkästchen **Hat API-Zugriff** nicht aktiviert ist. Der generierte Code kann allerdings nur für den Zugriff auf das Alfabet RESTful API verwendet werden, wenn das Kontrollkästchen **Hat API-V2-Zugriff** für den Anwender aktiviert ist.

Aktivieren von Berichten, ADIF-Schemata und Workflow-Vorlagen, die über RESTful-Serviceaufrufe ausgeführt werden sollen

Einige Endpunkte umfassen die Ausführung von Anwenderkonfigurationsobjekten, die von einem Lösungsadministrator in Alfabet Expand erstellt wurden:

- Konfigurierte Berichte können über einen Serviceaufruf an den Endpunkt `objects` ausgeführt werden, um vordefinierte Objektdaten zu exportieren.
- ADIF-Schemata können über einen Serviceaufruf an die Endpunkte `adifimport` oder `adifexport` ausgeführt werden.
- Workflows können für eine Workflow-Vorlage über einen Serviceaufruf an den Endpunkt `workflow` gestartet werden.

Die erforderlichen Zugriffsberechtigungen zur Ausführung über RESTful-Serviceaufrufe müssen in den Attributen des konfigurierten Berichts, ADIF-Schemas oder der Workflow-Vorlage in Alfabet Expand definiert sein.


- 1) Klicken Sie auf den konfigurierten Bericht, die Workflow-Vorlage oder das ADIF-Schema, die Sie über die RESTful-API im jeweiligen Explorer von Alfabet Expand ausführen möchten.
- 2) Setzen Sie das Attribut **Für REST-API verwendbar** im Attributfenster auf `True`.
- 3) Überprüfen Sie, ob folgende zusätzliche Konfigurationsanforderungen erfüllt sind:
 - **Für Workflow-Vorlagen:** Nur Workflow-Vorlagen, bei denen das Attribut **Automatischer Start** auf `True` festgelegt ist, können über RESTful Serviceaufrufe ausgeführt werden.
 - **Für ADIF-Schemata:** Keine.
 - **Für konfigurierte Berichte:** Der konfigurierte Bericht muss einen tabellarischen Datensatz zurückgeben.
- 4) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste auf die Schaltfläche **Speichern**



Aktivieren des Zugriffs auf Ordner in der internen Dokumentenauswahl

Ordnerzugriffsberechtigungen, die in der **internen Dokumentenauswahl** angegeben sind, werden für REST-API-Aufrufe an `idocupload`, `idocdownload` und `idocfilelist` und an die Endpunkte `adifimport` und `adifexport` ausgewertet, wenn der Import oder Export Dateien in der **internen Dokumentenauswahl** betrifft.

Ordnerberechtigungen werden in der Funktionalität **Interne Dokumente** auf der Alfabet Benutzeroberfläche festgelegt:

- 1) Klicken Sie in der Funktionalität **Interne Dokumente** auf den übergeordneten Ordner desjenigen Ordners, für den Sie Zugriffsberechtigungen festlegen möchten.
- 2) Klicken Sie in der Tabelle auf den Ordner, für den Sie Zugriffsberechtigungen festlegen möchten.
- 3) Klicken Sie in der Symbolleiste auf die Schaltfläche **Bearbeiten** .
- 4) Setzen Sie im Editor die Häkchen für die folgenden Zugriffsberechtigungen im Feld **Standard-Zugriffsberechtigungen**:
 - Für die Endpunkte `idocupload` und `adifexport`: **Elemente verwalten**
 - Für die Endpunkte `idocdownload` und `adifimport`: **Elemente öffnen**
 - Für den Endpunkt `idocfilelist`: **Listenelemente**

- 5) Klicken Sie zum Schließen des Editors auf **OK**.



Aus Sicherheitsgründen sind in Alfabet eine Blacklist und Whitelist verfügbar, um das Hochladen und Herunterladen von Dateien in die bzw. aus der **interne(n) Dokumentenauswahl** auf bestimmte Dateierweiterungen zu beschränken. Dateierweiterungen, die nicht zum Hochladen und Herunterladen zugelassen sind, können in einer Blacklist angegeben werden. Alternativ kann eine Whitelist konfiguriert werden, die explizit und ausschließlich die angegebenen Dateierweiterungen für das Hochladen und Herunterladen zulässt. Diese Einstellungen gelten auch für das Hoch- und Herunterladen von Dateien über die Alfabet -RESTful-Services.

Weitere Einzelheiten zur Definition von Blacklist und Whitelist finden Sie im Abschnitt *Konfiguration der Zulässigkeit von Dateien und Weblinks in Alfabet* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

Konfigurieren nach Objektklassenberechtigungen für das Lesen oder Beschreiben der Alfabet-Datenbank

Wenn Sie möchten, dass Objektdaten über das Alfabet RESTful API gelesen oder geschrieben werden, müssen Sie ein Anwenderprofil definieren, das die erforderlichen Berechtigungen über die Klasseneinstellungen einräumt, die dem Ansichtsschema des Anwenderprofils zugeordnet sind.

In Alfabet bestimmen das Anwenderprofil, mit dem ein Anwender angemeldet ist, und das Ansichtsschema sowie die Klasseneinstellungen, die diesem Anwenderprofil zugewiesen sind, den Umfang der Funktionalität, die dem Anwender zur Verfügung steht.




Detailinformationen über das Konfigurieren von Klasseneinstellungen, Ansichtsschemata und Anwenderprofilen finden Sie im Kapitel *Konfigurieren von Anwenderprofilen für die Anwender-Community* des Referenzhandbuchs *Konfigurieren von Alfabet mit Alfabet Expand*.

Hinweis: Mit dem vordefinierten Anwenderprofil `Admin` und jedem als administratives Anwenderprofil gekennzeichneten Anwenderprofil ist der Zugriff auf alle Objekte über die Alfabet -Benutzeroberfläche unabhängig von Zugriffsberechtigungskonzepten möglich, was aber nicht im Kontext der Alfabet -RESTful-Services verwendet werden sollte.

Der überwiegende Teil der Konfiguration eines Anwenderprofils und des zugrunde liegenden Ansichtsschemas und der Klasseneinstellungen ist unerheblich für die Verwendung des Alfabet RESTful API. Die erforderliche Einstellung für das Erteilen einer Berechtigung zum Erzeugen von Objekten für eine Objektklasse wird ausschließlich über zwei Attribute in der Klasseneinstellung der Objektklasse festgelegt. Es wird empfohlen, die Klasseneinstellungen für exklusive Verwendung über das Alfabet RESTful API festzulegen und sie nur einem Ansichtsschema und einem Anwenderprofil zuzuweisen, das über das Alfabet RESTful API ausdrücklich für die Verwendung definiert worden ist. Das Anwenderprofil kann nur Anwendern zugeordnet werden, die autorisiert sind, Daten über das Alfabet RESTful API zu lesen oder zu ändern.


Das Konfigurieren von Klasseneinstellungen, Ansichtsschemata und Anwenderprofilen geschieht in Alfabet Expand. Die folgende Konfiguration ist erforderlich:

- 1) Wechseln Sie zur Registerkarte **Präsentation**, und erweitern Sie den Knoten **Klasseneinstellungen**.
- 2) Navigieren Sie im Ordner **Klasseneinstellungen** zu dem Objektklassenordner, für den Sie Klasseneinstellungen definieren wollen. Klicken Sie mit der rechten Maustaste auf die zu kopierende öffentliche oder private Klasseneinstellungsvorlage , und wählen Sie die Option

Neue Klasseneinstellungen als Kopie aus, um eine neue Klasseneinstellungsvorlage für die gewählte Objektklasse zu erzeugen.



Die Option **Neue Klasseneinstellungen für Stereotyp als Kopie** darf in diesem Kontext nicht verwendet werden. Die REST-API-Berechtigungen für Objektklassenstereotypen werden ignoriert. Berechtigungen können nur auf der Ebene der Objektklasse definiert werden.

- 3) Eine Kopie der Klasseneinstellungsvorlage  wird dem Klasseneinstellungsordner hinzugefügt. Alle für die kopierte Klasseneinstellung angegebenen Attribute werden ebenfalls in die neue Klasseneinstellung kopiert.
- 4) Klicken Sie auf die neue Klasseneinstellungsvorlage, um das Attributfenster aufzurufen, und geben Sie im Attribut **Name** eine technische Bezeichnung für die Klasseneinstellung ein.



Es wird empfohlen, dass der Name der Klasseneinstellung auf das Ansichtsschema hinweisen sollte, für das sie definiert wird, und dass der Name des Ansichtsschemas auf das Anwenderprofil hinweist, dem es zugeordnet ist. Das bedeutet, dass der Name der Klasseneinstellung auf das Anwenderprofil hinweisen sollte, zu dem sie gehört.

Beim Festlegen eines technischen Namens wird die Syntax über einen Prüfmechanismus auf Korrektheit geprüft. Eine Liste der unzulässigen Sonderzeichen finden Sie im Abschnitt *Definieren von Attributen für Konfigurationsobjekte* im Kapitel *Erste Schritte mit Alfabet Expand* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.




- 5) Abhängig von den Rechten, die für die Klasse gewährt werden sollen, setzen Sie ein oder beide der folgenden Attribute auf `True`:
 - **Lesen über REST-API zulassen** : Stellen Sie `True` ein, um das Lesen der Daten von Objekten dieser Objektklasse über den Alfabet RESTful API-Endpunkt `objects` zuzulassen.
 - **Schreiben über REST-API zulassen** : Stellen Sie `True` ein, um das Aktualisieren von Objekten dieser Objektklasse, das Erzeugen von Objekten und das Löschen von Objekten über die `delete`-und `update`-Endpunkte zuzulassen.
- 6) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste auf die Schaltfläche **Speichern** .
- 7) Wiederholen Sie die Schritte 2) bis 6) für alle Objektklassen, die über das Alfabet RESTful API bearbeitbar sein sollen.
- 8) Wechseln Sie zur Registerkarte **Präsentation**, und erweitern Sie den Knoten **Ansichtsschemata**.
- 9) Wenden Sie zur Erstellung von Ansichtsschemata eines der folgenden Verfahren an:
 - Klicken Sie mit der rechten Maustaste auf den Ordner **Ansichtsschemata**, und wählen Sie **Neues Ansichtsschema**, um ein völlig neues Ansichtsschema zu erstellen, oder
 - Klicken Sie auf ein Ansichtsschema, das Sie kopieren möchten, und wählen Sie **Neues Ansichtsschema als Kopie**, um ein neues Ansichtsschema auf Basis eines vorhandenen Ansichtsschemas zu erstellen.
- 10) Das neue Ansichtsschema  wird dem Ordner **Ansichtsschemata** hinzugefügt. Geben Sie im Attributfenster einen Namen in das Attribut **Name** ein.



Beim Festlegen eines technischen Namens wird die Syntax über einen Prüfmechanismus auf Korrektheit geprüft. Der technische Name eines Konfigurationsobjekts darf nicht mit einem Leerschritt beginnen und keines der folgenden Sonderzeichen enthalten: `\ / * ? " ' > < | ' :`

Darüber hinaus ist zu beachten, dass der Name nicht mit einem Leerschritt beginnen und keine Sonderzeichen enthalten darf. Eine Liste der unzulässigen Sonderzeichen finden Sie im Abschnitt *Definieren von Attributen für Konfigurationsobjekte* im Kapitel *Erste Schritte mit Alfabet Expand*.

Außerdem wird der technische Name eines Konfigurationsobjekts bei seiner Änderung in anderen Konfigurationsobjekten, die auf das geänderte Objekt während der Entwicklungszeit verweisen, korrekt aktualisiert. Dabei ist zu beachten, dass der Name des geänderten Konfigurationsobjekts auf den Navigationsseiten nicht aktualisiert wird und die Navigationsseiten auch nicht auf ein Konfigurationsobjekt verweisen, das in der Funktionalität **Nutzung anzeigen** aufgelistet ist. Wenn Sie planen, den Namen eines Konfigurationsobjekts anzupassen, sollten Sie erst den Verweis auf der Navigationsseite und dann den Namen des Konfigurationsobjekts ändern.

- 11) Sie sollten nun systematisch bestimmen, ob die einzelnen Objektklassen/Objektklassenstereotypen eine benutzerdefinierte Klasseneinstellung benötigen oder ob die Standardklasseneinstellung für Ihre Bedürfnisse ausreicht. Klicken Sie zum Öffnen des Editors **Ansichtsschema** für das ausgewählte Ansichtsschema doppelt auf das Ansichtsschema, das Sie konfigurieren. Der Editor **Ansichtsschema** wird im mittleren Bereich geöffnet.
- 12) In der Tabelle mit dem Titel **Objektklassenkonfiguration** werden Objektklassen/Objektklassenstereotypen auf der ersten Ebene angezeigt. Klicken Sie auf „+“, um die Tabelle unter der zu konfigurierenden Objektklasse bzw. dem Objektklassenstereotyp zu erweitern. Wenn Sie den Objektklassenknoten erweitern, sehen Sie alle vorhandenen von Ihrem Unternehmen für die ausgewählte Objektklasse konfigurierten Standardklasseneinstellungen und benutzerdefinierten Klasseneinstellungen. Um anzugeben, welche Klasseneinstellung in das ausgewählte Ansichtsschema implementiert werden soll, klicken Sie in die Spalte **In Ansichtsschema verwenden**, setzen Sie ein X für die Klasseneinstellung, die für das ausgewählte Ansichtsschema implementiert werden soll. Führen Sie diese Prozedur für alle Objektklassen mit einer Konfiguration der Klasseneinstellung für das Alfabet RESTful API durch.
- 13) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste auf die Schaltfläche **Speichern** .
- 14) Wechseln Sie zur Registerkarte **Admin**, klicken Sie mit der rechten Maustaste auf den Knoten **Anwenderprofile**, und wählen Sie **Neues Anwenderprofil** aus. Ein neues Anwenderprofil  wird im Explorer angezeigt.
- 15) Definieren Sie im Attributfenster folgende Attribute:
 - **Name:** Geben Sie einen Titel für das Anwenderprofil ein. Dies ist der Name des Anwenderprofils, den Anwender später zu sehen bekommen, wenn sie sich bei Alfabet anmelden.
 - **Typ :** Sofern das Anwenderprofil über Bearbeitungsberechtigungen verfügen soll, wählen Sie `Read/Write` aus. Sofern das Anwenderprofil nur über Ansichtsberechtigungen verfügen soll, wählen Sie `ReadOnly` aus.
 - **Ansichtsschema:** Klicken Sie auf die Schaltfläche **Dropdown** , um das entsprechende Ansichtsschema auszuwählen, das beim Zugriff durch eine externe Applikation oder einen Hyperlink in einer E-Mail-Benachrichtigung für das Anwenderprofil gilt.

- 16) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste auf die Schaltfläche **Speichern**



Nachdem Sie das Anwenderprofil definiert haben, müssen Sie es dem Anwender zuordnen, der für den Zugriff auf die Alfabet RESTful API bereits definiert worden ist (siehe Abschnitt [Generieren eines REST-API-Passworts für einen Anwender](#)).

Die Anwenderkonfiguration erfolgt in der Funktionalität **Anwenderverwaltung** auf der Benutzeroberfläche (aufrufbar über das Anwenderprofil `Admin`).

- 1) Wählen Sie in der Tabelle der Funktionalität **Anwenderverwaltung** den Anwender aus, der zum Versenden von Anfragen für die Veränderung von Daten über das Alfabet RESTful API verwendet werden soll.
- 2) Klicken Sie in der Symbolleiste auf die Schaltfläche **Navigieren** . Das Objektprofil für den Anwender wird angezeigt.
- 3) Klicken Sie im Abschnitt **Lösungskonfiguration des Anwenders** auf den Link **Zugeordnete Anwenderprofile**. Die Ansichtsseite **Zugeordnete Anwenderprofile** wird geöffnet.
- 4) Klicken Sie in der Symbolleiste auf **Neu > Anwenderprofile zuordnen**. Der Selektor **Anwenderprofil** wird angezeigt.
- 5) Wählen Sie das Anwenderprofil aus, das Sie für das Ändern von Daten über das Alfabet RESTful API konfiguriert haben.
- 6) Klicken Sie auf **OK**, um Ihre Auswahl zu speichern.

Zugriff über Mandanten steuern

Wenn Ihr Unternehmen das Mandantenkonzept verwendet, werden für Antworten der Alfabet RESTful API beim Anfragen von Objektdateien oder dem Erzeugen, Löschen oder Aktualisieren von Objekten auch die Mandanteneinstellungen für Anwender und Objekte in der Alfabet-Datenbank in Betracht gezogen.



Mehr Informationen zum Mandantenkonzept und den erforderlichen Konfigurationsschritten finden Sie unter *Implementieren der Mandantenfähigkeit für eine föderative Architektur* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

In den Nutzdaten des Serviceaufrufs kann mit dem Feld `CurrentMandate` ein Mandant für den Aufruf definiert werden, wie in der Beschreibung der Serviceaufrufe im Abschnitt [Serviceaufrufe und Rückgabewerte](#) beschrieben. Diese Einstellung ist optional, es wird jedoch empfohlen, den Mandanten in dem Aufruf zu definieren. Das System überprüft, ob der in dem Aufruf definierte Mandant auch für den Anwender definiert ist, der zum Senden des Serviceaufrufs verwendet wird. Dem Anwender muss entweder der Mandant zugeordnet sein, oder der Anwender muss als Mandant-Master definiert sein. Ansonsten wird der Serviceaufruf abgelehnt.

Wenn das Mandantenkonzept implementiert ist, gelten die folgenden Regeln für die Rückgabe von Daten über Aufrufe des Alfabet RESTful API:

	In dem Serviceaufruf ist kein CurrentMandate definiert	In dem Serviceaufruf ist ein CurrentMandate definiert
Der Anwender ist ein Mandant-Master und hat zusätzlich einen oder mehrere Mandanten zugeordnet.	Der Anwender sieht Daten über alle Objekte, unabhängig von der Mandantenzuordnung der Objekte in den Rückgabeaufrufen.	Der Anwender sieht Daten über alle dem Mandanten zugeordneten Objekte, die mit CurrentMandate definiert sind, und über alle Objekte, denen kein Mandant zugeordnet ist.
Der Anwender hat einen einzelnen zugeordneten Mandanten und ist nicht der Mandant-Master.	Der Anwender sieht Daten über alle Objekte, die dem ihm zugeordneten Mandanten zugeordnet sind, und über alle Objekte, denen kein Mandant zugeordnet ist.	Wenn der mit CurrentMandate definierte Mandant dem Anwender zugeordnet ist, sieht der Anwender Daten über alle diesem Mandanten zugeordneten Objekte und über alle Objekte, denen kein Mandant zugeordnet ist. Wenn der mit CurrentMandate definierte Mandant dem Anwender nicht zugeordnet ist, wird der Serviceaufruf mit einer Meldung abgelehnt, die Auskunft darüber gibt, dass der CurrentMandate dem Anwender nicht zugeordnet ist
Der Anwender hat mehrere zugeordnete Mandanten und ist nicht der Mandant-Master.	Der Anwender sieht Daten über alle Objekte, die dem ihm zugeordneten Standardmandanten zugeordnet sind, und über alle Objekte, denen kein Mandant zugeordnet ist. Der erste Mandant in alphabetischer Reihenfolge der Mandanten, die dem Benutzer zugeordnet sind, ist der verwendete Standardmandant.	Wenn der mit CurrentMandate definierte Mandant dem Anwender zugeordnet ist, sieht der Anwender Daten über alle diesem Mandanten zugeordneten Objekte und über alle Objekte, denen kein Mandant zugeordnet ist. Wenn der mit CurrentMandate definierte Mandant dem Anwender nicht zugeordnet ist, wird der Serviceaufruf mit einer Meldung abgelehnt, die Auskunft darüber gibt, dass der CurrentMandate dem Anwender nicht zugeordnet ist
Der Anwender hat keine zugeordneten Mandanten und ist kein Mandant-Master.	Der Anwender sieht Daten über alle Objekte, die keinem Mandanten zugeordnet sind.	Der Serviceaufruf wird mit einer Meldung abgelehnt, die Auskunft darüber gibt, dass dem Anwender keine Mandanten zugeordnet sind.

Wenn Daten über ein Objekt mit einem Serviceaufruf an den objects-Endpunkt angefragt werden und die Mandanteneinstellungen des Objekts nicht mit den Mandanteneinstellungen des Anwenders oder dem in dem Serviceaufruf definierten Mandanten übereinstimmen, werden keine Daten zurückgegeben. Stattdessen werden dem Rückgabewert zwei Meldungen über die Nichtübereinstimmung des Mandanten hinzugefügt:

"RejectedObjects": [

```

    {
      "Id": "",
      "RefStr": "76-3473-0",
      "Message": "Mandate mismatch"
    }
  ],
  "AccessDenied": {
    "76-3473-0": "Mandate mismatch"
  }
}

```

Wenn Daten eines Objekts mit einem Serviceaufruf an den `update`-Endpunkt aktualisiert werden sollen oder ein Objekt mit einem Serviceaufruf an den `delete`-Endpunkt gelöscht werden soll, enthält der Rückgabewert auch das Feld `AccessDenied`, das über die Nichtübereinstimmung des Mandanten für einzelne Objekte informiert.

Zugriff pro Objekt über Zugriffsberechtigungskonzepte steuern

Wenn dem Benutzer, der den Serviceaufruf an das Alfabet RESTful API wie im Abschnitt [Konfigurieren nach Objektklassenberechtigungen für das Lesen oder Beschreiben der Alfabet-Datenbank](#) beschrieben ausführt, Zugriffsberechtigungen pro Objektklasse erteilt wurden und auch die Anwender-Mandantenkonzepte den Zugriff gemäß der Beschreibung im Abschnitt [Zugriff über Mandanten steuern](#) erlauben, kann der Schreibzugriff auf Objekte möglicherweise aufgrund der Zugriffsberechtigungen pro Objekt dennoch abgelehnt werden. Die Zugriffsberechtigungskonzepte für einen Anwender, die in Ihrer Alfabet -Lösung implementiert sind, werden ebenfalls für den Zugriff auf Objekte über die REST-API-Serviceaufrufe ausgewertet.



Detaillierte Informationen über die in Alfabet verfügbaren Zugriffsberechtigungskonzepte sowie die Konfigurationen, die zu ihrer Implementierung erforderlich sind, finden Sie unter *Konfigurieren von Zugriffsberechtigungen für Alfabet* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.

Folgendes muss berücksichtigt werden:

- Anwenderprofile können dazu konfiguriert werden, entweder Lese/Schreib-Zugriffsberechtigungen oder nur Lesezugriffsberechtigungen für Alfabet -Objekte zu gewähren. Die Konfiguration des für das Alfabet RESTful API verwendeten Anwenderprofils muss so konfiguriert werden, dass die erforderlichen Zugriffsrechte gewährt sind.
- Um Objektdaten über einen REST-API-Serviceaufruf zu aktualisieren oder zu löschen, muss der Anwender über Schreibzugriffsberechtigungen für ein Objekt besitzen, die beispielsweise auf dem Konzept des autorisierten Anwenders basieren.

Üblicherweise wird ein REST-API-spezifischer Anwender für Serviceaufrufe an das Alfabet RESTful API verwendet. Dieser Anwender wird nicht zum Erstellen und Aktualisieren von Objekten in der Alfabet -Benutzeroberfläche verwendet. Daher sind Lesezugriffsberechtigungen, die beispielsweise auf der Zuweisung als autorisierter Anwender oder auf der Verfügbarkeit offener Aufgaben für das Objekt basieren, im Kontext des Alfabet RESTful API nicht relevant. Um dem REST-API-spezifizierten Anwender Zugriff auf Objekte zu gewähren, können Sie z. B. den Anwender relevanten autorisierten Anwendergruppen zuordnen oder eine Berechtigungsregel definieren, die dem

spezifischen Anwender Schreibzugriff auf Objekte gewährt, wenn er mit dem spezifischen Anwenderprofil angemeldet ist.

Erforderliche Implementierung auf Clientseite

Die RESTful-Serviceschnittstelle auf Clientseite muss so implementiert werden, dass vor der Übermittlung einer Datenanfrage eine Anfrage für den Autorisierungscode gesendet wird. Standardmäßig kann der Autorisierungscode, der in der Antwort der Autorisierungsanfrage enthalten ist, in Aufrufen zum Alfabet RESTful API für die Datenanfrage verwendet werden, die innerhalb von 20 Minuten nach Erhalt des Autorisierungs-codes übermittelt werden. Der Zeitraum für den Anwender kann im **Anwender**-Editor geändert werden.



Weitere Informationen zum Ändern der Gültigkeitsdauer des Autorisierungs-codes finden Sie unter [Erforderliche Konfiguration auf Serverseite](#).

Der Autorisierungsaufruf muss an folgende URL gesendet werden:

```
URLofTheAlfabetWebApplication/api/token
```



Bei der Angabe der URL muss die Groß-/Kleinschreibung beachtet werden.

Für den Aufruf wird die Methode **POST** verwendet.

Das Feld `Content-Type` des HTTP-Headers muss wie folgt definiert werden:

```
Content-Type: application/x-www-form-urlencoded
```

Die Nutzdaten der Anfrage müssen wie folgt definiert sein:

```
grant_type=password&username=Alfabet&password=AFPUVOIM462LG3EOMY3A6G4T4WNATJ  
I3
```

Die Antwort ist eine JSON-Antwort mit folgender Struktur:

```
{  
  "token_type": "bearer"  
  "access_token": "adf93nfpornpor"  
  "expires_in": 1200  
}
```

Das Feld `expires_in` gibt die Zeit, in der das Token gültig ist, in Sekunden an.

Die Aufrufe an die Alfabet RESTful-API müssen die folgende URL mit der Spezifikation des Serviceaufrufs verwenden, wie in den folgenden Kapiteln beschrieben:

```
URLofTheAlfabetWebApplication/api/v2
```



Bei der Angabe der URL muss die Groß-/Kleinschreibung beachtet werden.

Die Daten in der Antwort des Autorisierungsaufrufs müssen als Autorisierungsschlüssel dem Header aller Anfragen hinzugefügt werden, die an das Alfabet RESTful API gesendet werden, und zwar in folgendem Format:

```
Authorization: TypWert
```

mit

Typ = Wert im Feld `token_type`

Wert = Wert im Feld `access_token`



Für die Antwort oben würde die Autorisierung wie folgt aussehen:

```
Authorization: bearer adf93nfporesopor
```

Der Autorisierungsschlüssel läuft nach ca. 20 Minuten ab, wenn das für den Anwender, der für die Generierung des Schlüssels verwendet wird, im **Anwender** -Editor nicht anders konfiguriert ist. Die Clientschnittstelle sollte so konfiguriert sein, dass in regelmäßigen Intervallen oder vor jeder Anfrage ein neuer Autorisierungsschlüssel angefordert wird.

Übersicht über für jeden Endpunkt erforderliche Zugriffsberechtigungen

Um maximale Sicherheit für alle Anwendungsfälle zu ermöglichen, ist die Konfiguration der Zugriffsberechtigungen für die Alfabet RESTful Services ein komplexer Prozess. Um die Konfiguration für einen bestimmten RESTful-Service zu vereinfachen, der auf das Alfabet RESTful API verweist, enthält die folgende Tabelle eine Auflistung aller verfügbaren Arten von Serviceaufrufen sowie der Zugriffsberechtigungen, die konfiguriert werden müssen.

Funktionalität	Endpunkt	API-Zugriffsoptionen für den benannten Anwender, bei dem die Option "Hat Zugriff auf API V2" aktiviert ist	Attribut, das in den relevanten Klasseinstellungen der Objektklasse aktiviert werden muss	Zusätzliche anwendbare Berechtigungskonzepte
Exportieren von Informationen für das gesamte Alfabet -Klassenmodell, einschließlich Aufzählungen und Landeseinstellungen	metamodel	Hat Zugriff auf Metamodel		

Funktionalität	Endpunkt	API-Zugriffsoptionen für den benannten Anwender, bei dem die Option "Hat Zugriff auf API V2" aktiviert ist	Attribut, das in den relevanten Klaseinstellungen der Objekt-klasse aktiviert werden muss	Zusätzliche anwendbare Berechtigungskonzepte
Exportieren von Informationen zu allen oder mehreren Klassen des Alfabet -Klassenmodells	classes	Hat Zugriff auf Metamodell		
Exportieren von Informationen zu allen oder mehreren Aufzählungen im Alfabet -Klassenmodell	enums	Hat Zugriff auf Metamodell		
Exportieren von Daten zu Objekten ohne definierte REFSTR-Werte	objects	Hat Zugriff auf GetObjectsByRefs (Objekte durch Referenzen erhalten)	Lesen über REST-API zulassen	Mandanteneinstellungen für Anwender und Objekt
Exportieren von Informationen zu Objekten, die von einem konfigurierten Bericht gefunden werden	objects	Hat Zugriff auf GetObjectsByReport (Objekte durch Bericht erhalten)		Mandanteneinstellungen für Anwender und Objekt REST-API-Zugriffsberechtigungen für die Anwender- und Ausführungsberechtigung für den konfigurierten Bericht
Daten über Objekte einer definierten Objekt-klasse exportieren, die einer Filterdefinition entsprechen	objects	Hat den GetObjectsByFilter-Zugriff	Lesen über REST-API zulassen	Mandanteneinstellungen für Anwender und Objekt

Funktionalität	Endpoint	API-Zugriffsoptionen für den benannten Anwender, bei dem die Option "Hat Zugriff auf API V2" aktiviert ist	Attribut, das in den relevanten Klasseinstellungen der Objektklasse aktiviert werden muss	Zusätzliche anwendbare Berechtigungskonzepte
Löschen von Objekten aus der Alfabet-Datenbank	delete	Hat Zugriff auf DeleteObjects (Objekte löschen)	Schreiben über REST-API zulassen	Zugriffsberechtigung des Anwenders für Objekt
Archivieren von Objekten aus der Alfabet-Datenbank	Archiv	Hat Zugriff auf GetObjectsByRefs (Objekte durch Referenzen erhalten) (für alle Aufrufe) Hat Zugriff auf DeleteObjects (Objekte löschen) (nur wenn die Objekte nach der Archivierung gelöscht werden sollen)	Lesen über REST-API zulassen (für alle Aufrufe) Schreiben über REST-API zulassen (nur wenn die Objekte nach der Archivierung gelöscht werden sollen)	Zugriffsberechtigung des Anwenders für Objekt
Erzeugen eines neuen Objekts in der Alfabet-Datenbank	update	Hat Zugriff auf CreateObjects (Objekte erstellen)	Schreiben über REST-API zulassen	Zugriffsberechtigung des Anwenders für Objekt
Ändern der Eigenschaften eines vorhandenen Objekts in der Alfabet-Datenbank	update	Hat Zugriff auf UpdateObjects (Objekte aktualisieren)	Schreiben über REST-API zulassen	Zugriffsberechtigung des Anwenders für Objekt
ADIF-Import über einen RESTful-Serviceaufruf starten	adifimport	Hat den ADIFAPIInvocation-Zugriff		REST-API-Ausführungsberechtigung für ADIF-Importschemata Beim ADIF-Import aus einer in der internen Dokumentenauswahl befindlichen Datei: Zugriffsberechtigungen

Funktionalität	Endpunkt	API-Zugriffsoptionen für den benannten Anwender, bei dem die Option "Hat Zugriff auf API V2" aktiviert ist	Attribut, das in den relevanten Klasseneinstellungen der Objektklasse aktiviert werden muss	Zusätzliche anwendbare Berechtigungskonzepte
				Elemente öffnen auf den IDOC-Ordner und Dateierweiterungsberechtigung über Blacklist und Whitelist
ADIF-Export über einen RESTful-Serviceaufruf starten	adifexport	Hat den ADIFAPIInvocation-Zugriff		REST-API-Ausführungsberechtigung für ADIF-Exportschemata Beim ADIF-Export in eine in der internen Dokumentenauswahl befindliche Datei: Zugriffsberechtigungen Elemente verwalten auf den IDOC-Ordner und Dateierweiterungsberechtigung über Blacklist und Whitelist
Starten eines Workflows über einen RESTful-Serviceaufruf	workflow	Hat den WorkflowAPIInvocation-Zugriff		REST-API-Ausführungsberechtigung für Workflow-Vorlagen
Erneutes Generieren des Passworts eines Alfabet-Anwenders	regeneratepassword	Hat PasswordAPI-Zugriff Hat Zugriff auf UpdateObjects (Objekte aktualisieren)	Lesen über Rest-API für die Objektklasse zulassen Person Schreiben über Rest-API für die Objektklasse zulassen Person	
Anonymisieren von Daten für bestimmte Anwender	anonymizeuser	Hat den AnonymizeUser-Zugriff		Die Anonymisierung muss für die Objektklasse "Person" aktiviert werden.

Funktionalität	Endpunkt	API-Zugriffsoptionen für den benannten Anwender, bei dem die Option "Hat Zugriff auf API V2" aktiviert ist	Attribut, das in den relevanten Klasseinstellungen der Objektklasse aktiviert werden muss	Zusätzliche anwendbare Berechtigungskonzepte
Exportieren von Informationen über den Inhalt der internen Dokumentenauswahl	idocfilelist	Hat Zugriff auf IDOC-Dateiliste		Zugriffsberechtigungen Listenelemente auf den IDOC-Ordner und Dateierweiterungsberechtigung über Blacklist und Whitelist
Hochladen von Dokumenten in die interne Dokumentenauswahl	idocupload	Hat IDOC-Zugriff zum Hochladen		Zugriffsberechtigungen Elemente verwalten auf den IDOC-Ordner und Dateierweiterungsberechtigung über Blacklist und Whitelist
Herunterladen von Dokumenten aus der internen Dokumentenauswahl	idocdownload	Hat IDOC-Zugriff zum Herunterladen		Zugriffsberechtigungen Elemente öffnen auf den IDOC-Ordner und Dateierweiterungsberechtigung über Blacklist und Whitelist
Es wird geprüft, ob die Alfabet-Komponenten ausgeführt werden	monitor	Hat MonitoringAPI-Zugriff		
Aktualisierung des Metamodells	updateMM	Hat Zugriff auf Metamodell-Aktualisierung		

Kapitel 5: Konfigurieren der Handhabung von Datums-, Uhrzeit- und Zahlenformaten für API-Aufrufe

Datums- und Uhrzeitformate müssen standardmäßig definiert werden und werden in den durch die Spracheinstellungen definierten Standardformaten zurückgegeben:

Die Alfabet-Webapplikation kann so konfiguriert werden, dass sowohl Informationen zu Datum und Uhrzeit in anderen Formaten zurückgegeben und sogar andere Datums- und Uhrzeitformate in Anfragen akzeptiert werden. Unterschiedliche Sätze von Datums- und Uhrzeitformaten können für gleichzeitige Verwendung durch verschiedene Anwendungen konfiguriert werden, die alle Anfragen an das Alfabet RESTful API senden und ein anderes Datums-/Uhrzeitformat verwenden.



Bitte beachten Sie jedoch, dass für Anfragen an den Endpunkt `objects`, die Ergebnisse von einem konfigurierten Bericht mit Parametern anfragen, die Parametereinstellungen im Feld **ReportArgs** in den oben angegebenen Standardformaten definiert werden müssen, unabhängig von dem Datums-/Uhrzeitformat, das für die Anfrage spezifiziert wurde.

Um ein Datums- und Uhrzeitformat für das Alfabet RESTful API zu definieren, muss eine **API-Landeseinstellung** für die Alfabet -Lösung mit dem Tool Alfabet Expand definiert werden. Die **API-Landeseinstellung** muss anschließend in dem Serviceaufruf referenziert werden, um die definierten Datums-, Datum/Uhrzeit- und Uhrzeitformate dieser API-Landeseinstellung zu verwenden:

- [Konfigurieren einer neuen API-Landeseinstellung](#)
- [Eine API-Landeseinstellung in einem Serviceaufruf verwenden](#)

Konfigurieren einer neuen API-Landeseinstellung

Datums- und Uhrzeitdefinitionen für die Alfabet RESTful API-Serviceaufrufe werden im Metamodell-Explorer des Tools Alfabet Expand definiert.



Informationen über das Zugreifen auf und das Arbeiten mit Alfabet Expand finden Sie im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.



Beachten Sie folgende Syntax für die Spezifikation von Zeit- und Datumsformaten.

- Nutzen Sie zur Spezifikation des 24-Stunden-Zeitsystems zur Angabe von Stunden "H" oder "HH".
 - Nutzen Sie zur Spezifikation des 12-Stunden-Zeitsystems zur Angabe von Stunden "h" oder "hh". Diese Einstellung erfordert eine weitere Angabe zu "AM" oder "PM" für die Schreibweise der Uhrzeit.
 - Nutzen Sie "M" für Monat und "m" für Minute.
- 1) Wechseln Sie in Alfabet Expand zur Registerkarte Metamodell.
 - 2) Klicken Sie im Explorer mit der rechten Maustaste auf den Knoten **API-Landeseinstellung** und wählen Sie **Neue API-Landeseinstellung hinzufügen** aus. Ein neues API-

Landeseinstellungselement wird dem Explorer hinzugefügt, und die Attribute der neuen Elemente werden im Attributfenster angezeigt.

- 3) Ändern Sie im Attribut **Name der API-Landeseinstellung** den Standardnamen des API-Landeseinstellungselements in einen eindeutigen und aussagekräftigen Namen. Der Name des API-Landeseinstellungselements wird zum Referenzieren der API-Landeseinstellung in Serviceaufrufen an das Alfabet RESTful API verwendet.
- 4) Wählen Sie eine der folgenden Optionen aus der Dropdown-Liste im Feld **Datumsformat** aus, um Datums- sowie Datum/Uhrzeitformate zu definieren:
 - **Default:** Die Standardmuster yyyy-MM-dd für das Datum und yyyy-MM-dd HH:mm:ss.fff für Datum und Uhrzeit werden verwendet.
 - **Pattern:** Ein benutzerdefiniertes Muster wird verwendet. Wenn Sie diese Option auswählen, werden zwei neue Attribute **Datumsmuster** und **Datum-/Zeitmuster** angezeigt. Schreiben Sie das erforderliche Datumsmuster für reine Datumswerte in das Attribut **Datumsmuster** und das erforderliche Datum-/Zeitmuster in das Attribut **Datum-/Zeitmuster**.
 - **Posix:** Daten und Datums-/Uhrzeitwerte werden im Posix-Format angenommen und zurückgegeben.
 - **Windows:** Die Windows-Muster MM/dd/yyyy für das Datum und MM/dd/yyyy HH:mm für Datum und Uhrzeit werden verwendet.
- 5) Wählen Sie eine der folgenden Optionen aus der Dropdown-Liste im Feld **Zahlenformat** aus, um Zahlenformate zu definieren:
 - **Default:** Das Standardzahlenformat wird verwendet. Das Dezimalzeichen ist ein Punkt, die Anzahl der Dezimalstellen ist nicht beschränkt, und das Gruppierungszeichen ist ein Komma (z. B. 3,125.987).
 - **Custom:** Ein benutzerdefiniertes Muster wird verwendet. Wenn Sie diese Option auswählen, werden die drei neuen Attribute **Zahlengruppierungszeichen**, **Anzahl der Dezimalziffern** und **Dezimalzahlensymbol** angezeigt. Definieren Sie das erforderliche Zahlenformat in den neuen Feldern. Wenn ein Feld nicht definiert ist, wird für diese Einstellung der Standardwert verwendet.
- 6) Wählen Sie eine der folgenden Optionen aus der Dropdown-Liste im Feld **Uhrzeitformat** aus, um Uhrzeitformate zu definieren:
 - **Default:** Das Standarduhrzeitmuster HH:mm:ss wird verwendet.
 - **Pattern:** Ein benutzerdefiniertes Muster wird verwendet. Wenn Sie diese Option wählen, wird ein neues Attribut **Uhrzeitmuster** angezeigt. Geben Sie das gewünschte Uhrzeitmuster in das Feld ein.
- 7) Klicken Sie zum Speichern Ihrer Änderungen in der Symbolleiste auf die Schaltfläche **Speichern**



Eine API-Landeseinstellung in einem Serviceaufruf verwenden

Für die Alfabet -Lösung definierte API-Landeseinstellungen können in Serviceaufrufen an die Endpunkte `objects`, `delete` und `update` verwendet werden. Um eine API-Landeseinstellung in einem Serviceaufruf

zu verwenden, muss dem JSON-Objekt in den Nutzdaten des Aufrufs ein Feld `ApiCulture` hinzugefügt werden. Der Feldwert muss den Namen einer definierten API-Landeseinstellung angeben.



Beispiel: Die folgende Anfrage für Daten über Objekte der definierten REFSTR-Werte gibt Ergebnisse mit Daten zurück, die im Posix-Format definiert sind:

```
{
  "ApiCulture": "ApiCulture_Posix",
  "EmptyValues": true,
  "Refs": ["76-3472-0", "76-3473-0", "76-3474-0"] }
}
```

Dies führt zu einem Rückgabewert mit Posix-Datumsformaten:

```
{
  "Objects": [
    {
      "ClassName": "Application",
      "RefStr": "76-3473-0",
      "Values": {
        "applicationgroups": null,
        "baseapplicationid": null,
        "color": null,
        "creation_date": " 1504696528 ",
        "creation_user": "ALFABET", "description": null,
        ...
      }
    }
  ]
}
```

Kapitel 6: Serviceaufrufe und Rückgabewerte

Über Serviceaufrufe werden Informationen darüber versendet, welche Aktion in der Alfabet -Datenbank ausgeführt werden muss oder welche Informationen zurück zur URL der Alfabet RESTful API einer Alfabet-Webapplikation gesendet werden müssen. Die für den Zugriff auf die Daten und deren Bearbeitung verfügbaren Endpunkte sind vordefiniert und erfordern bestimmte Parameter, die an die Alfabet RESTful API übermittelt werden.

Um einen Serviceaufruf durchführen zu können, muss eine gültige URL, bestehend aus der Adresse der Alfabet RESTful API und dem Anfrage-Endpunkt, über einen HTTP-Client an den Web-Server gesendet werden. Die Kopfzeile der Anfrage muss eine gültige Autorisierung enthalten, wie im Kapitel oben beschrieben. Bei einigen Endpunkten müssen die für die Ausführung der Anfrage erforderlichen Parameter im JSON-Format in den Nutzdaten der Anfrage übermittelt werden.

Eine gültige URL hat folgende Struktur:

ServerAdress/api/v2/EndpointName/?Parameters

mit:

Variable in URL	Erforderlicher Wert
Serveradresse	Die Webadresse der Alfabet-Webapplikation. Bei der Angabe des Webadresse muss die Groß-/Kleinschreibung beachtet werden.
api/v2	Die Angabe der Alfabet RESTful API. Die aktuelle Version ist <code>api/v2</code> .
EndPoint-Name	Der aufzurufende Endpunkt. Zulässige Werte: <ul style="list-style-type: none"> <code>metamodel</code> für Aufrufe, die Informationen zu allen Klassen und Landeseinstellungen im Alfabet -Metamodell zurückgeben. Weiterführende Informationen finden Sie unter Exportieren von Informationen für das gesamte Alfabet -Klassenmodell, einschließlich Aufzählungen und Landeseinstellungen. <code>classes</code> für Aufrufe, die Informationen zu allen Klassen oder einer Teilmenge der Klassen im Alfabet -Metamodell zurückgeben. <code>enums</code> für Aufrufe, die Informationen zu allen Aufzählungen oder einer Teilmenge der Aufzählungen im Alfabet -Metamodell zurückgeben <code>objects</code> für Aufrufe, die Eigenschaftswerte für ein oder mehrere definierte Objekte in der Alfabet-Datenbank oder das Ergebnis einer Abfrage zurückgeben, die über einen konfigurierten Alfabet -Bericht ausgeführt wurde. <code>delete</code> zum Löschen von Objekten und Beziehungen aus der Alfabet-Datenbank. <code>archiveobject</code> zum Exportieren von Informationen über ausgewählte Objekte in der Alfabet-Datenbank in eine ZIP-Datei, die die relevantesten Ansichtsseiten

Variable in URL	Erforderlicher Wert
	<p>für die Objekte im HTML-Format enthält. Der Endpunkt kann die Objekte nach der Archivierung optional aus der Alfabet-Datenbank löschen.</p> <ul style="list-style-type: none"> • <code>update</code> zum Aktualisieren von Daten in der Alfabet-Datenbank und zum Erstellen von neuen Objekten und Beziehungen. • <code>regeneratepassword</code> zum erneuten Generieren des Passworts der ausgewählten Anwender in der Alfabet-Datenbank. • <code>anonymizeuser</code> zur Anonymisierung der Daten eines oder mehrerer ausgewählter Anwender in der Alfabet-Datenbank. • <code>workflow</code> zum Starten eines Workflows basierend auf einer vorhandenen Workflow-Vorlage in der Alfabet-Datenbank. • <code>adifimport</code> zum Starten eines ADIF-Jobs, der auf einem ADIF-Importschema in der Alfabet-Datenbank basiert. • <code>adifexport</code> zum Starten eines ADIF-Jobs, der auf einem ADIF-Exportschema in der Alfabet-Datenbank basiert. • <code>monitor</code> zur Überprüfung, ob die Alfabet-Komponenten, die an der Ausführung einer RESTful-Serviceanfrage beteiligt sind, ausgeführt werden und ob auf sie zugegriffen werden kann.
Parameter:	<p>Die Angabe der Parameter gilt nur für die Methoden <code>metamodel</code>, <code>classes</code> und <code>enums</code>.</p> <p>Ein Parameter ist definiert als:</p> <pre>ParameterName=ParameterValue</pre> <p>Wenn mehrere Parameter definiert sind, werden sie mit dem Zeichen "&" miteinander verkettet:</p> <pre>ParameterName1=ParameterValue1&ParameterName2=ParameterValue2</pre>

Bei Anfragen sämtlicher Endpunkte außer `metamodel`, `classes` und `enums` werden die Parameter für die Anfrage in den Nutzdaten der Anfrage im JSON-Format definiert. In den folgenden Kapiteln erhalten Sie für jede Anfrage ausführliche Informationen zu den Aufrufen.

Jeder EndPoint stellt Daten im JSON-Format bereit. Wenn die Antwort Daten des Typs „Datum“ enthält, bestehen diese Daten standardmäßig aus Strings im Format JJJJ-MM-TT. Das Datumsformat kann mit einer Parametereinstellung in den Nutzdaten des Aufrufs verändert werden. Wenn die Antwort Namen von Objektklasseneigenschaften zurückgibt, sind diese in Kleinschreibung angegeben.

Die Endpunkte werden in den folgenden Abschnitten ausführlich erläutert. Sie erhalten darin auch Informationen über die verwendete HTTP-Methode sowie über die Struktur des Serviceaufrufs und den Rückgabewert:

- [Exportieren von Informationen für das gesamte Alfabet -Klassenmodell, einschließlich Aufzählungen und Landeseinstellungen](#)

- [Exportieren von Informationen zu allen oder mehreren Klassen des Alfabet -Klassenmodells](#)
- [Exportieren von Informationen zu allen oder mehreren Aufzählungen im Alfabet -Klassenmodell](#)
- [Exportieren von Informationen zu in der Alfabet-Datenbank gespeicherten Objektdaten](#)
 - [Exportieren von Daten zu Objekten ohne definierte REFSTR-Werte](#)
 - [Daten über Objekte einer definierten Objektklasse exportieren, die einer Filterdefinition entsprechen](#)
 - [Exportieren von Informationen zu Objekten, die von einem konfigurierten Bericht gefunden werden](#)
 - [Konfigurieren eines Berichts, der in einer RESTful-Serviceanfrage aufgerufen werden kann](#)
 - [Serviceaufruf zur direkten Ausführung eines konfigurierten Berichts](#)
 - [Serviceaufrufe zum Ausführen offline ausgeführter Berichte](#)
 - [Rückgabewert für die asynchrone ReportResultDataSet Berichtsausführung, die die im Bericht definierten Daten zurückgibt](#)
 - [Rückgabewert für die ReportResult-Objects, der alle Eigenschaften der im Bericht gefundenen Objekte zurückgibt](#)
 - [Rückgabewert für das Berichtsergebnis, der den REFSTR der im Bericht gefundenen Objekte zurückgibt](#)
- [Löschen von Objekten aus der Alfabet-Datenbank](#)
- [Erstellen und Aktualisieren von Objektdaten in der Alfabet-Datenbank](#)
 - [Erzeugen eines neuen Objekts in der Alfabet-Datenbank](#)
 - [Ändern der Eigenschaften eines vorhandenen Objekts in der Alfabet-Datenbank](#)
 - [Erstellen oder Aktualisieren einer Beziehung zwischen Objekten in der Alfabet-Datenbank](#)
- [Archivieren von Objekten aus der Alfabet-Datenbank](#)
- [Erneutes Generieren des Passworts eines Alfabet -Anwenders](#)
- [Anonymisieren von Daten für bestimmte Anwender](#)
- [Starten eines Workflows über einen RESTful-Serviceaufruf](#)
- [ADIF-Import über einen RESTful-Serviceaufruf starten](#)
 - [Auslösen des ADIF-Imports aus einer externen Datenbank oder einem Dokument in der Alfabet-Datenbank](#)
 - [Auslösen des ADIF-Imports aus einem Dateistrom im Serviceaufruf](#)
 - [Überprüfen des Ergebnisstatus der ADIF-Ausführung](#)
 - [Herunterladen der Protokolldatei zur ADIF-Ausführung](#)
- [ADIF-Export über einen RESTful-Serviceaufruf starten](#)

- [Voraussetzungen für das Ausführen eines ADIF-Exports](#)
- [Synchrone Ausführung eines ADIF-Exports](#)
- [Asynchrone Ausführung eines ADIF-Exports](#)
 - [Auslösen der asynchronen Ausführung eines ADIF-Exports](#)
 - [Überprüfen des Ergebnisstatus der ADIF-Ausführung](#)
 - [Herunterladen der Protokolldatei zur ADIF-Ausführung](#)
- [Exportieren von Informationen über den Inhalt der internen Dokumentenauswahl](#)
- [Herunterladen von Dokumenten aus der internen Dokumentenauswahl](#)
- [Hochladen von Dokumenten in die interne Dokumentenauswahl](#)
- [Es wird geprüft, ob die Alfabet-Komponenten ausgeführt werden](#)
- [Aktualisierung des Metamodells](#)

Exportieren von Informationen für das gesamte Alfabet -Klassenmodell, einschließlich Aufzählungen und Landeseinstellungen

Der Endpunkt `metamodel` gibt Informationen zur Definition aller Alfabet -Objektklassen, Aufzählungen und Landeseinstellungen im Metamodell zurück.



Ausführliche Informationen zur Struktur des Alfabet -Metamodells und zur Speicherung der Daten in den Alfabet -Datenbanktabellen finden Sie im Kapitel *Das Alfabet -Metamodell in der Alfabet -Datenbank* im Referenzhandbuch *Alfabet-Datenintegrationsframework* oder im Referenzhandbuch *Das Alfabet-Metamodell*.

Endpunkt-Name : `metamodel`

HTTP-Methode : `GET`

Serviceaufruf:

`ServerAdress/api/v2/metamodel/?emptyValues=true`

mit den folgenden Parametern:

Parameter	Obligatorisch/Optional	Erforderlicher Wert
<code>emptyValues=true/false</code>	Optional	Wenn <code>emptyValues=true</code> zum Serviceaufruf hinzugefügt wird, werden alle relevanten Attribute der Objektklassen, Aufzählungen und Landeseinstellungen zurückgegeben, auch wenn sie gar nicht festgelegt sind. Wenn <code>emptyValues=false</code> hinzugefügt wird, werden nur festgelegte

Parameter	Obligato- risch/Op- tional	Erforderlicher Wert
		Attribute zurückgegeben. Standardmäßig ist für das Attribut <code>false</code> ausgewählt.

Kopfzeilenfelder:

`Authorization: TypWert`



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

Rückgabewert:

Der Rückgabewert ist ein JSON-Objekt mit vier Feldern:

- `alias`: Der Name des Server-Alias der Alfabet-Webapplikation, die auf die Anfrage antwortet
- `cultures`: Eine JSON-Liste der im Alfabet -Metamodell definierten Landeseinstellungen
- `classes`: Eine JSON-Liste der im Alfabet -Metamodell definierten Klassen
- `enums`: Eine JSON-Liste der im Alfabet -Metamodell definierten Aufzählungen



```
{
  "alias": "Alfabet"
  -"cultures": [...]
  -"classes": [...]
  -"enums": [...]
}
```

Die Felder `cultures`, `classes` und `enums` bestehen alle aus einer JSON-Liste von Objekten. Jede Objekt-klasse, Aufzählung und Landeseinstellung im Alfabet -Metamodell entspricht einem Objekt in der jeweiligen JSON-Liste. Die Attribute des Metamodellobjekts sind Felder des JSON-Objekts und haben einen Namen, der dem Namen des Attributs entspricht. Die Attribute legen die technischen Daten zu dem Metamodellobjekt fest, das zum Aufbauen der Datenbanktabellen und zum Verarbeiten der Objektdaten innerhalb der Alfabet -Lösung verwendet wird. Die Liste der Attribute ist auf die Attribute beschränkt, die ggf. für externe Anwendungen relevant sind. Beispielsweise werden das Datum der letzten Aktualisierung oder für das Objekt definierte Tags nicht exportiert.

Untergeordnete Objekte, z. B. Aufzählungselemente für Aufzählungen, werden als ein Feld aufgeführt, das die untergeordneten Objekte in Form einer JSON-Objektliste enthält.

Folgende Informationen sind optional:

Objekttyp im Alfabet - Metamodell	In der Antwort enthaltene Informationen	Beispiel
Landeseinstellungen	<p>Jede Landeseinstellung im Alfabet -Metamodell ist ein JSON-Objekt in der JSON-Liste des Felds <code>cultures</code>.</p> <p>Jedes für die Landeseinstellung im Alfabet -Metamodell definierte relevante Attribut ist als ein Feld im JSON-Objekt definiert. Die Feldnamen ähneln den Namen der Attribute.</p>	<pre>"cultures": [7] 0: { "BaseCulture": 1031 "DateFormatCulture": 1031 "NumberFormatCulture": 1031 "HelpCulture": 1031 "IsDefault": false "IsInstanceTranslation": true "Picture": "Flag_German" "TimePattern": "HH:mm:ss" "MeasurementUnit": "Metric" } ...</pre>
Objektclassen	<p>Jede Objektclass im Alfabet -Metamodell, das im Explorer Klassen auf der Registerkarte Metamodell des Konfigurationstools Alfabet Expand angezeigt wird, ist ein JSON-Objekt in der JSON-Liste des Felds <code>classes</code>.</p> <p>Die Informationen zu Objektclassen umfassen folgende Elemente:</p> <ul style="list-style-type: none"> Attribute der Objektclass <p>Die für die Objektclass im Alfabet -Metamodell definierten relevanten Attribute werden jeweils als ein Feld im JSON-Objekt definiert. Die Feldnamen ähneln den Namen der Attribute.</p> <p>Hinweis: Objektclassen, für die das Attribut Automatisch verwaltet auf <code>true</code> gesetzt ist, sind Objektclassen, die nur durch Mechanismen geändert</p> 	<pre>"classes": [266] 0: { "Name": "Domain" "Id": 359 "Comment": "" "Caption": "Domain" "AutomaticallyManaged": false "TechName": "DOMAIN" "Audit": true "HasMandates": true "Hint": "" "IDPrefix": "DOM" "Stereotypes": [5] 0: { "Name": "Area" "Caption": "Area" "CaptionPlural": "Areas" "comment": ""</pre>

Objekttyp im Alfabet - Metamodell	In der Antwort enthaltene Informationen	Beispiel
	<p>werden dürfen, die von den Komponenten der Alfabet - Software angestoßen werden. Sie dürfen nicht von Komponenten von Drittanbietern geändert werden.</p> <ul style="list-style-type: none"> Objektklasseneigenschaftswerte <p>Für jede Alfabet -Objekt-klasse gibt es in der Alfabet-Datenbank eine Datenbanktabelle. Jede Objektklasseneigenschaft entspricht einer Spalte in der Datenbanktabelle. Das heißt, dass in den Objektklasseneigenschaften die Informationen zu den einzelnen Objekten in der Klasse gespeichert sind. Alle Objektklasseneigenschaften, die im Explorer Klassen auf der Registerkarte Metamodell des Konfigurationstools Alfabet Expand angezeigt werden, werden in das Feld <code>Properties</code> als JSON-Objektliste aufgenommen, wobei jedes JSON-Objekt eine Eigenschaft und die Felder des JSON-Objekts einen Teil der Attribute der Eigenschaft darstellen, die möglicherweise für externe Prozesse relevant sind.</p> <p>Hinweis: Objektklasseneigenschaften, für die das Attribut Automatisch verwaltet auf <code>true</code> gesetzt ist, sind Eigenschaften, die nur von den Komponenten der Alfabet -Software geändert werden dürfen. Sie dürfen nicht von Komponenten von Drittanbietern geändert werden.</p> 	<pre> "HasMandates": false } ... "Properties": [31] 0: { "Name": "ID" "Guid": "6736AEFBF52C416C94D161F AABAF3D7F" "TechName": "ID" "Caption": "ID" "Comment": "" "Hint": "" "Alias": "" "Type": "String" "AutomaticallyManaged": false "DefaultValue": null "Validator": "" "EnumInfo": "" } ... "Keys": [3] 0: { "Name": "Domain_Key1" "TechName": "C359_K1" "Content": "BelongsTo" "Unique": false "IsCaseIns": false "Descending": false "IsActivated": true } ... } </pre>

Objekttyp im Alfabet - Metamodell	In der Antwort enthaltene Informationen	Beispiel
	<ul style="list-style-type: none"> Objektklassenstereotyp Wenn für die Objektklasse Stereotype konfiguriert sind, werden die Stereotype als JSON-Objektliste in das Feld <code>Stereotypes</code> des Klassenobjekts aufgenommen. Jedes Attribut des Stereotyps wird als ein Feld im JSON-Objekt exportiert. Klassenschlüssel Mit einem Klassenschlüssel, dessen Attribut Eindeutig auf <code>True</code> gesetzt ist, wird eine Objektklasseneigenschaft oder eine Kombination von Objektklasseneigenschaften angegeben, die für diese Klasse eindeutig sein muss. Wenn die Daten nicht die Anforderungen erfüllen, die in der Klassenschlüsseldefinition festgelegt sind, kann das Objekt nicht in der Datenbank erzeugt werden. Das Alfabet -Metamodell lässt auch die Definition von Klassenschlüsseln zu, die keine Eindeutigkeit erfordern. In diesem Fall wird das Klassenschlüsselattribut Eindeutig auf <code>False</code> gesetzt. Zweck eines solchen Klassenschlüssels ist, die Suchfunktionalität zu beschleunigen, indem für jeden Klassenschlüssel ein Index erzeugt wird. Im Rückgabewert ist jeder Klassenschlüssel ein Objekt. Er wird in einem Array von Objekten im Feld <code>unique keys</code> im Objektklassenobjekt aufgelistet. 	

Objekttyp im Alfabet - Metamodell	In der Antwort enthaltene Informationen	Beispiel
Aufzählungen	<p>Jede Aufzählung im Alfabet -Metamodell ist ein JSON-Objekt in der JSON-Liste des Felds <code>enums</code>.</p> <p>Informationen zu Aufzählungen umfassen folgende Elemente:</p> <ul style="list-style-type: none"> Attribute der Aufzählung <p>Jedes für die Aufzählung im Alfabet -Metamodell definierte relevante Attribut ist als ein Feld im JSON-Objekt definiert. Die Feldnamen ähneln den Namen der Attribute.</p> Aufzählungselemente und ihre Attribute <p>Die für die Aufzählung definierten Aufzählungselemente werden in den Feldelementen aufgeführt. Die Feldelemente bestehen aus einer JSON-Liste von Objekten. Jedes Aufzählungselement ist ein Objekt in der Liste, und die Felder des Objekts geben die relevanten Attribute des Aufzählungselements zurück. Die Feldnamen ähneln den Namen der Attribute.</p> 	<pre> "enums": [90] 0: { "Name": "AlfaDocCategory" "Guid": "6E7E549625034B788C5725F254 1BD412" "Comment": "" "Hint": "" "HelpFile": "Enum_AlfaDocCategory.html" "Items": [7] 0: { "Value": "" "Comment": "" "Hint": "" } 1: { "Value": "Manual" "Comment": "" "Hint": "" } ... } </pre>

Exportieren von Informationen zu allen oder mehreren Klassen des Alfabet -Klassenmodells

Der Endpunkt `classes` gibt Informationen zur Definition aller oder einer Teilmenge der Alfabet -Objektclassen im Alfabet -Metamodell zurück.



Ausführliche Informationen zur Struktur des Alfabet -Metamodells und zur Speicherung der Daten in den Alfabet -Datenbanktabellen finden Sie im Kapitel *Das Alfabet -Metamodell in der*

Alfabet -Datenbank im Referenzhandbuch *Alfabet-Datenintegrationsframework* oder im Referenzhandbuch *Das Alfabet-Metamodell*.

Endpunkt-Name : classes

HTTP-Methode : GET

Serviceaufruf:

`ServerAdress/api/v2/classes/?names=Domain,Application&emptyValues=true`

mit den folgenden Parametern:

Parameter	Obligato- risch/Op- tional	Erforderlicher Wert
names= Class- Name, ClassName	Optional	Es muss eine durch Trennzeichen getrennte Liste von Objektklassen für diese Informationen zurückgegeben werden. Die Objektklasse muss vom Wert des Attributs Name der Objektklasse definiert werden. Wenn der Parameter nicht zum Aufruf hinzugefügt wird, werden Informationen zu allen Objektklassen im Alfabet -Metamodell zurückgegeben.
emp- tyValues=true/false	Optional	Wenn <code>emptyValues=true</code> zum Serviceaufruf hinzugefügt wird, werden alle relevanten Attribute der Objektklassen zurückgegeben, auch wenn sie gar nicht festgelegt sind. Wenn <code>emptyValues=false</code> hinzugefügt wird, werden nur festgelegte Attribute zurückgegeben. Standardmäßig ist für das Attribut <code>false</code> ausgewählt.

Kopfzeilenfelder:

`Authorization: TypWert`



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

Rückgabewert:

Der Rückgabewert ist ein JSON-Objekt mit zwei Feldern:

- `alias`: Der Name des Server-Alias der Alfabet-Webapplikation, die auf die Anfrage antwortet
- `classes`: Eine JSON-Liste der im Alfabet -Metamodell definierten Klassen



```
{
  "alias": "Alfabet"
  -"classes": [...]
}
```


Die Feldklassen bestehen aus einer JSON-Liste von Objekten. Jede Objektklasse im Alfabet -Metamodell entspricht einem Objekt in der JSON-Liste. Die Attribute der Metamodell-Objektklasse sind Felder des jeweiligen JSON-Objekts. Der Name des Felds entspricht dem Namen des Attributs. Die Attribute legen die technischen Daten zu dem Metamodellobjekt fest, das zum Aufbauen der Datenbanktabellen und zum Verarbeiten der Objektdaten innerhalb der Alfabet -Lösung verwendet wird. Die Liste der Attribute ist auf die Attribute beschränkt, die ggf. für externe Anwendungen relevant sind. Beispielsweise werden das Datum der letzten Aktualisierung oder für das Objekt definierte Tags nicht exportiert.

Untergeordnete Objekte, z. B. für die Klasse definierte Stereotype, werden als ein Feld aufgeführt, das die untergeordneten Objekte in Form einer JSON-Objektliste enthält.

Folgende Informationen sind optional:

Wenn der Parameter `names` in der Anfrage nicht angegeben ist, ist jede Objektklasse im Alfabet -Metamodell, die im Explorer **Klassen** auf der Registerkarte **Metamodell** des Konfigurationstools Alfabet Expand angezeigt wird, ein JSON-Objekt in der JSON-Liste des Felds `classes`. Die Klassen werden in alphabetischer Reihenfolge angezeigt.

Wenn der Parameter `names` in der Anfrage angegeben ist, enthält die Klassenliste nur die Klassen, die im Parameter `names` definiert sind, und zwar in der im Parameter `names` festgelegten Reihenfolge.

Die Informationen zu Objektklassen umfassen folgende Elemente:

- **Attribute der Objektklasse**

Die für die Objektklasse im Alfabet -Metamodell definierten relevanten Attribute werden jeweils als ein Feld im JSON-Objekt definiert. Die Feldnamen ähneln den Namen der Attribute.

Hinweis: Objektklassen, für die das Attribut **Automatisch verwaltet** auf `true` gesetzt ist, sind Objektklassen, die nur durch Mechanismen geändert werden dürfen, die von den Komponenten der Alfabet -Software angestoßen werden. Sie dürfen nicht von Komponenten von Drittanbietern geändert werden.

```
"classes": [266]
  0: {
    "Name": "Domain"
    "Id": 359
    "Comment": ""
    "Caption": "Domain"
    "AutomaticallyManaged": false
    "TechName": "DOMAIN"
    "Audit": true
    "HasMandates": true
    "Hint": ""
    "IDPrefix": "DOM"
    "Stereotypes": [5]
      0: {
        "Name": "Area"
        "Caption": "Area"
        "CaptionPlural": "Areas"
        "comment": ""
        "HasMandates": false
      }
      ...
    "Properties": [31]
      0: {
        "Name": "ID"
        "Guid":
        "6736AEFBF52C416C94D161FAABAF3
        D7F"
        "TechName": "ID"
        "Caption": "ID"
        "Comment": ""
```

- **Objektklasseneigenschaftswerte**

Für jede Alfabet -Objektklasse gibt es in der Alfabet-Datenbank eine Datenbanktabelle. Jede Objektklasseneigenschaft entspricht einer Spalte in der Datenbanktabelle. Das heißt, dass in den Objektklasseneigenschaften die Informationen zu den einzelnen Objekten in der Klasse gespeichert sind. Alle Objektklasseneigenschaften, die im Explorer **Klassen** auf der Registerkarte **Metamodell** des Konfigurationstools Alfabet Expand angezeigt werden, werden in das Feld `Properties` als JSON-Objektliste aufgenommen, wobei jedes JSON-Objekt eine Eigenschaft und die Felder des JSON-Objekts einen Teil der Attribute der Eigenschaft darstellen, die möglicherweise für externe Prozesse relevant sind.

Hinweis: Objektklasseneigenschaften, für die das Attribut **Automatisch verwaltet** auf `true` gesetzt ist, sind Eigenschaften, die nur von den Komponenten der Alfabet -Software geändert werden dürfen. Sie dürfen nicht von Komponenten von Drittanbietern geändert werden.

- **Objektklassenstereotyp**

Wenn für die Objektklasse Stereotype konfiguriert sind, werden die Stereotype als JSON-Objektliste in das Feld `Stereotypes` des Klassenobjekts aufgenommen. Jedes Attribut des Stereotyps wird als ein Feld im JSON-Objekt exportiert.

- **Klassenschlüssel**

Mit einem Klassenschlüssel, dessen Attribut **Eindeutig** auf `True` gesetzt ist, wird eine

```
"Hint": ""
"Alias": ""
"Type": "String"
"AutomaticallyManaged": false
"DefaultValue": null
"Validator": ""
"EnumInfo": ""
}
...
"Keys": [3]
0: {
  "Name": "Domain_Key1"
  "TechName": "C359_K1"
  "Content": "BelongsTo"
  "Unique": false
  "IsCaseIns": false
  "Descending": false
  "IsActivated": true }
...
}
```

Objektklasseneigenschaft oder eine Kombination von Objektklasseneigenschaften angegeben, die für diese Klasse eindeutig sein muss. Wenn die Daten nicht die Anforderungen erfüllen, die in der Klassenschlüsseldefinition festgelegt sind, kann das Objekt nicht in der Datenbank erzeugt werden. Das Alfabet -Metamodell lässt auch die Definition von Klassenschlüsseln zu, die keine Eindeutigkeit erfordern. In diesem Fall wird das Klassenschlüsselattribut **Eindeutig** auf `False` gesetzt. Zweck eines solchen Klassenschlüssels ist, die Suchfunktionalität zu beschleunigen, indem für jeden Klassenschlüssel ein Index erzeugt wird. Im Rückgabewert ist jeder Klassenschlüssel ein Objekt. Er wird in einem Array von Objekten im Feld `unique keys` im Objektklassenobjekt aufgelistet.

Exportieren von Informationen zu allen oder mehreren Aufzählungen im Alfabet -Klassenmodell

Der Endpunkt `enums` gibt Informationen zur Definition aller Aufzählungen oder einer Teilmenge der Aufzählungen im Alfabet -Metamodell zurück. Aufzählungen definieren einen vordefinierten Satz an Werten, die für eine Objektklasseneigenschaft festgelegt werden dürfen.



Ausführliche Informationen zur Struktur des Alfabet -Metamodells und zur Speicherung der Daten in den Alfabet -Datenbanktabellen finden Sie im Kapitel *Das Alfabet -Metamodell in der Alfabet -Datenbank* im Referenzhandbuch *Alfabet-Datenintegrationsframework* oder im Referenzhandbuch *Das Alfabet-Metamodell*.

Endpunkt-Name : `enums`

HTTP-Methode : `GET`

Serviceaufruf:

```
ServerAdress/api/v2/enums?names=Enum1,Enum2&emptyValues=true
```

mit den folgenden Parametern:

Parameter	Obligato- risch/Op- tional	Erforderlicher Wert
<code>names= EnumName, EnumName</code>	Optional	Es muss eine durch Trennzeichen getrennte Liste von Aufzählungen für diese Informationen zurückgegeben werden. Die Aufzählung muss vom Wert des Attributs Name der Aufzählung definiert werden. Wenn der Parameter nicht zum Aufruf hinzugefügt wird, werden Informationen zu allen Aufzählungen im Alfabet -Metamodell zurückgegeben.
<code>emptyValues=true/false</code>	Optional	Wenn <code>emptyValues=true</code> zum Serviceaufruf hinzugefügt wird, werden alle relevanten Attribute der Aufzählung und Aufzählungselemente zurückgegeben, auch wenn sie gar nicht festgelegt sind. Wenn <code>emptyValues=false</code> hinzugefügt wird, werden nur festgelegte Attribute zurückgegeben. Standardmäßig ist für das Attribut <code>false</code> ausgewählt.

Kopfzeilenfelder:

`Authorization: TypWert`



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

Rückgabewert:

Der Rückgabewert ist ein JSON-Objekt mit zwei Feldern:

- `alias`: Der Name des Server-Alias der Alfabet-Webapplikation, die auf die Anfrage antwortet
- `classes`: Eine JSON-Liste der im Alfabet -Metamodell definierten Klassen



```
{
  "alias": "Alfabet"
  -"enum": [...]
}
```

Die Feldklassen bestehen aus einer JSON-Liste von Objekten. Jede Aufzählung im Alfabet -Metamodell entspricht einem Objekt in der JSON-Liste. Die Attribute der Aufzählung sind Felder des jeweiligen JSON-Objekts. Der Name des Felds entspricht dem Namen des Attributs. Die Attribute legen die technischen Daten zu dem Metamodellobjekt fest, das zum Aufbauen der Datenbanktabellen und zum Verarbeiten der Objektdaten innerhalb der Alfabet -Lösung verwendet wird. Die Liste der Attribute ist auf die Attribute beschränkt, die ggf. für externe Anwendungen relevant sind. Beispielsweise werden das Datum der letzten Aktualisierung oder für das Objekt definierte Tags nicht exportiert.

Untergeordnete Objekte, z. B. für die Aufzählung definierte Aufzählungselemente, werden als ein Feld aufgeführt, das die untergeordneten Objekte in Form einer JSON-Liste von Objekten enthält.

Folgende Informationen sind optional:

Wenn der Parameter `names` in der Anfrage nicht angegeben ist, ist jede Aufzählung im Alfabet -Metamodell, die im Explorer **Klassen** auf der Registerkarte **Metamodell** des Konfigurationstools Alfabet Expand angezeigt wird, ein JSON-Objekt in der JSON-Liste des Felds `enums`. Die Aufzählungen werden in alphabetischer Reihenfolge angezeigt.

Wenn der Parameter `names` in der Anfrage angegeben ist, enthält die Liste der Aufzählungen nur die Aufzählungen, die im Parameter `names` definiert sind, und zwar in der im Parameter `names` festgelegten Reihenfolge.

Die Informationen zu Objektklassen umfassen folgende Elemente:

- **Attribute der Aufzählung**

Jedes für die Aufzählung im Alfabet -Metamodell definierte relevante Attribut ist als ein Feld im JSON-Objekt definiert. Die Feldnamen ähneln den Namen der Attribute.

- **Aufzählungselemente und ihre Attribute**

Die für die Aufzählung definierten Aufzählungselemente werden in den Feldelementen aufgeführt. Die Feldelemente bestehen aus einer JSON-Liste von Objekten. Jedes Aufzählungselement ist ein Objekt in der Liste, und die Felder des Objekts geben die relevanten Attribute des Aufzählungselements zurück. Die Feldnamen ähneln den Namen der Attribute.

```
"enums": [90]
  0: {
    "Name": "AlfaDocCategory"
    "Guid":
      "6E7E549625034B788C5725F2541BD412"
    "Comment": ""
    "Hint": ""
    "HelpFile":
      "Enum_AlfaDocCategory.html"
    "Items": [7]
      0: {
        "Value": ""
        "Comment": ""
        "Hint": ""
      }
      1: {
        "Value": "Manual"
        "Comment": ""
        "Hint": ""
      }
      ...
    }
  }
```

Exportieren von Informationen zu in der Alfabet-Datenbank gespeicherten Objektdaten

Der Endpunkt `objects` gibt die Informationen zurück, die zu einem oder mehreren Objekten in einer Datenbanktabelle der Alfabet-Datenbank gespeichert sind. Zur Auswahl der Objekte, über die Informationen von diesem Endpunkt zurückgegeben werden, gibt es drei Methoden:

- **Objekte nach Referenzen**

Diese Methode setzt voraus, dass der Wert der Eigenschaft `REFSTR` des Objekts bekannt ist. `REFSTR` ist eine eindeutige Kennung für Objekte in der Alfabet-Datenbank. Sie muss in den Nutzdaten der Anfrage definiert werden, damit alle Eigenschaftswerte des Objekts exportiert werden.

- **Objekte nach Filter**

Die Methode gibt alle Eigenschaftswerte zurück, die für alle Objekte einer bestimmten Objektklasse festgelegt sind und den Suchkriterien entsprechen, die im Serviceaufruf definiert sind. Der Suchfilter im Serviceaufruf kann nur Objektklasseneigenschaften suchen, die einen Text zurückgeben, d. h. die Objektklasseneigenschaften des Typs `Text` oder `String`.

- **Objekte nach Bericht**

Die Nutzdaten der Anfrage können auf einen konfigurierten Bericht des Typs `NativeSQL` oder `Query` in der Alfabet-Datenbank zeigen. Der Rückgabewert enthält alle Informationen, die im Datensatz des Berichts oder in einer Teilmenge davon enthalten sind.

Folgende Informationen sind verfügbar:

- [Exportieren von Daten zu Objekten ohne definierte REFSTR-Werte](#)
- [Daten über Objekte einer definierten Objektklasse exportieren, die einer Filterdefinition entsprechen](#)
- [Exportieren von Informationen zu Objekten, die von einem konfigurierten Bericht gefunden werden](#)
 - [Konfigurieren eines Berichts, der in einer RESTful-Serviceanfrage aufgerufen werden kann](#)
 - [Serviceaufruf zur direkten Ausführung eines konfigurierten Berichts](#)
 - [Serviceaufrufe zum Ausführen offline ausgeführter Berichte](#)
 - [Serviceaufruf zum Auslösen einer Offline-Ausführung](#)
 - [Überprüfen des Status einer asynchronen Berichtsausführung](#)
 - [Exportieren von Ergebnissen asynchroner Berichtsausführung](#)
 - [Rückgabewert für die asynchrone ReportResultDataSet Berichtsausführung, die die im Bericht definierten Daten zurückgibt](#)
 - [Rückgabewert für die ReportResult-Objects, der alle Eigenschaften der im Bericht gefundenen Objekte zurückgibt](#)
 - [Rückgabewert für das Berichtsergebnis, der den REFSTR der im Bericht gefundenen Objekte zurückgibt](#)

Exportieren von Daten zu Objekten ohne definierte REFSTR-Werte

Endpunkt-Name : objects

HTTP-Methode : POST

Serviceaufruf:

`ServerAdress/api/v2/objects`

Kopfzeilenfelder:

Authorization: *TypWert*



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

Content-Type: application/json; charset=utf-8

Nutzdaten

Die Nutzdaten liegen als JSON-Objekt mit folgender Struktur vor:

```
{
  "CurrentProfile": "UserProfile1",
  "CurrentMandate": "Mandat1",
  "ApiCulture": "ApiCulture_Posix",
  "Language": 1033,
  "DataCulture": "de-DE",
  "EmptyValues": true,
  "Refs": ["76-2518-0", "76-3246-0", "405-3-0"]
}
```

Sie können folgende Felder enthalten:

Feld	Obligatorisch/Optional	Erforderlicher Wert
"Refs":[" RefstrOfObject ", " RefstrOfObject "]	Obligatorisch	Ein Array mit einem oder mehreren Werten der REFSTR-Eigenschaft der Objekte in der Alfabet-Datenbank. Die Antwort enthält Informationen zu den Objekten, wobei die Objekte durch den jeweiligen REFSTR angegeben sind.
"EmptyValues":true/false	Optional	Wenn EmptyValues auf "true" gesetzt ist, werden alle Objektklasseneigenschaften des Objekts zurückgegeben, auch wenn sie gar nicht festgelegt sind. Wenn EmptyValues auf "false" gesetzt ist, werden nur die Objektklasseneigenschaften des Objekts zurückgegeben, die

Feld	Obligatorisch/Optional	Erforderlicher Wert
		festgelegt sind. Standardmäßig ist für das Attribut True ausgewählt.
"CurrentProfile": " UserProfileName "	Obligatorisch	Der Name eines Anwenderprofils, das dem Alfabet -Anwender, der den Aufruf sendet, zugeordnet ist, welches für den Zugriff auf Alfabet verwendet wird. Das Anwenderprofil ist erforderlich, um auszuwerten, ob der Anwender, der den Serviceaufruf sendet, befugt ist, Daten über ein Objekt zu lesen. Weitere Informationen zu Zugriffsberechtigungen, die von dem Anwenderprofil abhängen, finden Sie im Abschnitt Autorisierung .
"CurrentMandate": " MandateName "	Optional	Dieses Feld ist nur relevant, wenn der Zugriff auf Objekte durch das Mandantenkonzept für föderative Architekturen gesteuert wird, das in Alfabet implementiert ist. Das Feld muss den Namen eines Mandanten angeben, welcher dem Alfabet -Anwender zugeordnet ist, der den Aufruf sendet. Ausführliche Informationen über die Berücksichtigung von Mandanteneinstellungen in REST-API-Serviceaufrufen finden Sie im Abschnitt Zugriff über Mandanten steuern .
"Language": " LanguageLCID "	Optional	<p>Dieses Feld ist nur relevant, wenn in Ihrer Alfabet -Applikation die Instanzübersetzung verwendet wird. Das Feld muss das Sprachencodeximal (Language Code Decimal, LCID) der Landeseinstellung angeben, für die Übersetzungen für den Namen und die Beschreibung des Objekts in der Alfabet-Datenbank vorhanden sind. Standardmäßig wird die Standardsprache der Datenbank zurückgegeben.</p> <p>Hinweis: Es werden keine Werte zurückgegeben, wenn für eine Objektklasseneigenschaft keine Instanzübersetzung verfügbar ist, selbst wenn ein Wert in der Standardsprache bereitgestellt wird.</p> <p>Um alle Werte auf die ursprünglichen Werte zurückzusetzen und Übersetzungen in einer zweiten Sprache hinzuzufügen, sollte der Parameter <code>DataCulture</code> anstatt des Parameters <code>Language</code> verwendet werden. <code>Language</code> löst <code>DataCulture</code> ab. Wenn beide Felder verfügbar sind wird nur <code>Language</code> verwendet.</p>
"DataCulture": " LanguageCode "	Optional	Dieses Feld ist nur relevant, wenn in Ihrer Alfabet -Applikation die Instanzübersetzung verwendet wird. Das Feld muss das Sprachencodeximal (Language Code Decimal, LCID) der Landeseinstellung angeben, für die Übersetzungen für den Namen und die Beschreibung des Objekts in der Alfabet-Datenbank (zum Beispiel <code>de-DE</code> für

Feld	Obligato- risch/Op- tional	Erforderlicher Wert
		<p>Deutsch) vorhanden sind, oder ALL, um Übersetzungen für alle Sprachen zurückzugeben, für die Übersetzungen definiert sind. Zusätzlich zu den Werten in der ursprünglichen Sprache werden dem Rückgabewert in einem separaten Feld <code>Translations</code> Übersetzungen hinzugefügt.</p> <p>Das Feld <code>Language</code> hat Vorrang vor <code>DataCulture</code>. Wenn beide Felder verfügbar sind, wird nur <code>Language</code> verwendet und <code>DataCulture</code> wird ignoriert.</p>
<pre>"ApiCulture": "APICul- tureName"</pre>	Optional	<p>Dieses Feld gibt eine API-Landeseinstellung an, die in Alfabet Expand als Teil der Konfiguration des Alfabet -Metamodells definiert wurde. Die API-Landeseinstellung ermöglicht das Definieren von Daten, Datum/Uhrzeit und Uhrzeitmustern sowie Zahlenformaten, die vom Alfabet -Standard abweichen. Wird dieses Feld der Anfrage hinzugefügt, werden Daten, Uhrzeiten und Zahlen im Rückgabewert in den Formaten geschrieben, die in der Definition der API-Landeseinstellung angegeben sind. Informationen hierzu finden Sie im Abschnitt Konfigurieren der Handhabung von Datums-, Uhrzeit- und Zahlenformaten für API-Aufrufe.</p>
<pre>"SortOrder" : " Ob- jectClassPropertyName "</pre>		<p>Definieren Sie den Namen einer Objektklasseneigenschaft, um die Ergebnisse nach dieser Eigenschaft im Rückgabewert in aufsteigender Reihenfolge alphanumerisch zu sortieren.</p> <p>Standardmäßig werden die Objekte nach aufsteigendem <code>REFSTR</code> sortiert.</p> <p>Wenn Objekte für die ausgewählte Objektklasseneigenschaft keinen Wert aufweisen, werden sie nach aufsteigendem <code>REFSTR</code> sortiert und zuerst aufgeführt.</p> <p>Wenn entweder <code>Language</code> oder <code>DataCulture</code> gesetzt und für die Objektklasseneigenschaft eine Übersetzung aktiviert ist, werden die Werte nach übersetztem Wert sortiert. Wenn für Objektklasseneigenschaftswerte keine Übersetzung angegeben ist, werden diese nach aufsteigendem <code>REFSTR</code> sortiert und zuerst aufgeführt.</p>

Rückgabewert:

Der Rückgabewert ist ein JSON-Objekt mit zwei Feldern:

- `Count`: Es wird die Anzahl der Objekte, die in der Datenbank für diese Daten gefunden wurden, zurückgegeben. Die Anzahl kann sich von der Anzahl der definierten `REFSTR`-Werte in der Anfrage unterscheiden: Wenn ein Objekt nicht gefunden wird, weil der in der Anfrage definierte `REFSTR` nicht in der Alfabet-Datenbank existiert, wird das Objekt bei der Anzahl und den zurückgegebenen Objektdaten nicht berücksichtigt.

- **Objects:** Eine JSON-Liste mit Objekten, wobei jedes Objekt ein Datenbankobjekt darstellt, für das Daten zurückgegeben werden.

```

{
  "Objects": [...],
  "Count": 3
}
    
```

Das Feld `Objects` enthält ein JSON-Objekt pro Datenbankobjekt, das für den Aufruf gefunden wurde. Das Objekt enthält die folgenden Felder:

- **ClassName:** Der Name der Alfabet -Objektklasse, zu der das Objekt gehört.
- **RefStr:** Der Wert des REFSTR des Objekts.
- **Values:** Ein JSON-Objekt, das alle Informationen enthält, die zum Objekt in der Datenbank über die Objektklasseneigenschaften definiert sind. Jede Eigenschaft ist ein Feld mit einem Namen, der dem Eigenschaftsnamen entspricht. Der Wert des Felds gibt Auskunft über den Wert der Eigenschaft. Der Rückgabewert für eine Eigenschaft hängt von dem Werttyp der Eigenschaft ab:

Eigen-schaftstyp	Rückgabewert
String, Text, StringArray	String, Text wird als Zeichenfolge zurückgegeben (in Anführungszeichen): <pre>"description": "This application manages customer relations."</pre>
StringArray	Alle ausgewählten Optionen des String-Arrays werden in einer Zeichenfolge zurückgegeben: <pre>"subcategories": "APP_SecurityAssessment APP_CloseSubworkflow"</pre>
Boolean	Die Boolean-Werte sind 1 für „true“, 0 für „false“ oder null, wenn die Eigenschaft nicht festgelegt ist: <pre>"variant": 1</pre>
Real, Integer	Die Real- und Integer-Werte werden als Zahl zurückgegeben: <pre>"xpos": 485.4</pre>
Date, DateTime	Die Date- und DateTime-Informationen werden als Datumszeichenfolgen im Format JJJJ-MM-TT zurückgegeben: <pre>"last_update": "2012-02-21"</pre>
URL	URLs werden als Zeichenfolge zurückgegeben (in Anführungszeichen):

Eigen-schaftstyp	Rückgabewert
	<code>"url": "http://company.com/ReportServer"</code>
Referenz	<p>Es wird der REFSTR des Objekts, auf den die Referenz verweist, zurückgegeben:</p> <pre>"ictobject": "26-608-0"</pre>
Refer-enceArray	<p>Für Referenz-Arrays werden die REFSTRs aller Objekte, auf die das aktuelle Objekt mit dieser Eigenschaft verweist, in einem Array aufgelistet:</p> <pre>"applicationgroups": [3] 0: "95-38-0" 1: "95-8-0" 2: "95-9-0"</pre> <p>Der Rückgabewert ist unabhängig von der Einstellung des Attributs Referenzunterstützung der Eigenschaft. Wenn das Attribut Referenzunterstützung der Eigenschaft auf <code>false</code> gesetzt ist, wird der REFSTR-Wert der Referenzziele direkt in der Datenbanktabelle der Objektklasse in einer Spalte für die Eigenschaft gespeichert. Ist Referenzunterstützung auf <code>true</code> gesetzt, werden die Referenzen in der Tabelle RELATIONS gespeichert. Dieser Unterschied ist im Rückgabewert für den Endpunkt <code>objects</code> nicht sichtbar.</p> <p>Wenn Informationen zu den Objekten, auf die vom aktuellen Objekt verwiesen wird, in den Rückgabewert aufgenommen werden sollen, können Sie die Methode zum Auffinden von Objekten über einen konfigurierten Bericht verwenden, der ebenfalls für den Endpunkt <code>objects</code> zur Verfügung steht.</p>

- **Translations:** Dieses Feld ist nur verfügbar, wenn das Feld `DataCulture` in den Nutzdaten des Serviceaufrufs definiert ist. Das Feld enthält eine JSON-Objektliste mit einem Objekt pro Landeseinstellung für Daten. Jedes Objekt hat ein Feld `DataCulture`, das die Sprache zurückgibt, für die Übersetzungen zurückgegeben werden, und ein Feld `Values`, das ein JSON-Objekt mit einem Feld für jede übersetzte Eigenschaft ist. Der Wert des Felds gibt Auskunft über den übersetzten Wert der Eigenschaft.
- **GenericAttributes:** Wenn Attribute für das Objekt über die Objektklasse `GenericAttribute` definiert sind, werden alle generischen Attribute mit der für das aktuelle Objekt eingestellten Objektklasseneigenschaft `Owner` in diesem Feld als JSON-Objektliste aufgeführt. Jedes Objekt in der Liste gibt ein Objekt der Klasse `GenericAttribute` mit den Feldern `Refstr`, `Values` und `Translations` zurück, die Daten über das generische Attribut im gleichen Format wie für das Hauptobjekt zurückgeben.

Daten über Objekte einer definierten Objektklasse exportieren, die einer Filterdefinition entsprechen

Endpunkt-Name : `objects`

HTTP-Methode : `POST`

Serviceaufruf:

`ServerAdress/api/v2/objects`

Kopfzeilenfelder:

`Authorization: TypWert`



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

`Content-Type: application/json; charset=utf-8`

Nutzdaten

Die Nutzdaten liegen als JSON-Objekt mit folgender Struktur vor:

```
{
  "CurrentProfile": "UserProfile1",
  "CurrentMandate": "Mandate1",
  "ApiCulture": "ApiCulture_Posix",
  "Language": 1033,
  "DataCulture": "de-DE",
  "EmptyValues": true,
  "Class": "Application",
  "Limit": "80"
  "FilterTextProperties": [
    { "Name": "Content*", "ShortName": "CMS*" },
    { "Name": "Document*" }
  ]
}
```

Sie können folgende Felder enthalten:

Feld	Obligatorisch/Optional	Erforderlicher Wert
<code>"Class": "ObjectClassName"</code>	Obligatorisch	Der Wert des Attributs „Name“ der Alfabet -Objektklasse, für die Objektdaten zurückgegeben werden sollen. Für einen Serviceaufruf kann nur eine Objektklasse definiert werden.
<code>"FilterTextProperties": [{ "PropertyName": "FilterValue", "PropertyName": "FilterValue" },</code>	Obligatorisch	Ein Array mit der Filterdefinition. Das Array muss mindestens ein JSON-Objekt mit einem Feld enthalten. Jedes Feld definiert eine Filterbedingung mit der folgenden Syntax: <code>"ObjectClassPropertyName": "SearchCondition"</code>

Feld	Obligato- risch/Op- tional	Erforderlicher Wert
<pre>{{"Property- tyName": "Filter- Value", "Property- tyName": "Filter- Value"}}</pre>		<p>Beachten Sie Folgendes bezüglich der Filterdefinitionen:</p> <ul style="list-style-type: none"> • Der Suchfilter im Serviceaufruf darf nur Eigenschaften der Objektklasse enthalten, die einen Text zurückgeben, d. h. Objektklasseneigenschaften des Typs <code>Text</code> oder <code>String</code>. • Sie können Sternchen als Platzhalter in der Suchzeichenfolge verwenden. • Wenn die Suchzeichenfolge leer ist, gibt die Suche nur Objekte zurück, für die diese Objektklasseneigenschaft nicht definiert ist. • Jedes JSON-Objekt im Filter-Array kann mehrere Felder enthalten. Die Felder werden mit einer AND-Bedingung bewertet. Beispiel: Der Filter <code>{"Name": "Content*", "ShortName": "CMS*"}</code> findet Objekte mit einem Namen, der mit <code>Content</code> beginnt; UND einem Kurznamen, der mit <code>CMS</code> beginnt. • Das Filter-Array kann mehrere JSON-Objekte enthalten. Die Filterdefinitionen in verschiedenen JSON-Objekten werden mit einer ODER-Bedingung ausgewertet. Beispiel: Der Filter <code>{"Name": "Content*"}, {"ShortName": "CMS*"}</code> findet Objekte mit einem Namen, der mit <code>Content</code> oder einem Kurznamen, der mit <code>CMS</code> beginnt.
<pre>"Limit" :Number</pre>	Optional	<p>Dieses Feld definiert die maximale Anzahl von Objekten, für die die Antwort des Aufrufs Daten zurückgeben soll. Dieser Wert kann festgelegt werden, um die zurückgegebenen Daten für Filterdefinitionen einzugrenzen, die sehr viele Objekte zurückgeben. In Kombination mit dem Feld <code>Offset</code> kann dieses Feld verwendet werden, um das Ergebnis einer großen Suche in mehreren Schritten abzurufen.</p> <p>Ist <code>Limit</code> nicht festgelegt, wird eine Höchstzahl von 1000 Ergebnisdatensätzen aus dem konfigurierten Bericht zurückgegeben.</p>
<pre>"Offset" :Number</pre>	Optional	<p>Dieses Feld definiert die Startposition für die Rückgabe von Werten im Ergebnisdatensatz des konfigurierten Berichts. Wenn z. B. das Feld <code>Limit</code> auf 20 und das Feld <code>Offset</code> auf 10 festgelegt sind, werden die in Zeile 11 bis 30 in der tabellarischen Ausgabe des konfigurierten Berichts aufgeführten Ergebnisse im Serviceaufruf zurückgegeben. Standardmäßig ist die Startposition die erste Zeile im Datensatz mit der Zahl "0".</p>

Feld	Obligato- risch/Op- tional	Erforderlicher Wert
"EmptyValues":true/false	Optional	Wenn <code>EmptyValues</code> auf "true" gesetzt ist, werden alle Objektklasseneigenschaften des Objekts zurückgegeben, auch wenn sie gar nicht festgelegt sind. Wenn <code>EmptyValues</code> auf "false" gesetzt ist, werden nur die Objektklasseneigenschaften des Objekts zurückgegeben, die festgelegt sind. Standardmäßig ist für das Attribut <code>True</code> ausgewählt.
"CurrentProfile": "UserProfile-Name "	Obligato- risch	Der Name eines Anwenderprofils, das dem Alfabet -Anwender, der den Aufruf sendet, zugeordnet ist, welches für den Zugriff auf Alfabet verwendet wird. Das Anwenderprofil ist erforderlich, um auszuwerten, ob der Anwender, der den Serviceaufruf sendet, befugt ist, Daten über ein Objekt zu lesen. Weitere Informationen zu Zugriffsberechtigungen, die von dem Anwenderprofil abhängen, finden Sie im Abschnitt Autorisierung .
"CurrentMandate": "MandateName "	Optional	Dieses Feld ist nur relevant, wenn der Zugriff auf Objekte durch das Mandantenkonzept für föderative Architekturen gesteuert wird, das in Alfabet implementiert ist. Das Feld muss den Namen eines Mandanten angeben, welcher dem Alfabet -Anwender zugeordnet ist, der den Aufruf sendet. Ausführliche Informationen über die Berücksichtigung von Mandanteneinstellungen in REST-API-Serviceaufrufen finden Sie im Abschnitt Zugriff über Mandanten steuern .
"Language": "LanguageLCID "	Optional	<p>Dieses Feld ist nur relevant, wenn in Ihrer Alfabet -Applikation die Instanzübersetzung verwendet wird. Das Feld muss das Sprachencodeximal (Language Code Decimal, LCID) der Landeseinstellung angeben, für die Übersetzungen für den Namen und die Beschreibung des Objekts in der Alfabet-Datenbank vorhanden sind. Standardmäßig wird die Standardsprache der Datenbank zurückgegeben.</p> <p>Hinweis: Es werden keine Werte zurückgegeben, wenn für eine Objektklasseneigenschaft keine Instanzübersetzung verfügbar ist, selbst wenn ein Wert in der Standardsprache bereitgestellt wird.</p> <p>Um alle Werte auf die ursprünglichen Werte zurückzusetzen und Übersetzungen in einer zweiten Sprache hinzuzufügen, sollte der Parameter <code>DataCulture</code> anstatt des Parameters <code>Language</code> verwendet werden. <code>Language</code> löst <code>DataCulture</code> ab. Wenn beide Felder verfügbar sind wird nur <code>Language</code> verwendet.</p>
"DataCulture": "LanguageCode "	Optional	Dieses Feld ist nur relevant, wenn in Ihrer Alfabet -Applikation die Instanzübersetzung verwendet wird. Das Feld muss das Sprachencodeximal (Language Code Decimal, LCID) der Landeseinstellung angeben, für die Übersetzungen für den Namen und die Beschreibung des Objekts in der Alfabet-Datenbank

Feld	Obligato- risch/Op- tional	Erforderlicher Wert
		<p>(zum Beispiel de-DE für Deutsch) vorhanden sind, oder ALL, um Übersetzungen für alle Sprachen zurückzugeben, für die Übersetzungen definiert sind. Zusätzlich zu den Werten in der ursprünglichen Sprache werden dem Rückgabewert in einem separaten Feld <code>Translations</code> Übersetzungen hinzugefügt.</p> <p>Das Feld <code>Language</code> hat Vorrang vor <code>DataCulture</code>. Wenn beide Felder verfügbar sind, wird nur <code>Language</code> verwendet und <code>DataCulture</code> wird ignoriert.</p>
<pre>"ApiCulture": "API- CultureName"</pre>	<p>Optional</p>	<p>Dieses Feld gibt eine API-Landeseinstellung an, die in Alfabet Expand als Teil der Konfiguration des Alfabet -Metamodells definiert wurde. Die API-Landeseinstellung ermöglicht das Definieren von Daten, Datum/Uhrzeit und Uhrzeitmustern sowie Zahlenformaten, die vom Alfabet -Standard abweichen. Wird dieses Feld der Anfrage hinzugefügt, werden Daten, Uhrzeiten und Zahlen im Rückgabewert in den Formaten geschrieben, die in der Definition der API-Landeseinstellung angegeben sind. Informationen hierzu finden Sie im Abschnitt Konfigurieren der Handhabung von Datums-, Uhrzeit- und Zahlenformaten für API-Aufrufe.</p>
<pre>"SortOrder" : " Ob- jectClassProper- tyName "</pre>		<p>Definieren Sie den Namen einer Objektklasseneigenschaft, um die Ergebnisse nach dieser Eigenschaft im Rückgabewert in aufsteigender Reihenfolge alphanumerisch zu sortieren.</p> <p>Standardmäßig werden die Objekte nach aufsteigendem <code>REFSTR</code> sortiert.</p> <p>Wenn Objekte für die ausgewählte Objektklasseneigenschaft keinen Wert aufweisen, werden sie nach aufsteigendem <code>REFSTR</code> sortiert und zuerst aufgeführt.</p> <p>Wenn entweder <code>Language</code> oder <code>DataCulture</code> gesetzt und für die Objektklasseneigenschaft eine Übersetzung aktiviert ist, werden die Werte nach übersetztem Wert sortiert. Wenn für Objektklasseneigenschaftswerte keine Übersetzung angegeben ist, werden diese nach aufsteigendem <code>REFSTR</code> sortiert und zuerst aufgeführt.</p>

Rückgabewert:

Der Rückgabewert ist ein JSON-Objekt mit drei Feldern:

- `Count`: Es wird die Anzahl der Objekte, die in der Datenbank für diese Daten gefunden wurden, zurückgegeben.
- `RejectedObjects`: Die Objekte, die gefunden wurden, für die aber aufgrund fehlender Zugriffsberechtigungen keine Daten zurückgegeben wurden. Das Feld enthält eine JSON-Objektliste mit einem JSON-Objekt für jedes abgelehnte Alfabet -Objekt. Das Objekt hat ein Feld

RefStr, das den REFSTR des Alfabet -Objekts zurückgibt, und ein Feld Message, das Informationen über die Ablehnung enthält.

- Objects: Eine JSON-Liste mit Objekten, wobei jedes Objekt ein Datenbankobjekt darstellt, für das Daten zurückgegeben werden.

```

{
  "Objects": [...],
  "Count": 3,
  "RejectedObjects": [Array[0]]
}
    
```

Das Feld Objects enthält ein JSON-Objekt pro Datenbankobjekt, das für den Aufruf gefunden wurde. Das Objekt enthält die folgenden Felder:

- ClassName: Der Name der Alfabet -Objektklasse, zu der das Objekt gehört.
- RefStr: Der Wert des REFSTR des Objekts.
- Values: Ein JSON-Objekt, das alle Informationen enthält, die zum Objekt in der Datenbank über die Objektklasseneigenschaften definiert sind. Jede Eigenschaft ist ein Feld mit einem Namen, der dem Eigenschaftsnamen entspricht. Der Wert des Felds gibt Auskunft über den Wert der Eigenschaft. Der Rückgabewert für eine Eigenschaft hängt von dem Werttyp der Eigenschaft ab:

Eigen-schaftstyp	Rückgabewert
String, Text, StringArray	String, Text wird als Zeichenfolge zurückgegeben (in Anführungszeichen): <pre>"description": "This application manages customer relations."</pre>
StringArray	Alle ausgewählten Optionen des String-Arrays werden in einer Zeichenfolge zurückgegeben: <pre>"subcategories": "APP_SecurityAssessment APP_CloseSubworkflow"</pre>
Boolean	Die Boolean-Werte sind 1 für „true“, 0 für „false“ oder null, wenn die Eigenschaft nicht festgelegt ist: <pre>"variant": 1</pre>
Real, Integer	Die Real- und Integer-Werte werden als Zahl zurückgegeben: <pre>"xpos": 485.4</pre>
Date, DateTime	Die Date- und DateTime-Informationen werden als Datumszeichenfolgen im Format JJJJ-MM-TT zurückgegeben:

Eigen- schaftstyp	Rückgabewert
	<pre>"last_update": "2012-02-21"</pre>
URL	URLs werden als Zeichenfolge zurückgegeben (in Anführungszeichen): <pre>"url": "http://company.com/ReportServer"</pre>
Referenz	Es wird der REFSTR des Objekts, auf den die Referenz verweist, zurückgegeben: <pre>"ictobject": "26-608-0"</pre>
Refer- enceArray	Für Referenz-Arrays werden die REFSTRs aller Objekte, auf die das aktuelle Objekt mit dieser Eigenschaft verweist, in einem Array aufgelistet: <pre>"applicationgroups": [3] 0: "95-38-0" 1: "95-8-0" 2: "95-9-0"</pre> <p>Der Rückgabewert ist unabhängig von der Einstellung des Attributs Referenzunterstützung der Eigenschaft. Wenn das Attribut Referenzunterstützung der Eigenschaft auf <code>false</code> gesetzt ist, wird der REFSTR-Wert der Referenzziele direkt in der Datenbanktabelle der Objektklasse in einer Spalte für die Eigenschaft gespeichert. Ist Referenzunterstützung auf <code>true</code> gesetzt, werden die Referenzen in der Tabelle RELATIONS gespeichert. Dieser Unterschied ist im Rückgabewert für den Endpunkt <code>objects</code> nicht sichtbar.</p> <p>Wenn Informationen zu den Objekten, auf die vom aktuellen Objekt verwiesen wird, in den Rückgabewert aufgenommen werden sollen, können Sie die Methode zum Auffinden von Objekten über einen konfigurierten Bericht verwenden, der ebenfalls für den Endpunkt <code>objects</code> zur Verfügung steht.</p>

- **Translations:** Dieses Feld ist nur verfügbar, wenn das Feld `DataCulture` in den Nutzdaten des Serviceaufrufs definiert ist. Das Feld enthält eine JSON-Objektliste mit einem Objekt pro Landeseinstellung für Daten. Jedes Objekt hat ein Feld `DataCulture`, das die Sprache zurückgibt, für die Übersetzungen zurückgegeben werden, und ein Feld `Values`, das ein JSON-Objekt mit einem Feld für jede übersetzte Eigenschaft ist. Der Wert des Felds gibt Auskunft über den übersetzten Wert der Eigenschaft.
- **GenericAttributes:** Wenn Attribute für das Objekt über die Objektklasse `GenericAttribute` definiert sind, werden alle generischen Attribute mit der für das aktuelle Objekt eingestellten Objektklasseneigenschaft `Owner` in diesem Feld als JSON-Objektliste aufgeführt. Jedes Objekt in der Liste gibt ein Objekt der Klasse `GenericAttribute` mit den Feldern `Refstr`, `Values` und `Translations` zurück, die Daten über das generische Attribut im gleichen Format wie für das Hauptobjekt zurückgeben.

Exportieren von Informationen zu Objekten, die von einem konfigurierten Bericht gefunden werden

Ein konfigurierter Bericht, der einen tabellarischen Datensatz zurückgibt, der über das Konfigurationstool Alfabet Expand konfiguriert und in der Alfabet-Datenbank gespeichert wird, kann verwendet werden, um alle Daten oder eine Teilmenge der Daten zu exportieren, die über Objekte in der Alfabet-Datenbank zur Verfügung stehen. Über diese Methode werden Daten zu allen Objekten zurückgegeben, die als Basisobjekte für den Bericht gefunden werden.



Bei Alfabet -Abfragen sind die Basisobjekte in den Berichten die Objekte der Objektklasse, die in der `FIND`-Klausel definiert sind. In Native-SQL-Abfragen wird die erste Spalte des Ergebnissatzes nicht in den Abfrageergebnissen angezeigt. Sie müssen den `REFSTR` der Objektklasse angeben, die als Basisklasse ausgewählt ist.

Beachten Sie, dass die Basisklasse in eine andere Klasse für diese Daten geändert werden kann. Sie wird dem Ergebnisdatensatz über die Anweisung `SetRowReference` hinzugefügt.

Der Bericht kann vom Objekt-Endpunkt der Alfabet RESTful API verwendet werden, um die folgenden Daten zu den Objekten bereitzustellen, die von der Abfrage des Berichts gefunden wurden:

- Die Antwort gibt die Daten zurück, die in dem Bericht enthalten sind. Diese Methode hat die folgenden Vorteile:
 - Die Teilmenge der Daten, die zurückgegeben werden, kann von einem bestimmten Parameter abhängen, z. B. alle Applikationen in einer Applikationsgruppe oder alle Komponenten, für die ein Indikator auf einen bestimmten Wert gesetzt ist.
 - Die zu den Objekten zurückgegebenen Informationen sind konfigurierbar. So können Daten mit den Feldnamen bereitgestellt werden, die über den Bericht definiert sind. Beispielsweise wird ein Verweis auf ein anderes Objekt in der Datenbanktabelle des Objekts in einer Spalte mit dem technischen Namen der Eigenschaft und dem Wert, der für den `REFSTR` des referenzierten Objekts festgelegt ist, gespeichert. Der Datensatz in der Abfrage kann diese Informationen in einer Spalte mit einem kundendefinierten Namen unter dem Namen des referenzierten Objekts anstelle des `REFSTR` zurückgeben.
- Die Antwort gibt Informationen zu allen Eigenschaftswerten für alle Basisobjekte zurück, die im Bericht gefunden wurden, und zwar unabhängig von den Daten, die im Datensatz des Berichts definiert sind. Diese Methode liefert das gleiche Ergebnis pro Objekt wie die Methode, die Objekte nach `REFSTR` findet. Diese Methode hat die folgenden Vorteile:
 - Der `REFSTR` der Objekte darf der Client-Applikation nicht bekannt sein, wenn die Anfrage gesendet wird.
 - Die Teilmenge der Daten, die zurückgegeben werden, kann von einem bestimmten Parameter abhängen, z. B. alle Applikationen in einer Applikationsgruppe oder alle Komponenten, für die ein Indikator auf einen bestimmten Wert gesetzt ist.
- Die Antwort gibt alle `REFSTR`-Werte aller Basisobjekte zurück, die in dem Bericht gefunden wurden. Diese Art von Antwort ist hilfreich, wenn die `REFSTR`-Werte einer definierten Objektgruppe als Eingabe erforderlich sind, z. B. bei externen Applikationen, die Links zum Öffnen der Alfabet -Benutzeroberfläche bereitstellen sollen. Weiterführende Informationen finden Sie unter [Zugreifen auf die Alfabet -Benutzeroberfläche von externen Anwendungen](#).

Zur Ausführung dieser Endpunkt-Methoden muss ein konfigurierter Bericht verfügbar sein, bevor der RESTful-Serviceaufruf erfolgt. Im Folgenden werden die erforderliche Konfiguration und der Aufruf beschrieben.

- [Konfigurieren eines Berichts, der in einer RESTful-Serviceanfrage aufgerufen werden kann](#)
- [Serviceaufruf zur direkten Ausführung eines konfigurierten Berichts](#)
- [Serviceaufrufe zum Ausführen offline ausgeführter Berichte](#)
 - [Serviceaufruf zum Auslösen einer Offline-Ausführung](#)
 - [Überprüfen des Status einer asynchronen Berichtsausführung](#)
 - [Exportieren von Ergebnissen asynchroner Berichtsausführung](#)
- [Rückgabewert für die asynchrone ReportResultDataSet Berichtsausführung, die die im Bericht definierten Daten zurückgibt](#)
- [Rückgabewert für die ReportResult-Objects, der alle Eigenschaften der im Bericht gefundenen Objekte zurückgibt](#)
- [Rückgabewert für das Berichtsergebnis, der den REFSTR der im Bericht gefundenen Objekte zurückgibt](#)

Konfigurieren eines Berichts, der in einer RESTful-Serviceanfrage aufgerufen werden kann

Die Konfiguration konfigurierter Berichte und zugrunde liegender Abfragen in Alfabet Expand wird im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand* in den Kapiteln *Berichte konfigurieren* und *Definieren von Abfragen* beschrieben. Diese Thematik wird hier nicht noch einmal erörtert. Die folgende Beschreibung gilt nur für zusätzliche Einstellungen, die für den Bericht erforderlich sind.

Für die Einstellungen der Attribute des Berichts gelten die folgenden besonderen Anforderungen:

- Das Attribut **Für REST-API verwendbar** des konfigurierten Berichts muss auf `True` gesetzt sein.
- Der Bericht muss vom **Typ** `Query` oder `NativeSQL` sein.
- Bei einer langen Ausführungsdauer konfigurierter Berichte kann eine Offline-Ausführung konfiguriert werden. Diese konfigurierten Berichte werden asynchron über die Alfabet -RESTful-Services ausgeführt und benötigen zum Datenexport zwei Serviceaufrufe, einen zum Auslösen der Ausführung und einen zum Herunterladen der Ergebnisse. Die Ausführung dieses Berichtstyps wird im Folgenden getrennt beschrieben. Hinweis: Der Datenexport aus konfigurierten Berichten, die asynchron ausgeführt werden, unterliegt einer Reihe von Beschränkungen. Die Ergebnisse können nur in der Originalsprache heruntergeladen werden. Es werden keine Übersetzungen bereitgestellt. Die Optionen, den konfigurierten Bericht als Objektliste zu verwenden und nur die REFSTR-Werte oder die komplette Gruppe an Objektklasseneigenschaftswerten der vom konfigurierten Bericht gefundenen Objekte herunterzuladen, sind nicht verfügbar.

Für die Abfragedefinition gelten die folgenden besonderen Anforderungen:

- SQL-Abfragen müssen einen REFSTR-Wert als erstes Argument zurückgeben. Es ist möglich, einen Bericht basierend auf nativem SQL zu erzeugen, das Daten zurückgibt, ohne die Daten an ein Basisobjekt zu koppeln, z. B. durch Setzen des ersten Arguments in der `SELECT`-Anweisung auf `NULL`. Der Rückgabewert des Alfabet RESTful-Serviceaufrufs strukturiert die zurückgegebenen

Daten nach Basisobjekt, das im Bericht gefunden wurde. Daher würde diese Art von Bericht dazu führen, dass keine Objekte gefunden werden. Folglich werden keine Daten angezeigt, es sei denn, die Anweisung `SetRowReference` wird verwendet, um das Basisobjekt mit einer anderen Zeile im Datensatz zu definieren und so die Basisobjekte zu identifizieren.

- Alfabet -Abfragesprache-Parameter können sowohl in Alfabet -Abfragen als auch in Native-SQL-Abfragen verwendet werden. Wenn der konfigurierte Bericht Parameterdefinitionen enthält, ist es nicht erforderlich, eine **Berichtansicht** mit Filter-Definitionen zu definieren. Die Parameter werden direkt im Serviceaufruf gesetzt.
- Alfabet -Abfragesprache-Anweisungen können sowohl in Alfabet -Abfragen als auch in Native-SQL-Abfragen genutzt werden. Der Datensatz nach der Ausführung der Anweisungen wird zum Generieren der Rückgabewerte verwendet. Wenn z. B. die `JoinColumns`-Anweisung im konfigurierten Bericht verwendet wird, werden auch die in einer Spalte verbundenen Daten in einem Feld des resultierenden JSON-Objekts angezeigt. Die Anweisung `SetRowReference` ändert das Basisobjekt des Berichtsdatensatzes und auch für den Rückgabewert des Serviceaufrufs.
- Der konfigurierte Bericht muss eine einfache Tabelle zurückgeben. Gruppierete Datensätze werden nicht korrekt verarbeitet. Daten für untergeordnete Ebenen werden ignoriert.

Beachten Sie, dass dem Anwender, der in der Autorisierung der RESTful-Serviceanfrage verwendet wird, ein Anwenderprofil zugewiesen sein muss, das über Zugriffsberechtigungen für den konfigurierten Bericht verfügt. Wenn keines der dem Anwender zugewiesenen Anwenderprofile Zugriff auf den konfigurierten Bericht ermöglicht, werden von der Serviceanfrage keine Daten zurückgegeben.



Informationen über das Konfigurieren eines benutzerdefinierten Wizards finden Sie im Kapitel *Definieren und Verwalten des Anwenderzugriffs auf konfigurierte Berichte* im Referenzhandbuch *Anwender- und Lösungsverwaltung*. Während die Zugriffsberechtigungen für einen konfigurierten Bericht auf der Alfabet -Benutzeroberfläche von mehreren Faktoren abhängig sind, hängen die Zugriffsberechtigungen für den Zugriff auf den konfigurierten Bericht über die RESTful-Service-API lediglich von den auf das Anwenderprofil bezogenen Zugriffsberechtigungen ab, die für den konfigurierten Bericht definiert sind.

Serviceaufruf zur direkten Ausführung eines konfigurierten Berichts

Endpunkt-Name : `objects`

HTTP-Methode : `POST`

Serviceaufruf:

`ServerAdress/api/v2/objects`

Kopfzeilenfelder:

`Authorization: TypWert`



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

`Content-Type: application/json; charset=utf-8`

Nutzdaten

Die Nutzdaten liegen als JSON-Objekt mit folgender Struktur vor:

```

{
  "CurrentProfile": "UserProfile1",
  "CurrentMandate": "Headquarters",
  "Language": 1033,
  "DataCulture": "fr-FR",
  "ApiCulture": "APICultureReporting",
  "EmptyValues": true,
  "Report": "Report_1",
  "ReportResult": "DataSet",
  "Limit": 500,
  "Offset": 250,
  "ReportArgs":
  {
    "arg1": "val1",
    "arg2": "val2",
  }
}
    
```

Es kann folgende Felder enthalten:

Feld	Obligato- risch/Op- tional	Erforderlicher Wert
"Report": " Report- Name "	Obligato- risch	Der Name des konfigurierten Berichts, aus dem der Aufruf Daten liest. Der Name des konfigurierten Berichts ist der Wert der Eigenschaft Name des konfigurierten Berichts.
ReportResult: "Da- taSet/References/Ob- jects"	Obligato- risch	ReportResult muss eines der folgenden sein: <ul style="list-style-type: none"> • DataSet für die Rückgabe aller im Bericht enthaltenen Informationen • References für die Rückgabe einer Liste des REFSTR der im Bericht gefundenen Objekte • Objects für die Rückgabe aller Eigenschaftswerte, die für sämtliche im Bericht gefundenen Objekte festgelegt sind
"Limit" :Number	Optional	Dieses Feld definiert die maximale Anzahl von Objekten, für die die Antwort des Aufrufs Daten zurückgeben soll. Dieser Wert kann festgelegt werden, um die zurückgegebenen Daten für Berichte einzugrenzen, die sehr viele Objekte enthalten. In Kombination mit dem Feld Offset kann dieses Feld

Feld	Obligato- risch/Op- tional	Erforderlicher Wert
		verwendet werden, um das Ergebnis eines großen Berichts in mehreren Schritten abzurufen. Ist <code>Limit</code> nicht festgelegt, wird eine Höchstzahl von 1000 Ergebnisdatensätzen aus dem konfigurierten Bericht zurückgegeben.
<code>"Offset" :Number</code>	Optional	Dieses Feld definiert die Startposition für die Rückgabe von Werten im Ergebnisdatensatz des konfigurierten Berichts. Wenn z. B. das Feld <code>Limit</code> auf 20 und das Feld <code>Offset</code> auf 10 festgelegt sind, werden die in Zeile 11 bis 30 in der tabellarischen Ausgabe des konfigurierten Berichts aufgeführten Ergebnisse im Serviceaufruf zurückgegeben. Standardmäßig ist die Startposition die erste Zeile im Datensatz mit der Zahl "0".
<code>"ReportArgs":</code> <pre> { "arg1": "val1", "arg2": "val2", } </pre>	Optional	Wenn der konfigurierte Bericht Filter-Definitionen enthält, müssen die Werte, die für die Filter bei Ausführung des Berichts über den Serviceaufruf festgelegt werden sollen, mit diesem Feld definiert werden. Der Feldwert ist ein JSON-Objekt, mit einem Feld für jeden Filterparameter des konfigurierten Berichts. Der Feldname muss mit dem Parameter-Namen identisch sein, allerdings ohne das Präfix „@“, und der Feldwert muss der Wert sein, der den Parameter in der Abfrage ersetzen soll. Beachten Sie Folgendes: <ul style="list-style-type: none"> • Werte können vom Typ Ganzzahl, Boolean oder String sein. • Die Platzhalter % und * können in String-Werten verwendet werden. • Alle Filterdefinitionen des konfigurierten Berichts müssen über <code>ReportArgs</code> im Serviceaufruf angegeben werden. Wenn Sie einen der Filterwerte nicht festlegen möchten, können Sie einen Platzhalter als Wert definieren, damit alle Ergebnisse zurückgegeben werden.
<code>"Empty- tyValues":true/false</code>	Optional	Wenn <code>EmptyValues</code> auf <code>true</code> gesetzt ist, werden alle Objektklasseneigenschaften des Objekts zurückgegeben, auch wenn sie gar nicht festgelegt sind. Wenn <code>EmptyValues</code> auf <code>false</code> gesetzt ist, werden nur die Objektklasseneigenschaften des Objekts zurückgegeben, die festgelegt sind. Standardmäßig ist für das Attribut <code>false</code> ausgewählt.

Feld	Obligato- risch/Op- tional	Erforderlicher Wert
"CurrentProfile": " UserProfileName "	Obligato- risch	Der Name eines Anwenderprofils, das dem Alfabet -Anwender, der den Aufruf sendet, zugeordnet ist, welches für den Zugriff auf Alfabet verwendet wird. Das Anwenderprofil ist erforderlich, um auszuwerten, ob der Anwender, der den Serviceaufruf sendet, befugt ist, Daten über ein Objekt zu lesen. Weitere Informationen zu Zugriffsberechtigungen, die von dem Anwenderprofil abhängen, finden Sie im Abschnitt Autorisierung .
"CurrentMandate": " MandateName "	Optional	Dieses Feld ist nur relevant, wenn der Zugriff auf Objekte durch das Mandantenkonzept für föderative Architekturen gesteuert wird, das in Alfabet implementiert ist. Das Feld muss den Namen eines Mandanten angeben, welcher dem Alfabet -Anwender zugeordnet ist, der den Aufruf sendet. Ausführliche Informationen über die Berücksichtigung von Mandanteneinstellungen in REST-API-Serviceaufrufen finden Sie im Abschnitt Zugriff über Mandanten steuern .
"Language": " LanguageLCID "	Optional	<p>Dieses Feld ist nur relevant, wenn Instanzübersetzung in Ihrer Alfabet -Applikation verwendet wird, und wenn der Serviceaufruf alle Eigenschaften der Objekte zurückgibt, die in dem konfigurierten Bericht gefunden wurden, also <code>ReportResult</code> auf <code>Objects</code> gesetzt ist. Das Feld muss das Sprachencoddezimal (Language Code Decimal, LCID) der Landeseinstellung angeben, für die Übersetzungen in der Alfabet-Datenbank vorhanden sind. Standardmäßig wird die Standardsprache der Datenbank zurückgegeben. Beachten Sie bezüglich der verschiedenen Einstellungen des Felds <code>ReportResult</code> Folgendes:</p> <ul style="list-style-type: none"> • <code>DataSet</code>: Wenn der konfigurierte Bericht auf einer Native-SQL-Abfrage basiert, werden Instanzübersetzungen nur zurückgegeben, wenn die Übersetzungen über die Alfabet -spezifische Anweisung <code>*/CULTURE_CODE/*</code> in der Native-SQL-Abfrage enthalten sind. Weitere Informationen über das Konfigurieren von Native-SQL-Abfragen zur Rückgabe von Datenübersetzungswerten finden Sie unter <i>Anzeigen übersetzter Objektdaten in der aktuellen Sprache der Benutzeroberfläche</i> im Referenzhandbuch <i>Konfigurieren von Alfabet mit Alfabet Expand</i>. • <code>Objects</code>: Wenn Objektklasseneigenschaften als übersetzbar definiert sind, werden die Werte nur in der ausgewählten Sprache zurückgegeben. Wenn für eine übersetzbare Objektklasseneigenschaft keine Übersetzung verfügbar ist, werden keine

Feld	Obligato- risch/Op- tional	Erforderlicher Wert
		<p>Werte zurückgegeben, selbst wenn ein Wert in der Standardsprache bereitgestellt wird.</p> <p>Um alle Werte auf die ursprünglichen Werte zurückzusetzen und Übersetzungen in einer zweiten Sprache hinzuzufügen, sollte der Parameter <code>DataCulture</code> anstatt des Parameters <code>Language</code> verwendet werden.</p> <p><code>Language</code> löst <code>DataCulture</code> ab. Wenn beide Felder verfügbar sind wird nur <code>Language</code> verwendet.</p>
<p><code>"DataCulture": " LanguageCode "</code></p>	<p>Optional</p>	<p>Dieses Feld ist nur relevant, wenn in Ihrer Alfabet -Applikation die Instanzübersetzung verwendet wird und übersetzte Werte zurückgegeben werden sollen. Wie die Daten zurückgegeben werden, hängt von der Einstellung im Feld <code>ReportResult</code> ab:</p> <ul style="list-style-type: none"> • Objects: Das Feld muss den Sprachencode (Language Code Decimal, LCID) der Landeseinstellung angeben, für die Übersetzungen in der Alfabet-Datenbank vorhanden sind (zum Beispiel <code>de-DE</code> für Deutsch), oder <code>ALL</code>, um Übersetzungen für alle Sprachen zurückzugeben, für die Übersetzungen definiert sind. Zusätzlich zu den Werten in der ursprünglichen Sprache werden dem Rückgabewert in einem separaten Feld <code>Translations</code> Übersetzungen hinzugefügt. • DataSet: Das Feld muss den Sprachencode (Language Code Decimal, LCID) der Landeseinstellung angeben, für die Übersetzungen in der Alfabet-Datenbank vorhanden sind (zum Beispiel <code>de-DE</code> für Deutsch). Wenn die Objektklasseneigenschaft übersetzbar ist und eine Übersetzung vorliegt, wird der Wert nicht in der Originalsprache, sondern in der Übersetzung zurückgegeben. Wenn keine Übersetzung vorliegt, wird der Wert in der Originalsprache zurückgegeben. <p>Wenn der konfigurierte Bericht auf einer Native-SQL-Abfrage basiert, werden Instanzübersetzungen nur zurückgegeben, wenn die Übersetzungen über die Alfabet -spezifische Anweisung <code>*/CULTURE_CODE/*</code> in der Native-SQL-Abfrage enthalten sind. Weitere Informationen über das Konfigurieren von Native-SQL-Abfragen zur Rückgabe von Datenübersetzungswerten finden Sie unter <i>Anzeigen übersetzter Objektdaten in der aktuellen Sprache</i></p>

Feld	Obligato- risch/Op- tional	Erforderlicher Wert
		<p><i>der Benutzeroberfläche im Referenzhandbuch Konfigurieren von Alfabet mit Alfabet Expand.</i></p> <p>Das Feld <code>Language</code> hat Vorrang vor <code>DataCulture</code>. Wenn beide Felder verfügbar sind, wird nur <code>Language</code> verwendet und <code>DataCulture</code> wird ignoriert.</p>
<code>"ApiCulture": " API-CultureName "</code>	Optional	<p>Dieses Feld gibt eine API-Landeseinstellung an, die in Alfabet Expand als Teil der Konfiguration des Alfabet -Metamodells definiert ist. Die API-Landeseinstellung ermöglicht das Definieren von Datum, Datum/Uhrzeit und Uhrzeitmustern sowie Zahlenformaten, die vom Alfabet -Standard abweichen. Wird dieses Feld der Anfrage hinzugefügt, werden Daten, Uhrzeiten und Zahlen im Rückgabewert in den Formaten geschrieben, die in der Definition der API-Landeseinstellung angegeben sind. Informationen hierzu finden Sie im Abschnitt Konfigurieren der Handhabung von Datums-, Uhrzeit- und Zahlenformaten für API-Aufrufe.</p> <p>Bitte beachten Sie jedoch, dass Werte, die im Feld <code>ReportArgs</code> definiert werden, unabhängig von der API-Landeseinstellung in Alfabet -Standardformaten definiert werden müssen.</p>

Serviceaufrufe zum Ausführen offline ausgeführter Berichte

Bei einer langen Ausführungsdauer konfigurierter Berichte kann eine Offline-Ausführung konfiguriert werden. Diese konfigurierten Berichte werden asynchron ausgeführt und erfordern eine Reihe aufeinander folgender Aufrufe an den Endpunkt `offlineExecution` in der vorgegebenen Reihenfolge. In den folgenden Abschnitten erhalten Sie zu den Endpunkten ausführliche Informationen:

- 1) Planen Sie die asynchrone Ausführung des konfigurierten Berichts über einen Aufruf an den `offlineExecution`-Endpunkt, wobei die Ausführungsdetails in den JSON-Nutzdaten definiert werden. Ein Sitzungstoken wird zurückgegeben. Sobald die zurückgegebene geschätzte Ausführungsdauer abgelaufen ist, kann der nächste Serviceaufruf geplant werden.
- 2) Überprüfen Sie den Status der Berichtsausführung über einen Aufruf an den Endpunkt `offlineExecution`, wobei Sie das zurückgegebene Sitzungstoken als Parameter verwenden. Wenn der zurückgegebene Status `Succeeded` lautet, kann der nächste Serviceaufruf zum Herunterladen geplant werden.
- 3) Laden Sie die Ergebnisse über einen Aufruf an den Endpunkt `offlineExecution` mit dem Sitzungstoken und zusätzlichen Download-Parametern herunter. Bei konfigurierten Berichten, die eine hohe Anzahl an Einträgen zurückgeben, können die Ergebnisse über mehrfache Serviceaufrufe Schritt für Schritt exportiert werden.

In den folgenden Abschnitten finden Sie weitere Informationen über die erforderlichen Aufrufe:

- [Serviceaufruf zum Auslösen einer Offline-Ausführung](#)

- [Überprüfen des Status einer asynchronen Berichtsausführung](#)
- [Exportieren von Ergebnissen asynchroner Berichtsausführung](#)

Serviceaufruf zum Auslösen einer Offline-Ausführung

EndPoint-Name: offlineExecution

HTTP-Methode: POST

Serviceaufruf:

ServerAdress/api/v2/offlineExecution

Kopfzeilenfelder:

Authorization: TypWert



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

Content-Type: application/json; charset=utf-8

Nutzdaten

Die Nutzdaten liegen als JSON-Objekt mit folgender Struktur vor:

```
{
  "CurrentProfile": "UserProfile1",
  "CurrentMandate": "Headquarters",
  "Report": "Report_1",
  "ReportArgs":
    {
      "arg1": "val1",
      "arg2": "val2",
    }
}
```

Sie können folgende Felder enthalten:

Feld	Obligatorisch/Optional	Erforderlicher Wert
"Report": " Report-Name "	Obligatorisch	Der Name des konfigurierten Berichts, aus dem der Aufruf Daten liest. Der Name des konfigurierten Berichts ist der Wert der Eigenschaft Name des konfigurierten Berichts.

Feld	Obligato- risch/Op- tional	Erforderlicher Wert
<pre>"ReportArgs": { "arg1": "val1", "arg2": "val2", }</pre>	Optional	<p>Wenn der konfigurierte Bericht Filter-Definitionen enthält, müssen die Werte, die für die Filter bei Ausführung des Berichts über den Serviceaufruf festgelegt werden sollen, mit diesem Feld definiert werden. Der Feldwert ist ein JSON-Objekt, mit einem Feld für jeden Filterparameter des konfigurierten Berichts. Der Feldname muss mit dem Parameter-Namen identisch sein, allerdings ohne das Präfix „@“, und der Feldwert muss der Wert sein, der den Parameter in der Abfrage ersetzen soll.</p> <p>Beachten Sie Folgendes:</p> <ul style="list-style-type: none"> • Werte können vom Typ Ganzzahl, Boolean oder String sein. • Die Platzhalter % und * können in String-Werten verwendet werden. • Alle Filterdefinitionen des konfigurierten Berichts müssen über <code>ReportArgs</code> im Serviceaufruf angegeben werden. Wenn Sie einen der Filterwerte nicht festlegen möchten, können Sie einen Platzhalter als Wert definieren, damit alle Ergebnisse zurückgegeben werden.
<pre>"CurrentProfile": " UserProfileName "</pre>	Obligato- risch	<p>Der Name eines Anwenderprofils, das dem Alfabet -Anwender, der den Aufruf sendet, zugeordnet ist, welches für den Zugriff auf Alfabet verwendet wird. Das Anwenderprofil ist erforderlich, um auszuwerten, ob der Anwender, der den Serviceaufruf sendet, befugt ist, Daten über ein Objekt zu lesen. Weitere Informationen zu Zugriffsberechtigungen, die von dem Anwenderprofil abhängen, finden Sie im Abschnitt Autorisierung.</p>
<pre>"CurrentMandate": " MandateName "</pre>	Optional	<p>Dieses Feld ist nur relevant, wenn der Zugriff auf Objekte durch das Mandantenkonzept für föderative Architekturen gesteuert wird, das in Alfabet implementiert ist. Das Feld muss den Namen eines Mandanten angeben, welcher dem Alfabet -Anwender zugeordnet ist, der den Aufruf sendet. Ausführliche Informationen über die Berücksichtigung von Mandanteneinstellungen in REST-API-Serviceaufrufen finden Sie im Abschnitt Zugriff über Mandanten steuern.</p>

Rückgabewert:

Der Rückgabewert ist ein JSON-Objekt im folgenden Format:

```
{
  "Count": 0,
  "Name": "RESTReportLongRunning",
```

```

    "Description": "This configured report is executed offline",
    "OfflineExecutionResultStatus": "Queued",
    "OfflineExecutionToken": "ux4zv0r13ocrv1t6dca3zrzi5",
    "OfflineExecutionErrorMessage": ""
    "EstimatedOfflineExecutionTime": "00:05:00"
}

```

Folgende Informationen sind optional:

- **OfflineExecutionToken:** Ein Token, das den Serviceaufrufen hinzugefügt werden muss, um den Status der Berichtsausführung zu überprüfen und Berichtsergebnisse zu exportieren.
- **EstimatedOfflineExecutionTime:** Die geschätzte Ausführungsdauer in Minuten, die im Attribut **Geschätzte Ausführungsdauer** des konfigurierten Berichts als Standardeinstellung zu verwenden ist. Der nächste Aufruf zum Überprüfen des Ausführungsergebnisses sollte nach der geschätzten Dauer der Offline-Ausführung geplant werden.
- **Name:** Der Name des zur Ausführung vorgesehenen konfigurierten Berichts.
- **Description:** Der zusammen mit dem Attribut **Beschreibung** des konfigurierten Berichts angegebene Text.
- **OfflineExecutionResultStatus:** Nach der erfolgreichen Planung der Berichtsausführung lautet der Status der Offline-Ausführung `Queued`.
- **OfflineExecutionErrorMessage:** Wenn ein Fehler auftritt, finden sich in diesem Feld genauere Angaben dazu. Im Falle einer erfolgreichen Ausführung bleibt dieses Feld leer.
- **Count:** Dieses Feld ist nicht relevant.

Überprüfen des Status einer asynchronen Berichtsausführung

EndPoint-Name: `offlineExecution`

HTTP-Methode: `GET`

Serviceaufruf:

```

ServerAdress/api/v2/offlineExecution?offlineExecutionId=OfflineExecutionToken

```

mit den folgenden Parametern:

Parameter	Obligato- risch/Optional	Erforderlicher Wert
<code>offlineExecutionId= OfflineExecutionToken</code>	Obligatorisch	Das <code>OfflineExecutionToken</code> vom Rückgabewert des Serviceaufrufs zur Planung der Berichtsausführung.

Kopfzeilenfelder:

Authorization: TypWert



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

Content-Type: application/json; charset=utf-8

Rückgabewert:

Der Rückgabewert ist ein JSON-Objekt im folgenden Format:

```
{
  "Count": 170,
  "OfflineExecutionResultStatus": "Succeeded",
  "OfflineResultExpirationDateTime": "2020-07-22T10:51:39.29"
}
```

Folgende Informationen sind optional:

- **Count:** Die Anzahl zurückgegebener Datensätze.
- **OfflineExecutionResultStatus:** Gibt Succeeded zurück, wenn Ergebnisse verfügbar sind. Während der Ausführung lautet der Status NotReady.
- **OfflineResultExpirationDateTime:** Informiert über Datum und Uhrzeit, wann die Berichtsergebnisse gelöscht werden. Die Ergebnisse der Offline-Ausführung sind für den im Attribut **Aufbewahrungszeit für Offline-Ergebnis** definierten Zeitraum verfügbar.
- **Error Message:** Wenn der Aufruf fehlschlagen sollte, finden Sie in diesem Feld Angaben zum aufgetretenen Fehler.
- **Error Code:** Wenn der Aufruf fehlschlagen sollte, finden Sie in diesem Feld den Fehlercode der Web Services.

Exportieren von Ergebnissen asynchroner Berichtsausführung

EndPoint-Name: offlineExecution

HTTP-Methode: GET

Serviceaufruf:

ServerAdress/api/v2/offlineExecution?offlineExecutionId=OfflineExecutionToken&offset=StartRowNumber&limit=NumberOfResults

mit den folgenden Parametern:

Parameter	Obligato- risch/Op- tional	Erforderlicher Wert
<code>offlineExecutionId= OfflineExecutionToken</code>	Obligato- risch	Das <code>OfflineExecutionToken</code> vom Rückgabewert des Serviceaufrufs zur Planung der Berichtsausführung.
<code>limit= NumberOfResults</code>	Obligato- risch	Die Anzahl an Datensätzen aus dem Ergebnisdatensatz des konfigurierten Berichts, die vom Aufruf zurückgegeben werden sollen.
<code>offset= StartRowIndex</code>	Obligato- risch	Die Startposition für die Rückgabe von Werten im Ergebnisdatensatz des konfigurierten Berichts. Wenn z. B. der Parameter <code>limit</code> auf 20 und der Parameter <code>offset</code> auf 10 festgelegt sind, werden die in Zeile 11 bis 30 in der tabellarischen Ausgabe des konfigurierten Berichts aufgeführten Ergebnisse im Serviceaufruf zurückgegeben. Standardmäßig ist die Startposition die erste Zeile im Datensatz mit der Zahl „0“.

Kopfzeilenfelder:

`Authorization: TypWert`



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

`Content-Type: application/json; charset=utf-8`

Rückgabewert:

Informationen über den Rückgabewert finden Sie im Folgenden.

Rückgabewert für die asynchrone ReportResultDataSet Berichtsausführung, die die im Bericht definierten Daten zurückgibt

Der Rückgabewert ist ein JSON-Objekt mit vier Feldern:

- **Count:** Die Anzahl der Zeilen im Datensatz des konfigurierten Berichts, auf dem der Serviceaufruf basiert.
- **Name:** Der Name des konfigurierten Berichts, aus dem die Daten laut Definition im Attribut **Name** des konfigurierten Berichts abgeleitet worden sind.
- **Description:** Die Beschreibung des konfigurierten Berichts, aus dem die Daten laut Definition im Attribut **Beschreibung** des konfigurierten Berichts abgeleitet worden sind.
- **Objects:** Eine JSON-Liste von Objekten, wobei jedes Objekt eine Zeile im Datensatz des konfigurierten Berichts darstellt. Es werden Informationen zum Basisobjekt der Zeile und den Werten in der Zeile des Berichts angegeben.

```

{
  "Objects": [...],
  "Count": 3,
  "Name": "ReportName",
  "Description": "Report Description"
}

```

Das Feld `Objects` enthält ein JSON-Objekt pro Zeile im Datensatz des konfigurierten Berichts. Das Objekt enthält die folgenden Felder:

- `ClassName`: Der Name der Alfabet -Objektklasse, zu der das Basisobjekt der aktuellen Zeile gehört.
- `RefStr`: Der Wert des `REFSTR` des Basisobjekts der aktuellen Zeile.
- `Values`: Ein JSON-Objekt, das alle Informationen enthält, die in den Zellen der aktuellen Zeile des konfigurierten Berichts definiert sind. Jede Spalte im Datensatz ist ein Feld mit einem Namen, der dem Spaltennamen entspricht. Der Wert des Felds gibt Auskunft über den Wert in der aktuellen Zeile. Alle Werte werden als Strings zurückgegeben.

Rückgabewert für die `ReportResult-Objects`, der alle Eigenschaften der im Bericht gefundenen Objekte zurückgibt

Der Rückgabewert listet alle Objektklasseneigenschaften auf, die für alle Basisobjekte des konfigurierten Berichts festgelegt sind.

Der Rückgabewert ist ein JSON-Objekt mit vier Feldern:

- `Count`: Die Anzahl der Zeilen im Datensatz des konfigurierten Berichts, auf dem der Serviceaufruf basiert. Die Anzahl kann von der Anzahl der Objekte abweichen, für die Daten zurückgegeben werden. Möglicherweise zeigt der Bericht mehrere Zeilen für das gleiche Basisobjekt an. In diesem Fall wird jedes Basisobjekt nur einmal berücksichtigt.
- `Name`: Der Name des konfigurierten Berichts, aus dem die Daten laut Definition im Attribut **Name** des konfigurierten Berichts abgeleitet worden sind.
- `Description`: Die Beschreibung des konfigurierten Berichts, aus dem die Daten laut Definition im Attribut **Beschreibung** des konfigurierten Berichts abgeleitet worden sind.
- `Objects`: Eine JSON-Liste mit Objekten, wobei jedes Objekt ein Datenbankobjekt darstellt, für das Daten zurückgegeben werden.

```

{
  "Objects": [...],
  "Count": 3
}

```

Das Feld `Objects` enthält ein JSON-Objekt pro Datenbankobjekt, das für den Aufruf gefunden wurde. Das Objekt enthält die folgenden Felder:

- `ClassName`: Der Name der Alfabet -Objektklasse, zu der das Objekt gehört.

- **RefStr:**Der Wert des REFSTR des Objekts.
- **Values:** Ein JSON-Objekt, das alle Informationen enthält, die zum Objekt in der Datenbank über die Objektklasseneigenschaften definiert sind. Jede Eigenschaft ist ein Feld mit einem Namen, der dem Eigenschaftsnamen entspricht. Der Wert des Felds gibt Auskunft über den Wert der Eigenschaft. Der Rückgabewert für eine Eigenschaft hängt von dem Werttyp der Eigenschaft ab:

Eigen-schaftstyp	Rückgabewert
String, Text, StringArray	String, Text wird als Zeichenfolge zurückgegeben (in Anführungszeichen): "description": "This application manages customer relations."
StringArray	Alle ausgewählten Optionen des String-Arrays werden in einer Zeichenfolge zurückgegeben: "subcategories": "APP_SecurityAssessment APP_CloseSubworkflow"
Boolean	Die Boolean-Werte sind 1 für „true“, 0 für „false“ oder null, wenn die Eigenschaft nicht festgelegt ist: "variant": 1
Real, Integer	Die Real- und Integer-Werte werden als Zahl zurückgegeben: "xpos": 485.4
Date, DateTime	Die Date- und DateTime-Informationen werden als Datumszeichenfolgen im Format JJJJ-MM-TT zurückgegeben: "last_update": "2012-02-21"
URL	URLs werden als Zeichenfolge zurückgegeben (in Anführungszeichen): "url": "http://company.com/ReportServer"
Reference	Es wird der REFSTR des Objekts, auf den die Referenz verweist, zurückgegeben: "ictobject": "26-608-0"
ReferenceArray	Für Referenz-Arrays werden die REFSTRs aller Objekte, auf die das aktuelle Objekt mit dieser Eigenschaft verweist, in einem Array aufgelistet: "applicationgroups": [3] 0: "95-38-0" 1: "95-8-0"

Eigen-
schaftstyp

Rückgabewert

2: "95-9-0"

Der Rückgabewert ist unabhängig von der Einstellung des Attributs **Referenzunterstützung** der Eigenschaft. Wenn das Attribut **Referenzunterstützung** der Eigenschaft auf `false` gesetzt ist, wird der `REFSTR`-Wert der Referenzziele direkt in der Datenbanktabelle der Objektklasse in einer Spalte für die Eigenschaft gespeichert. Ist **Referenzunterstützung** auf `true` gesetzt, werden die Referenzen in der Tabelle `RELATIONS` gespeichert. Dieser Unterschied ist im Rückgabewert für den Endpunkt `objects` nicht sichtbar.

Wenn Informationen zu den Objekten, auf die vom aktuellen Objekt verwiesen wird, in den Rückgabewert aufgenommen werden sollen, können Sie die Methode zum Auffinden von Objekten über einen konfigurierten Bericht verwenden, der ebenfalls für den Endpunkt `objects` zur Verfügung steht.

- **GenericAttributes:** Wenn Attribute für das Objekt über die Objektklasse `GenericAttribute` definiert sind, werden alle generischen Attribute mit der für das aktuelle Objekt eingestellten Objektklasseneigenschaft `Owner` in diesem Feld als JSON-Objektliste aufgeführt. Jedes Objekt in der Liste gibt ein Objekt der Klasse `GenericAttribute` mit den Feldern `Refstr`, `Values` und `Translations` zurück, die Daten über das generische Attribut im gleichen Format wie für das Hauptobjekt zurückgeben.

Rückgabewert für das Berichtsergebnis, der den REFSTR der im Bericht gefundenen Objekte zurückgibt

Der Rückgabewert enthält alle `REFSTR`-Werte für das Basisobjekt jeder Zeile im konfigurierten Bericht.

Der Rückgabewert ist ein JSON-Objekt mit vier Feldern:

- **Count:** Die Anzahl der Zeilen im Datensatz des konfigurierten Berichts, auf dem der Serviceaufruf basiert.
- **Name:** Der Name des konfigurierten Berichts, aus dem die Daten laut Definition im Attribut **Name** des konfigurierten Berichts abgeleitet worden sind.
- **Description:** Die Beschreibung des konfigurierten Berichts, aus dem die Daten laut Definition im Attribut **Beschreibung** des konfigurierten Berichts abgeleitet worden sind.
- **Refs:** Ein JSON-Array, das die `REFSTR`-Werte der Basisobjekte im Bericht enthält. Das Array kann einen `REFSTR`-Wert mehrmals enthalten, wenn das Basisobjekt mehrerer Zeilen identisch ist.



```
{
  "Refs": ["95-36-0", "95-43-0", "95-41-0"]
  "Count": 3
}
```

Löschen von Objekten aus der Alfabet-Datenbank

Der Endpunkt `delete` bietet eine Möglichkeit, Objekte und Beziehungen aus der Alfabet-Datenbank zu löschen. Eine Beziehung ist eine Referenz von einem Objekt zu einem anderen Objekt, das in einer Eigenschaft des Typs „Referenz-Array“ gespeichert ist. Die meisten Referenz-Arrays werden in der Beziehungstabelle der Alfabet-Datenbank gespeichert.



Weitere Informationen zur Speicherung von Beziehungen in der Alfabet-Datenbank finden Sie im Kapitel *Das Alfabet -Metamodell in der Alfabet -Datenbank* im Referenzhandbuch *Alfabet-Datenintegrationsframework*.

Die Anfrage muss die Definition der Datenbankmanipulation im Text der Anfrage im JSON-Format enthalten.

Endpunkt-Name : `delete`

HTTP-Methode : `POST`

Serviceaufruf:

`ServerAdress/api/v2/delete`

Kopfzeilenfelder:

`Authorization: TypWert`



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

`Content-Type: application/json; charset=utf-8`

Nutzdaten

Die Nutzdaten liegen als JSON-Objekt mit folgender Struktur vor:

```
{
  "CurrentProfile": "UserProfile1",
  "CurrentMandate": "Mandat1",
  "Refs": ["76-2518-0", "76-3246-0", "405-3-0"],
  "Relations": [{
    "FromRef": "76-2518-0",
    "Property": "BelongsTo",
    "ToRef": "76-3246-0"
  },
  {...}]
}
```

Es kann folgende Felder enthalten:

Feld	Obligatorisch/Optional	Erforderlicher Wert
<code>"Refs":[" RefstrOfObject "," RefstrOfObject "]</code>	Optional	Ein Array mit einem oder mehreren Werten der REFSTR-Eigenschaft der Objekte in der Alfabet-Datenbank, die gelöscht werden sollen
<pre> "Relations": [{ "FromRef": "RefstrOfObject", "Property": "PropertyName", "ToRef": "RefstrOfObject" }, {...}] </pre>	Optional	Eine JSON-Liste von Objekten, wobei jedes Objekt eine zu löschende Beziehung darstellt. Für jede Beziehung, die gelöscht werden soll, müssen die folgenden Felder definiert sein: <ul style="list-style-type: none"> • FromRef: Der REFSTR des Objekts, für das die Beziehung über eine seiner Objektklasseneigenschaften definiert ist. • Property: Der Wert des Attributs Name der Objektklasseneigenschaft, für die die Beziehung gespeichert wird • ToRef: Der REFSTR des Objekts, zu dem die Beziehung erstellt wird
<code>"CurrentProfile": " UserProfileName "</code>	Obligatorisch	Der Name eines Anwenderprofils, das dem Alfabet -Anwender, der den Aufruf sendet, zugeordnet ist, welches für den Zugriff auf Alfabet verwendet wird. Das Anwenderprofil ist erforderlich, um auszuwerten, ob der Anwender, der den Serviceaufruf sendet, Schreibberechtigung für ein Objekt hat. Weitere Informationen zu Zugriffsberechtigungen, die von dem Anwenderprofil abhängen, finden Sie im Abschnitt Autorisierung .
<code>"CurrentMandate": "MandateName"</code>	Optional	Dieses Feld ist nur relevant, wenn der Zugriff auf Objekte durch das Mandantenkonzept für föderative Architekturen gesteuert wird, das in Alfabet implementiert ist. Das Feld muss den Namen eines Mandanten angeben, welcher dem Alfabet -Anwender zugeordnet ist, der den Aufruf sendet. Ausführliche Informationen über die Berücksichtigung von Mandanteneinstellungen in REST-API-Serviceaufrufen finden Sie im Abschnitt Zugriff über Mandanten steuern .

Rückgabewert:

Der Rückgabewert ist ein JSON-Objekt mit einem "Count"-Feld, das Auskunft über die Anzahl der gelöschten Objekte und Beziehungen gibt.

Erstellen und Aktualisieren von Objektdaten in der Alfabet-Datenbank

Der Endpunkt `update` bietet eine Möglichkeit, den Inhalt der Alfabet-Datenbank zu ändern. Neue Objekte können für Alfabet -Objektklassen erzeugt, Eigenschaften für vorhandene Objekte geändert und Beziehungen zwischen Objekten erstellt werden. Die Anfrage muss die Definition der Datenbankmanipulation im Text der Anfrage im JSON-Format enthalten.

Als Ergebnis des Serviceaufrufs wird der Inhalt der Alfabet-Datenbank geändert. Es wird ein Rückgabewert übermittelt, der diese Änderung bestätigt.

EndPoint-Name : `update`

HTTP-Methode : `PUT`

Serviceaufruf:

`ServerAdress/api/v2/update`

Kopfzeilenfelder:

`Authorization: TypWert`



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

`Content-Type: application/json;charset=utf-8`

Nutzdaten

Die Nutzdaten liegen als JSON-Objekt mit folgender Struktur vor:

```
{
  "CurrentProfile": "UserProfile1",
  "CurrentMandate": "Mandate1",
  "ApiCulture": "ApiCultureName",
  "Objects": [...],
  "Relations": [...],
}
```

Es kann folgende Felder enthalten:

Feld	Obligato- risch/Op- tional	Erforderlicher Wert
"Objects"	Optional	Ein Array mit der Spezifikation der Objekte und der Daten, die für alle Objekte geändert werden sollen. Einzelheiten finden Sie weiter unten in den Abschnitten Erzeugen eines neuen Objekts in der Alfabet-Datenbank und Ändern der Eigenschaften eines vorhandenen Objekts in der Alfabet-Datenbank .

Feld	Obligato- risch/Op- tional	Erforderlicher Wert
"Rela- tions"	Optional	Ein Array mit der Spezifikation der Referenzen, die geändert werden sollen. Einzelheiten finden Sie weiter unten im Abschnitt Erstellen oder Aktualisieren einer Beziehung zwischen Objekten in der Alfabet-Datenbank .
"Cur- rentPro- file"	Obligato- risch	Der Name eines Anwenderprofils, das dem Alfabet -Anwender, der den Aufruf sendet, zugeordnet ist, welches für den Zugriff auf Alfabet verwendet wird. Das Anwenderprofil ist erforderlich, um auszuwerten, ob der Anwender, der den Serviceaufruf sendet, Schreibberechtigung für ein Objekt hat. Weitere Informationen zu Zugriffsberechtigungen, die von dem Anwenderprofil abhängen, finden Sie im Abschnitt Autorisierung .
"Current- Mandate"	Optional	Dieses Feld ist nur relevant, wenn der Zugriff auf Objekte durch das Mandantenkonzept für föderative Architekturen gesteuert wird, das in Alfabet implementiert ist. Das Feld muss den Namen eines Mandanten angeben, welcher dem Alfabet -Anwender zugeordnet ist, der den Aufruf sendet. Ausführliche Informationen über die Berücksichtigung von Mandanteneinstellungen in REST-API-Serviceaufrufen finden Sie im Abschnitt Zugriff über Mandanten steuern .
"ApiCul- ture"	Optional	Dieses Feld gibt eine API-Landeseinstellung an, die in Alfabet Expand als Teil der Konfiguration des Alfabet -Metamodells definiert wurde. Die API-Landeseinstellung ermöglicht das Definieren von Daten, Datum/Uhrzeit und Uhrzeitmustern sowie Zahlenformaten, die vom Alfabet -Standard abweichen. Wird dieses Feld der Anfrage hinzugefügt, können Daten, Uhrzeiten und Zahlen in den Objekdatendefinitionen zum Aktualisieren von Objektdaten in den Formaten definiert werden, die in der Definition der API-Landeseinstellung angegeben sind. Informationen hierzu finden Sie im Abschnitt Konfigurieren der Handhabung von Datums-, Uhrzeit- und Zahlenformaten für API-Aufrufe .

In den folgenden Abschnitten werden die verschiedenen Arten von Operationen sowie die erforderliche JSON-Anfrage beschrieben:

- [Erzeugen eines neuen Objekts in der Alfabet-Datenbank](#)
- [Ändern der Eigenschaften eines vorhandenen Objekts in der Alfabet-Datenbank](#)
- [Erstellen oder Aktualisieren einer Beziehung zwischen Objekten in der Alfabet-Datenbank](#)

Eine einzelne JSON-Anfrage kann mehrere unterschiedliche Operationen umfassen. Dies bedeutet, dass Sie z. B. neue Objekte erzeugen und Daten vorhandener Objekte in der gleichen Anfrage aktualisieren können.



Die Aktualisierung der Daten in der Alfabet-Datenbank erfordert die Kenntnis der Objektklassenkonfiguration des Alfabet -Metamodells und/oder vorhandener Objektdaten.

- Informationen zur aktuellen Definition der Objektklassen und Objektklasseneigenschaften können über den Endpunkt `metamodel` abgerufen werden.
- Informationen zu den aktuellen Objektdaten können über den Endpunkt `select` oder `object` abgerufen werden.
- Grundlegende Informationen zur Struktur des Alfabet -Klassenmodells, das erforderlich ist, um die Datenmanipulation direkt auf Datenbankebene durchzuführen, finden Sie unter *Das Alfabet -Metamodell in der Alfabet -Datenbank* im Referenzhandbuch *Alfabet-Datenintegrationsframework*.

Rückgabewert:

Der Rückgabewert ist ein JSON-Objekt, das Auskunft über vorgenommene Änderungen und aufgetretene Fehler gibt. Beispielsweise ist der Rückgabewert beim Erstellen von zwei neuen Objekten wie folgt:

```
{
  "NewObjects": {
    1: "95-61-0"
    2: "95-62-0" }
  "Count": 2
}
```

Der Rückgabewert kann die folgenden Felder haben:

Feld	Wert
"NewObjects"	Dieses Feld enthält ein JSON-Objekt mit einem Feld für jedes neue Objekt. Der Feldname ist die ID, die in der Anfrage definiert ist, und der Feldwert ist der Wert der REFSTR-Eigenschaft des neuen Objekts in der Alfabet-Datenbank. Wenn durch die Anfrage nur Objekte aktualisiert werden, ist dieses Feld leer.
"Access Denied"	Dieses Feld enthält ein JSON-Objekt mit einem Feld für jedes Objekt, für das der Zugriff aufgrund von Mandanteneinstellungen verweigert wird. Der Feldname entspricht dem REFSTR-Wert des Objekts, und der Feldwert ist der Grund für die Zugriffsverweigerung.
"RejectedObjects"	Dieses Feld informiert über ein Objekt, das nicht erstellt oder geändert werden konnte. Das Feld enthält eine JSON-Objektliste mit einem JSON-Objekt für jede abgelehnte Änderung an einem Alfabet -Objekt. Das Objekt hat ein Feld <code>RefStr</code> , das den REFSTR des Alfabet -Objekts zurückgibt, und ein Feld <code>Message</code> , das Informationen über die Ablehnung enthält.
"NotFound"	Wenn eine Beziehung erstellt werden soll und eine oder beide der beteiligten Objekte nicht in der Alfabet-Datenbank vorhanden sind, zeigt der Aufruf ein Feld <code>NotFound</code> an, das ein Array mit den REFSTR-Werten enthält, die nicht gefunden werden konnten.

Feld	Wert
"Count"	<p>Dieses Feld gibt Auskunft über die Anzahl der Objekte und Beziehungen, die aktualisiert oder erzeugt wurden.</p> <p>Hinweis: Die Anzahl an Aktualisierungen und Lösungen von Beziehungen stimmt nicht immer mit der Anzahl an Beziehungen überein, die über den Aufruf aktualisiert oder gelöscht werden sollten. Dies ist auf interne Mechanismen zurückzuführen, beispielsweise Rückbeziehungen, die dazu führen, dass zwei Beziehungen geändert werden, wenn eine Beziehung geändert werden soll. Anhand der zurückgegebenen Anzahl lässt sich somit nicht feststellen, ob die Anfrage vollständig ausgeführt wurde.</p>

Erzeugen eines neuen Objekts in der Alfabet-Datenbank

Das JSON-Objekt der Anfrage zum Erstellen mindestens eines neuen Objekts enthält die folgenden Felder:

- CurrentProfile:** Der Name des Anwenderprofils, das für den Anwender verwendet werden soll, wenn der Aufruf erfolgt. Die Berechtigung zum Erstellen eines Objekts aus einer Objektklasse in der Alfabet-Datenbank wird über das Anwenderprofil bewertet. Ausführliche Informationen dazu finden Sie in diesem Abschnitt. Das Feld ist optional. Wenn es nicht enthalten ist, wird das letzte Anwenderprofil, mit dem der Anwender angemeldet war, für den Aufruf verwendet. Wird die Information über das Anwenderprofil der letzten Anmeldung nicht gefunden, wird das in den Anwendereinstellungen als Standardprofil für den Anwender definierte Anwenderprofil als Standard für das Feld verwendet.
- Objects:** Eine Definition von Objektdaten, die erstellt werden sollen. Details sind unten definiert.

Der Wert des Feldes `Objects` ist eine JSON-Liste mit Objekten, wobei jedes Objekt ein Alfabet -Objekt definiert, das mit den folgenden Feldern erzeugt werden soll:

Feld-name	Erforderlicher Wert	Bemerkung
Class-Name	Der Wert des Attributs Name der Alfabet -Objekt-klasse, für die das Objekt erzeugt werden soll.	
Id	Eine Ganzzahl, die ein eindeutiger Bezeichner innerhalb der JSON-Anfrage ist.	Jedes Objekt, das in einer einzelnen JSON-Anfrage erzeugt wird, muss über eine andere ID verfügen. Die ID wird nur innerhalb einer einzelnen JSON-Anfrage verwendet. Sie hat keine Auswirkungen auf die Alfabet-Datenbank. Sobald die Objekte erzeugt wurden, sind die IDs nicht mehr an die Objekte gekoppelt und können in anderen Serviceaufrufen verwendet werden.

Feldname	Erforderlicher Wert	Bemerkung
Values	<p>Ein JSON-Objekt mit einem Feld für jede Eigenschaft, die definiert werden soll.</p> <p>Der Feldname ist identisch mit dem Namen der Objektklasseneigenschaft, nur geschrieben in Kleinbuchstaben.</p> <p>Der Feldwert ist identisch mit dem Wert der Objektklasseneigenschaft des erzeugten Objekts.</p>	<p>Damit das Objekt erzeugt werden kann, müssen alle Eigenschaften für das Objekt als obligatorisch definiert sein.</p> <p>Die Eigenschaft <code>REFSTR</code> kann nicht in der JSON-Anfrage definiert werden. Sie wird automatisch bei der Erzeugung des Objekts festgelegt und im Rückgabewert des Serviceaufrufs zurückgegeben. Technische Eigenschaften wie das Erzeugungsdatum und der erzeugende Anwender werden ebenfalls automatisch bei Erzeugung des Objekts definiert. Eigenschaften, für die das Attribut Automatisch verwaltet auf <code>true</code> gesetzt ist, müssen über diesen Mechanismus nicht definiert werden.</p> <p>Eigenschaften des Typs "Reference" und "ReferenceArray" können mit diesem Mechanismus nicht aktualisiert werden. Informationen zur Aktualisierung der Eigenschaften des Typs "Reference" und "ReferenceArray" finden Sie unter Erstellen oder Aktualisieren einer Beziehung zwischen Objekten in der Alfabet-Datenbank.</p> <p>Der Wert muss das Format und die Einschränkungen aufweisen, die für die Objektklasseneigenschaft gelten. Beispielsweise kann eine String-Längenbeschränkung für eine Objektklasseneigenschaft des Typs "String" implementiert sein. Ein Datum muss das Format JJJJ-MM-TT haben.</p>
Translations	<p>Eine Liste von JSON-Objekten, wobei jedes Objekt eine Übersetzung repräsentiert und zwei Felder hat: das Feld <code>DataCulture</code>, das den Sprachencode der Landeseinstellung für Daten definiert (z. B. <code>fr-FR</code>) und das Feld <code>Values</code>, das ein Feld für jede hinzuzufügende Übersetzung enthält. Dabei entspricht der Feldname dem Eigenschaftsnamen und der Feldwert der Übersetzung, die hinzugefügt werden soll.</p>	<p>Dieses Feld ist optional und muss hinzugefügt werden, wenn Übersetzungen von Objektklassennamen und -beschreibungen für eine oder mehrere in Alfabet Expand angegebene Landeseinstellungen für Daten vorliegen, die definiert wurden, um Datenübersetzung zu ermöglichen.</p>

Feld-name	Erforderlicher Wert	Bemerkung
GenericAttributes	<p>Eine JSON-Objektliste mit einem JSON-Objekt für jedes generische Attribut, das zu erstellen oder zu aktualisieren ist. Jedes JSON-Objekt hat drei Felder:</p> <ul style="list-style-type: none"> • RefStr: Das Feld muss mit einem leeren Wert definiert werden, um ein neues generisches Attribut zu erstellen. • Values: Ein JSON-Objekt mit einem Feld für jede Eigenschaft, die aktualisiert werden soll. Erforderliche Felder sind <code>Name</code>, <code>Type</code> und <code>Value</code>. Das Feld <code>Group</code> ist optional. Die Eigenschaft <code>Owner</code> der Objektklasse <code>GenericAttribute</code> muss nicht angegeben werden. Sie wird durch den Importmechanismus bei dem <code>REFSTR</code> zu dem Objekt festgelegt, für welches das generische Attribut definiert wird. • Translations: Wenn der Name des generischen Attributs übersetzt werden soll, kann ein Feld angegeben werden wie oben für das Hauptobjekt beschrieben. 	<p>Die Objektklasse <code>GenericAttribute</code> speichert Eigenschaften für verschiedene Objektklassen. Wenn generische Attribute für Objekte definiert werden sollen, die über einen Endpunkt <code>update</code> erstellt wurden, müssen die generischen Referenzdaten direkt in der Anfrage für das Objekt erstellt werden, zu dem diese gehört.</p>



Das folgende Beispiel zeigt die erforderliche JSON-Anfrage zum Erzeugen von zwei Applikationsgruppen mit einer Übersetzung in zwei Sprachen und einem generischen Attribut:

```
{
  "CurrentProfile": "RESTAccessProfile";
  "Objects": [{
    "ClassName" : "ApplicationGroup",
    "Id" : "1",
    "Values" : {
      "name": "Test Group 1",
      "shortname": "TestG1"
    }
  },
  "Translations": [
    { "DataCulture" : "de-DE",
      "Values": {"name": "Testgruppe 1"}
    },
    { "DataCulture" : "fr-FR",
```

```

        "Values":{"name": "Groupe Controle 1"}
    }
]
},
{
"ClassName" : "ApplicationGroup",
"Id" : "2",
"Values" : {
    "name": "Test Group 2",
    "shortname": "TestG2"
},
"Translations":[
    {"DataCulture" : "de-DE",
    "Values":{"name": "Testgruppe 2"}
    },
    {"DataCulture" : "fr-FR",
    "Values":{"name": "Groupe Controle 2"}
    }
],
"GenericAttributes": [{
    "RefStr": "",
    "Values": {"Name": "Integer Attribute 1",
    "Group": "",
    "Type": "Integer",
    "Value": "99999"},
    "Translations": [{
        "DataCulture": "de-DE",
        "Values": {"Name": "Integer-Attribut"}
    }]
    }]
}]
}

```

Ändern der Eigenschaften eines vorhandenen Objekts in der Alfabet-Datenbank

Um die Eigenschaftswerte eines vorhandenen Objekts in der Alfabet-Datenbank zu aktualisieren, muss das JSON-Objekt der Anfrage das Feld `Objects` enthalten. Der Wert dieses Feldes ist eine JSON-Liste mit

Objekten, wobei jedes Objekt ein Alfabeta-Objekt definiert, das mit den folgenden Feldern aktualisiert werden soll:

Feldname	Erforderlicher Wert	Bemerkung
REF-STR	Der Wert der REFSTR-Eigenschaft des Objekts, das aktualisiert werden soll.	
Values	<p>Ein JSON-Objekt mit einem Feld für jede Eigenschaft, die aktualisiert werden soll.</p> <p>Der Feldname ist identisch mit dem Namen der Objektklasseneigenschaft, nur geschrieben in Kleinbuchstaben.</p> <p>Der Feldwert ist identisch mit dem neuen Wert, der für die Objektklasseneigenschaft definiert werden soll.</p>	<ul style="list-style-type: none"> Eigenschaften, für die das Attribut Automatisch verwaltet auf <code>true</code> gesetzt ist, müssen über diesen Mechanismus nicht definiert werden. Eigenschaften des Typs „Reference“ und „ReferenceArray“ können mit diesem Mechanismus nicht aktualisiert werden. Informationen zur Aktualisierung der Eigenschaften des Typs „Reference“ und „ReferenceArray“ finden Sie unter Erstellen oder Aktualisieren einer Beziehung zwischen Objekten in der Alfabeta-Datenbank. Der Wert muss das Format und die Einschränkungen aufweisen, die für die Objektklasseneigenschaft gelten. Beispielsweise kann eine String-Längenbeschränkung für eine Objektklasseneigenschaft des Typs „String“ implementiert sein. Ein Datum muss das Format JJJJ-MM-TT haben. Objektklasseneigenschaften des Datentyps URL oder E-Mail umfassen einen Namen und die tatsächliche Internet-/E-Mail-Adresse, getrennt durch <code>\r\n</code>. Beispiel: <pre>Software Ag\r\nwww.softwareag.com</pre> <p>Wenn in der Definition kein <code>\r\n</code> verfügbar ist, wird die angegebene Zeichenfolge als Internet-/E-Mail-Adresse gespeichert.</p>
Translations	Eine Liste von JSON-Objekten, wobei jedes Objekt eine Übersetzung repräsentiert und zwei Felder hat: das Feld <code>DataCulture</code> , das den Sprachcode der Landeseinstellung für Daten	Dieses Feld ist optional und muss hinzugefügt werden, wenn Übersetzungen von Objektklassennamen und -beschreibungen für eine oder mehrere in Alfabeta Expand angegebene Landeseinstellungen für

Feldname	Erforderlicher Wert	Bemerkung
	<p>definiert (z. B. <code>fr-FR</code>) und das Feld <code>values</code>, das ein Feld für jede hinzuzufügende Übersetzung enthält. Dabei entspricht der Feldname dem Eigenschaftsnamen und der Feldwert der Übersetzung, die hinzugefügt werden soll.</p>	<p>Daten vorliegen, die definiert wurden, um Datenübersetzung zu ermöglichen.</p>
<p><code>GenericAttributes</code></p>	<p>Eine JSON-Objektliste mit einem JSON-Objekt für jedes generische Attribut, das zu erstellen oder zu aktualisieren ist. Jedes JSON-Objekt hat drei Felder:</p> <ul style="list-style-type: none"> • <code>RefStr</code>: Bei Aktualisierungen muss der Wert des Felds der <code>REFSTR</code> des vorhandenen generischen Attributs sein. Für das Erstellen eines neuen generischen Attributs muss das Feld mit einem leeren Wert definiert werden. • <code>Values</code>: Ein JSON-Objekt mit einem Feld für jede Eigenschaft, die aktualisiert werden soll. Erforderliche Felder sind <code>Name</code>, <code>Type</code> und <code>Value</code>. Das Feld <code>Group</code> ist optional. Die Eigenschaft "Eigentümer" der Objektklasse "Generisches Attribut" muss nicht angegeben werden. Sie wird durch den Importmechanismus bei dem <code>REFSTR</code> zu dem Objekt festgelegt, für welches das generische Attribut definiert wird. • <code>Translations</code>: Wenn der Name des generischen Attributs übersetzt werden soll, kann ein Feld angegeben werden wie oben für das Hauptobjekt beschrieben. 	<p>Die Objektklasse <code>GenericAttribute</code> speichert Eigenschaften für verschiedene Objektklassen. Wenn generische Attribute über einen Endpunkt <code>update</code> definiert werden sollen, müssen die generischen Referenzdaten direkt innerhalb der Aktualisierungsanfrage für das Objekt aktualisiert oder erstellt werden, zu dem diese gehört.</p>



Das folgende Beispiel zeigt die erforderliche JSON-Anfrage zum Aktualisieren von zwei Applikationsgruppen einschließlich Übersetzung der Namen für die zwei Objekte:

```
{
  "Objects": [{
    "RefStr" : "95-61-0",
    "Values" : {
      "shortname": "TG1",
      "status": "Planned"
    }
    "Translations":[
      {"DataCulture" : "de-DE",
      "Values":{"name": "Testgruppe 1"}
      },
      {"DataCulture" : "fr-FR",
      "Values":{"name": "Groupe Controle 1"}
      }
    ]
    "GenericAttributes": [{
      "RefStr": "",
      "Values": {"Name": "Integer Attribute 1",
      "Group": "",
      "Type": "Integer",
      "Value": "99999"},
      "Translations": [{
        "DataCulture": "de-DE",
        "Values": {"Name": "Integer-Attribut"}
      }
      ]
    }
  ],
  {
    "RefStr" : "95-62-0",
    "Values" : {
      "shortname": "TG2",
    }
    "Translations":[
      {"DataCulture" : "de-DE",
      "Values":{"name": "Testgruppe 1"}
      },
    ],
  }
}
```

```

        {"DataCulture" : "fr-FR",
        "Values":{"name": "Groupe Controle 1"}
        }
    ]
}]]
}

```

Erstellen oder Aktualisieren einer Beziehung zwischen Objekten in der Alfabet-Datenbank

Es können Beziehungen zwischen Objekten erstellt werden, die bereits in der Alfabet-Datenbank vorhanden sind. Ist ein Objekt nicht vorhanden, wird die Beziehung nicht festgelegt und der REFSTR des Objekts oder der Objekte, das/die nicht gefunden werden konnte/n, wird in dem Rückgabewert in einem Array in einem JSON-Feld `NotFound` zurückgegeben.

Eine Beziehung zwischen zwei Objekten wird über eine Eigenschaft eines der Objekte des Typs „Reference“ oder „ReferenceArray“ hergestellt. Es gibt zwar unterschiedliche Mechanismen für die Speicherung der Beziehungen von Eigenschaften des Typs „Reference“ oder „ReferenceArray“, diese Unterschiede sind für die Definition der Beziehung im Serviceaufruf jedoch nicht relevant. Dies wird von dem Alfabet RESTful API auf Serverseite gehandhabt.

Wird eine Beziehung definiert, die bereits in der Alfabet-Datenbank vorhanden ist, geschieht Folgendes:

- Wenn die Objektklasseneigenschaft, in der die Beziehung gespeichert ist, vom **Typ** `Reference` ist, wird die vorhandene Beziehung überschrieben.
- Wenn die Objektklasseneigenschaft, in der die Beziehung gespeichert ist, vom **Typ** `ReferenceArray` ist, wird die neue Referenz zum Array hinzugefügt und die vorhandenen Referenzen bleiben bestehen. Wenn Sie eine Beziehung des **Typs** `ReferenceArray` ersetzen wollen, müssen Sie die alte Beziehung mit einer Serviceanfrage des Endpunkts `delete` löschen und eine neue mit dieser Serviceanfrage erstellen.

Damit die Anfrage mit dem Serviceaufruf versendet werden kann, müssen die folgenden Informationen angegeben werden:

- Definition des „von“-Objekts. Dies ist das Objekt, für das die Eigenschaft, die die Beziehung herstellt, definiert wird.
- Definition des „bis“-Objekts. Dies ist das Objekt, zu der die Beziehung hergestellt wird.
- Definition der Eigenschaft, die die Beziehung herstellt.

Um eine neue Beziehung zu erzeugen, muss das JSON-Objekt der Anfrage ein Feld `Relations` enthalten. Der Wert dieses Feldes ist eine JSON-Liste mit Objekten, wobei jedes Objekt eine Beziehung zwischen Alfabet -Objekten definiert, die mit den folgenden Feldern erzeugt oder aktualisiert werden soll:

Feld-name	Erforderlicher Wert
FromRef	Der Wert der REFSTR-Eigenschaft des Alfabet -Objekts, für das die Eigenschaft, die die Beziehung herstellt, definiert wird.
Property	Der Wert des Attributs Name der Objektklasseneigenschaft, die die Beziehung herstellt.
ToRef	Der Wert der REFSTR-Eigenschaft des Alfabet -Objekts, zu dem die Beziehung hergestellt wird.



Das folgende Beispiel zeigt eine JSON-Anfrage, die eine Eigenschaft des Typs "Reference" aktualisiert und eine Beziehung für eine Eigenschaft des Typs "ReferenceArray" erzeugt. Das aktualisierte Objekt ist eine Applikationsgruppe. Die Eigenschaft `ResponsibleUser` ist vom Typ "Reference". Die vorhandene Referenz zu einem verantwortlichen Anwender wird also mit der in der Anfrage definierten Beziehung überschrieben. Die Eigenschaft `Applications` ist vom Typ "ReferenceArray". Die in der Anfrage definierte Beziehung wird zu bereits vorhandenen Beziehungen hinzugefügt:

```
{
  "Relations": [{
    "FromRef" : "95-61-0",
    "Property" : "ResponsibleUser",
    "ToRef" : "421-862-0"
  },
  {
    "FromRef" : "95-61-0",
    "Property" : "Applications",
    "ToRef" : "76-2518-0"
  }
]
```

Archivieren von Objekten aus der Alfabet-Datenbank

Der Endpunkt `archiveobject` bietet die Möglichkeit, eine ZIP-Archivdatei für ein oder mehrere ausgewählte Objekte aus der Alfabet-Datenbank zu erstellen. Das Objekt kann nach dem Erstellen der ZIP-Archivdatei optional gelöscht werden.

Wenn ein Alfabet -Objekt archiviert wird, wird eine ZIP-Datei erzeugt, die HTML-Dateien enthält, in denen das Objektprofil des archivierten Objekts sowie die Objektprofile der abhängigen Objekte angezeigt werden.

In jedem archivierten Objektprofil wird eine Reihe vorkonfigurierter Ansichtsseiten angezeigt, wobei die Sichtbarkeit dieser Ansichten von der für die Objektklasse konfigurierten Klasseneinstellung abhängig ist. Wenn auf einer Ansichtsseite abhängige Objekte angezeigt werden, können die Anwender das abhängige Objekt in der HTML-Ansicht anklicken, um eine andere HTML-Datei zu öffnen, in der das archivierte Objektprofil des ausgewählten abhängigen Objekts angezeigt wird.

Alfabet -Objekte werden üblicherweise von einem Lösungsadministrator mit der Funktion **Einfache Suche** im **Admin** -Anwenderprofil archiviert. Wird ein Archiv über die Alfabet -Bedienoberfläche angelegt, wird das archivierte Alfabet -Objekt nach der Generierung der ZIP-Archivdatei aus der Alfabet-Datenbank gelöscht. In der Funktionalität **Archiv-Manager** des **Admin-Anwenderprofils** ist die ZIP-Archivdatei dann zum Herunterladen auf eine lokale Festplatte verfügbar. Nach dem Extrahieren der ZIP-Datei kann die entsprechende HTML-Datei in einem Browser-Fenster angezeigt werden. Eine ZIP-Archivdatei enthält für jede Landeseinstellung, die in Ihrem Unternehmen unterstützt wird, einen Ordner.

Dies dient insbesondere dem Entfernen veralteter Objekte aus der Alfabet-Datenbank und einer damit einhergehenden Leistungssteigerung von Alfabet. Die archivierten Daten können gespeichert und bei Bedarf in einem Webbrowser angezeigt werden.



Informationen zum Archivieren von Alfabet -Objekten über die Alfabet -Benutzeroberfläche finden Sie unter *Löschen und Archivieren von Alfabet-Objekten* im Referenzhandbuch *Anwender- und Lösungsverwaltung*.

Die Archivierung von Objekten über einen RESTful-Serviceaufruf bietet mehr Flexibilität:

- Ein Archiv kann erstellt werden, ohne das Objekt zu löschen.
- Archive können entweder nur in der Funktionalität **Archiv-Manager** der Alfabet -Benutzeroberfläche zur Verfügung gestellt werden, oder sie können zusätzlich direkt während der Ausführung des RESTful-Serviceaufrufs in einem definierten Ordner im lokalen Dateisystem gespeichert werden.

Die Anfrage muss die Definition der Archivierung und Datenbankmanipulation im Text der Anfrage im JSON-Format enthalten.

Endpunkt-Name : archiveobject

HTTP-Methode : POST

Serviceaufruf:

`ServerAdress/api/v2/archiveobject`

Kopfzeilenfelder:

`Authorization: TypWert`



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

`Content-Type: application/json; charset=utf-8`

Nutzdaten

Die Nutzdaten liegen als JSON-Objekt mit folgender Struktur vor:

```
{
```

```

"CurrentProfile": "UserProfile1",
"CurrentMandate": "Mandate1",
"CurrentCulture": "1033",
"Refs": ["76-2518-0", "76-3246-0", "405-3-0"],
>DeleteAfterArchive": "true",
"ZipPath": "C:\\ArchiveObject\\Applications"
    }
    
```

Es kann folgende Felder enthalten:

Feld	Obligato- risch/Op- tional	Erforderlicher Wert
"Refs":[" RefstrO- fObject "," RefstrO- fObject "]	Obligato- risch	Ein Array mit einem oder mehreren Werten der REFSTR-Ei- genschaft der Objekte in der Alfabet-Datenbank, die archi- viert oder archiviert und gelöscht werden sollen.
"CurrentPro- file":" UserProfile- Name "	Obligato- risch	Der Name eines Anwenderprofils, das dem Alfabet -Anwen- der, der den Aufruf sendet, zugeordnet ist, welches für den Zugriff auf Alfabet verwendet wird. Das Anwenderprofil ist erforderlich, um auszuwerten, ob der Anwender, der den Ser- viceaufruf sendet, Schreibberechtigung für ein Objekt hat. Weitere Informationen zu Zugriffsberechtigungen, die von dem Anwenderprofil abhängen, finden Sie im Abschnitt Auto- risierung .
"CurrentMan- date":" Man- dateName "	Optional	Dieses Feld ist nur relevant, wenn der Zugriff auf Objekte durch das Mandantenkonzept für föderative Architekturen gesteuert wird, das in Alfabet implementiert ist. Das Feld muss den Namen eines Mandanten angeben, welcher dem Al- fabet -Anwender zugeordnet ist, der den Aufruf sendet. Aus- führliche Informationen über die Berücksichtigung von Man- danteneinstellungen in REST-API-Serviceaufrufen finden Sie im Abschnitt Zugriff über Mandanten steuern .
"CurrentCulture": " LCID decimal "	Optional	Das Sprachcodezimal (LCID) der Landeseinstellung, die für Meldungen im Rückgabewert verwendet werden soll. Dabei muss es sich um eine Landeseinstellung handeln, für die über die Vokabulardateien in der Alfabet-Datenbank Übersetzun- gen verfügbar sind. Standardmäßig ist die Meldung im Rück- gabewert Englisch.
"ZipPath":" Abso- lutePathToFolder "	Optional	Der absolute Pfad zu dem Ordner auf dem lokalen Dateisys- tem, in dem das Archiv zusätzlich zur Speicherung in der Al- fabet-Datenbank gespeichert werden soll. Wenn der Ordner nicht vorhanden ist, wird er während der Ausführung des Serviceaufrufs erstellt.

Feld	Obligato- risch/Op- tional	Erforderlicher Wert
		<p>Wenn dieses Feld nicht festgelegt ist, wird das Archiv nur in der Alfabet-Datenbank gespeichert und kann über die Funktionalität „Archiv-Manager“ in der Alfabet -Benutzeroberfläche heruntergeladen werden.</p> <p>Beachten Sie, dass die umgekehrten Schrägstriche im Pfad als doppelte umgekehrte Schrägstriche geschrieben werden müssen, damit sie in der JSON-Definition als Text akzeptiert werden.</p>
<code>"DeleteAfterArchive": "true false"</code>	Optional	<p>Setzen Sie den Feldwert auf <code>true</code>, wenn das Objekt nach der Generierung des Archivs gelöscht werden soll. Standardmäßig wird das Objekt nach der Generierung des Archivs nicht gelöscht.</p> <p>Zum Löschen des Objekts sind zusätzliche Zugriffsrechte erforderlich. Wenn Sie das Feld auf <code>true</code> setzen und die Berechtigungen für den Anwender und das Anwenderprofil für die Verarbeitung des Aufrufs nicht ausreichen, um das Objekt zu löschen, wird das Archiv auch dann nicht erstellt, wenn die Berechtigungen zum Erstellen von Archiven erteilt wurden.</p>

Rückgabewert:

Der Rückgabewert ist ein JSON-Objekt mit einem Feld "ResultMessages", das in einem separaten Feld pro Objekt im Format „REFSTR des Objekts“:„Meldung“ über den Erfolgsstatus informiert.

```
{
  "ResultMessages": {
    "95-44-0": "Instance not found",
    "76-3200-0": "Access denied: 'Instance cannot be deleted'",
    "95-43-0": "Object successfully archived."
  }
}
```

Erneutes Generieren des Passworts eines Alfabet -Anwenders

Der Endpunkt `regeneratepassword` bietet die Möglichkeit, das Passwort eines Anwenders zurückzusetzen oder ein erstes Kennwort für einen neuen Anwender zu erstellen. Den Anwendern wird automatisch ein Passwort zugewiesen, und der Anwender wird per E-Mail über die Anmeldedaten informiert.



Diese Funktionalität beinhaltet den Versand von E-Mails an den Anwender über das System. Stellen Sie sicher, dass für Ihre Alfabet -Installation System-E-Mails aktiviert sind. Weitere Informationen über das Aktivieren des Versands von E-Mails finden Sie im Abschnitt *Aktivieren des Sendens von E-Mail-Benachrichtigungen in Alfabet* im Referenzhandbuch *Systemadministration*.

Weitere Informationen zur Erstellung des Nachrichtentexts in den E-Mails oder zur Konfiguration von benutzerdefinierten Textvorlagen, die Sie anstelle der Standardtextvorlagen verwenden können, finden Sie im Abschnitt *Festlegen von benutzerdefinierten Textvorlagen für die Passwortgenerierung* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand* und im Abschnitt *Textvorlagen für die Aktivierung von Anwenderpasswörtern* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand - Anhang*.

Die Aktionen zum Neugenerieren von Passwörtern werden in eine Protokolldatei geschrieben. Standardmäßig ist dies die Datei `PasswordChangeLog.txt` im Arbeitsverzeichnis der Alfabet-Webapplikation. Informationen darüber, wie Sie den Speicherort der Protokolldatei ändern, finden Sie unter *Definition des Speicherorts der Protokolldatei für Aktionen zum Zurücksetzen und Neugenerieren von Passwörtern* im Referenzhandbuch.



Allgemeine Informationen zu den Konfigurations- und Administrationsanforderungen für die Anwenderanmeldung über Anwendername und Passwort finden Sie unter *Konfigurieren der Standardanmeldung* im Referenzhandbuch *Systemadministration*.

Die Anfrage muss die Definition der Archivierung und Datenbankmanipulation im Text der Anfrage im JSON-Format enthalten.

Endpunkt-Name : `regeneratepassword`

HTTP-Methode : `POST`

Serviceaufruf:

`ServerAdress/api/v2/regeneratepassword`

Kopfzeilenfelder:

`Authorization: TypWert`



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

`Content-Type: application/json; charset=utf-8`

Nutzdaten

Die Nutzdaten liegen als JSON-Objekt mit folgender Struktur vor:

```
{
  "CurrentProfile": "UserProfile1",
  "CurrentCulture": "1033",
  "Refs": ["421-2518-0", "421-3-0"],
  "UserNames": ["CUSTOMER", "CLIENTE"]
}
```

Es kann folgende Felder enthalten:

Feld	Obligatorisch/Optional	Erforderlicher Wert
"Refs": [" RefstrOfObject ", " RefstrOfObject "]	Optional	Ein Array, das einen oder mehrere Werte der Eigenschaft REFSTR der Anwender (Objektklasse Person) in der Alfabet-Datenbank enthält, für die das Passwort neu generiert werden soll.
"UserNames": ["CUSTOMER", "CLIENTE"]	Optional	Ein Array, das einen oder mehrere Anwendernamen der Anwender (Eigenschaft USER_NAME der Objektklasse Person) in der Alfabet-Datenbank enthält, für die das Passwort neu generiert werden soll.
"CurrentProfile": " UserProfileName "	Obligatorisch	Der Name eines Anwenderprofils, das dem Alfabet -Anwender, der den Aufruf sendet, zugeordnet ist, welches für den Zugriff auf Alfabet verwendet wird. Das Anwenderprofil ist erforderlich, um auszuwerten, ob der Anwender, der den Serviceaufruf sendet, Schreibberechtigung für ein Objekt hat. Weitere Informationen zu Zugriffsberechtigungen, die von dem Anwenderprofil abhängen, finden Sie im Abschnitt Autorisierung .
"CurrentCulture": " LCID decimal "	Optional	Das Sprachcodedezimal (LCID) der Landeseinstellung, die für Meldungen im Rückgabewert verwendet werden soll. Dabei muss es sich um eine Landeseinstellung handeln, für die über die Vokabulardateien in der Alfabet-Datenbank Übersetzungen verfügbar sind. Standardmäßig ist die Meldung im Rückgabewert Englisch.

Rückgabewert:

Der Rückgabewert ist ein JSON-Objekt mit einem Feld "ResultMessages", das in einem separaten Feld pro definiertem Anwender im Format „REFSTR des Anwenders“: „Meldung“ oder „Anwendername“: „Meldung“ über den Erfolgsstatus informiert.

```
{
  "ResultMessages": {
    "421-2518-0": "User not found",
    "421-3-0": "Password successfully changed.",
    "CUSTOMER": "Password successfully changed.",
    "CLIENTE": "Password successfully changed."
  }
}
```

Anonymisieren von Daten für bestimmte Anwender

Mit diesem Endpunkt können die Daten eines oder mehrerer ausgewählter Alfabet -Anwender anonymisiert werden. Der Endpunkt anonymisiert die Daten nur dann, wenn die folgenden Voraussetzungen erfüllt sind:

- Die Anonymisierung ist in der Alfabet-Datenbank für die Objektklasse `Person` aktiviert. Weitere Informationen zur Anonymisierungsfunktionalität und den erforderlichen Konfigurationsschritten finden Sie unter *Anonymisieren von Daten* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.
- Der Anwender ist standardmäßig nicht aus der Anonymisierung ausgeschlossen. Weitere Informationen zum Ausschluss von Anwendern aus der Anonymisierung finden Sie unter *Anwender von der Anonymisierung ausschließen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.
- Der Alfabet -Anwender, der den Aufruf ausführt, hat die erforderliche Zugriffsberechtigung **Hat den AnonymizeUser-Zugriff**. Weitere Informationen zu Zugriffsberechtigungen für REST-API-Serviceaufrufe finden Sie unter [Generieren eines REST-API-Passworts für einen Anwender](#).



Die Datenanonymisierung kann nicht rückgängig gemacht werden!

Endpunktname : `anonymizeuser`

HTTP-Methode : `PUT`

Serviceaufruf:

`ServerAdress/api/v2/anonymizeuser`

Kopfzeilenfelder:

`Authorization: TypWert`



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

`Content-Type: application/json; charset=utf-8`

Nutzdaten

Die Nutzdaten liegen als JSON-Objekt mit folgender Struktur vor:

```
{
  "Refs": ["421-5-0", "421-9-0"]
}
```

Sie können folgende Felder enthalten:

Feld	Obligato- risch/Op- tional	Erforderlicher Wert
"Refs":[" RefstrOfOb- ject "," RefstrOfOb- ject "]	Obligatorisch	Ein Array mit einem oder mehreren Werten der Eigen- schaft REFSTR von Objekten der Objektklasse Person mit dem Stereotyp User in der Alfabet-Datenbank. Alle im Ar- ray enthaltenen Anwender werden anonymisiert.

Rückgabewert:

Der Rückgabewert ist ein JSON-Objekt mit drei Feldern:

- **Count:** Die Anzahl der Anwender, die anonymisiert wurden.
- **RejectedObjects:** Eine JSON-Liste der Objekte, die nicht anonymisiert wurden, beispielsweise weil der Anwender von der Anonymisierung ausgeschlossen wurde. Das Feld enthält eine JSON-Objektliste mit einem JSON-Objekt für jedes abgelehnte Alfabet -Objekt. Das Objekt hat ein Feld `RefStr`, das den REFSTR des Alfabet -Objekts zurückgibt, und ein Feld `Message`, das Informationen über die Ablehnung enthält.



```
{
  "Count": 3,
  "RejectedObjects":{
    "RefStr": "421-999-0",
    "Message": "Cannot find instance"
  }
}
```

Starten eines Workflows über einen RESTful-Serviceaufruf

Dieser Endpunkt kann verwendet werden, um einen Workflow zu starten, der so konfiguriert ist, dass er automatisch gestartet wird und ein Starten über die REST-API zulässt.

Endpunktname: `workflow`

HTTP-Methode: `POST`

Serviceaufruf:

`ServerAdress/api/v2/workflow`

Kopfzeilenfelder:

`Authorization: TypWert`



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

```
Content-Type: application/json; charset=utf-8
```

Nutzdaten

Die Nutzdaten liegen als JSON-Objekt mit folgender Struktur vor:

```
{
  "WorkflowTemplate": "WorkflowTemplateName"
}
```

Sie können folgende Felder enthalten:

Feld	Obligato- risch/Optional	Erforderlicher Wert
"WorkflowTemplate": " Workflow- TemplateName "	Obligatorisch	Der Name der Workflow-Vorlage, die ausgeführt werden soll.

Rückgabewert:

Der Rückgabewert ist ein JSON-Objekt mit den folgenden Feldern:

- **ResultMessage:** Eine Nachricht mit der Anzahl der gestarteten Workflows.
- **ErrorMessage:** Wenn das Starten des Workflows fehlschlägt, wird in diesem Feld der Grund dafür angezeigt.
- **ErrorCode:** Wenn das Starten des Workflows fehlschlägt, gibt dieses Feld den Fehlercode für den Fehler zurück. Diese Informationen sind nur relevant, wenn Sie das Problem mithilfe der Informationen in der Fehlermeldung nicht beheben können. Sie können dann den Fehlercode und die Fehlermeldung an Ihren Systemadministrator senden, um Hilfe zu erhalten.



```
{
  "ResultMessage": "3 new workflows have been created."
}
```

ADIF-Import über einen RESTful-Serviceaufruf starten

Eine Gruppe von Endpunkten kann verwendet werden, um einen ADIF-Import basierend auf einem ADIF-Importschema zu starten, das in der Alfabet-Datenbank als Ziel für den REST-API-Aufruf gespeichert ist.

Die folgenden Attributeinstellungen werden im ADIF-Importschema für eine Ausführung über die RESTful-Services benötigt:

- **Für REST-API verwendbar :** Auf `True` setzen.
- **Geschätzte Ausführungsdauer:** Geben Sie die geschätzte Ausführungsdauer für den ADIF-Prozess in Minuten an. Die geschätzte Ausführungsdauer wird dem Rückgabewert des RESTful Service-Aufrufs hinzugefügt, der die Ausführung auslöst, um die Clientseite über die

geschätzte Wartedauer zu informieren, bis ein Ergebnis für den ausgelösten ADIF-Import erwartet und über den RESTful Service-Aufruf angefordert werden kann.

Wenn das ADIF-Importschema zum Importieren aus einer Datei konfiguriert ist, kann die Datei entweder über den RESTful-Serviceaufruf gestreamt oder in die **interne Dokumentenauswahl** der Alfabet-Datenbank hochgeladen werden, bevor über den RESTful-Serviceaufruf der ADIF-Import gestartet wird.

Wenn die Daten aus einer Datei importiert werden sollen, müssen die folgenden Voraussetzungen erfüllt sein:

- Der Import kann aus einer oder mehreren Dateien durchgeführt werden. Alle für den Import benötigten Dateien müssen sich in einem einzelnen ZIP-Archiv befinden.
- Wenn die zu importierende Datei sich in der internen Dokumentenauswahl befinden soll, muss die Option **Elemente öffnen** im Attribut **Standardzugriffsberechtigungen** des Ordners in der **internen Dokumentenauswahl** aktiviert werden. Weitere Informationen über die Einstellung der benötigten Zugriffsberechtigungen finden Sie unter *Hochladen von Dokumenten und Verwalten von Anwenderberechtigungen für Dokumentverzeichnisse im Selektor für interne Dokumente* im Referenzhandbuch *Anwender- und Lösungsverwaltung*.
- Wenn die zu importierende Datei sich in der **internen Dokumentenauswahl** befinden soll, kann die Datei in die **interne Dokumentenauswahl** über einen RESTful-Serviceaufruf an den Endpunkt `idocupload` hochgeladen werden.



Alternativ können Dokumente über die Funktionalität „Interne Dokumente“ der Alfabet-Benutzeroberfläche hochgeladen werden. Informationen zum Hochladen von Dokumenten in die **interne Dokumentenauswahl** finden Sie im Referenzhandbuch *Anwender- und Lösungsverwaltung* im Abschnitt *Hochladen von Dokumenten und Verwalten von Anwenderberechtigungen für Dokumentverzeichnisse im Selektor für interne Dokumente*.

ADIF-Importe über RESTful Service-Aufrufe werden entweder asynchron oder synchron ausgeführt und erfordern eine Reihe aufeinanderfolgender Aufrufe an die Endpunkte in der vorgegebenen Reihenfolge. In den folgenden Abschnitten erhalten Sie zu den Endpunkten ausführliche Informationen:



Beachten Sie, dass die Alfabet-Webapplikation so konfiguriert werden muss, dass eine Verbindung zu einem ausgeführten Alfabet-Server hergestellt wird, um ADIF-Prozesse über die RESTful-Services ausführen zu können. Informationen hierzu finden Sie im Referenzhandbuch *Systemadministration*.

- 1) Planen Sie die asynchrone Ausführung des ADIF-Imports über einen Aufruf an den `adifimport`-Endpunkt. Ein Sitzungstoken wird zurückgegeben.

ADIF-Importe aus einer Datei im lokalen Dateisystem und ADIF-Importe aus einer externen Datenbank oder aus einer Datei der **internen Dokumentenauswahl** der Alfabet-Datenbank machen unterschiedliche Einstellungen für den Aufruf an den Endpunkt `adifimport` erforderlich und werden deshalb nachfolgend getrennt beschrieben.

- 2) Überprüfen Sie den Ausführungsstatus des ADIF-Imports über den `adifsessionresult`-Endpunkt mithilfe des zurückgegebenen Sitzungstokens.
- 3) Wenn der vom Aufruf an den `adifsessionresult`-Endpunkt zurückgegebene Status `FAILED` oder `WARNING` lautet, überprüfen Sie die Protokolldatei der ADIF-Ausführung über den `adiflog`-Endpunkt.

In den folgenden Abschnitten finden Sie weitere Informationen über die erforderlichen Aufrufe:

- [Auslösen des ADIF-Imports aus einer externen Datenbank oder einem Dokument in der Alfabet-Datenbank](#)
- [Auslösen des ADIF-Imports aus einem Dateistrom im Serviceaufruf](#)
- [Überprüfen des Ergebnisstatus der ADIF-Ausführung](#)
- [Herunterladen der Protokolldatei zur ADIF-Ausführung](#)

Auslösen des ADIF-Imports aus einer externen Datenbank oder einem Dokument in der Alfabet-Datenbank

EndPoint-Name : adifimport

HTTP-Methode : POST

Serviceaufruf:

ServerAdress/api/v2/adifimport

Kopfzeilenfelder:

Authorization:TypeValue



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

Content-Type: application/json; charset=utf-8

Nutzdaten

Die Nutzdaten liegen als JSON-Objekt mit folgender Struktur vor:

```
{
  "Scheme": "ADIFImportSchemeName",
  "UserArgs": {"@AppRef":"76-2760-0"},
  "ImportfileName":"IDOC:\\FolderName\\FileName.zip";
  "Verbose": false
}
```

Sie können folgende Felder enthalten:

Feld	Obligato- risch/Op- tional	Erforderlicher Wert
"Scheme": " ADIFImportSchemeName "	Obligatorisch	Der Name des ADIF-Import- schemas, das ausgeführt wer- den soll.

Feld	Obligato- risch/Op- tional	Erforderlicher Wert
<pre>"UserArgs": {" arg1name ":" arg1value "," arg2name ": " arg2value "}</pre>	<p>Optional; nur erforderlich, wenn das ADIF-Schema Variablen verwendet.</p>	<p>Wenn das ADIF-Schema zur Verwendung von Variablen konfiguriert ist, können die Variablen im Feld <code>UserArgs</code> als JSON-Objekt mit einem Feld für jede Variable angegeben werden. Der Feldname muss mit dem Variablennamen übereinstimmen, und der Feldwert definiert den Variablenwert für die aktuelle Ausführung des ADIF-Importschemas.</p>
<pre>"ImportFileName":"IDOC:\\ Folder- Name\\FileName.zip"</pre>	<p>Optional, nur erforderlich, wenn die Daten aus einer Datei in der internen Dokumentenauswahl der Alfabet-Datenbank importiert werden.</p>	<p>Wenn die Daten aus einer Datei importiert werden sollen, geben Sie den Pfad zur Datei und den Dateinamen in der internen Dokumentenauswahl der Alfabet-Datenbank an. Der Pfad muss mit <code>IDOC:\</code> anfangen, und es müssen umgekehrte Schrägstriche zwischen den Verzeichnisnamen verwendet werden.</p> <p>Beachten Sie Folgendes:</p> <ul style="list-style-type: none"> • Umgekehrte Schrägstriche in JSON müssen mit einem weiteren umgekehrten Schrägstrich geschützt werden. <p>Wenn Sie die JSON bei der Implementierung eines RESTful-Clients in den Code einbeziehen, müssen die für JSON erforderlichen umgekehrten Schrägstriche möglicherweise geschützt werden, um den Code ausführen zu können. Möglicherweise müssen Sie vier umgekehrte Schrägstriche zwischen Ordnernamen setzen.</p>

Feld	Obligato- risch/Op- tional	Erforderlicher Wert
"Verbose": true/false	Optional, Standard ist false	<p>Wenn dieses Feld auf <code>true</code> ge- setzt ist, werden zusätzliche In- formationen über den laufen- den Prozess in den Protokollie- rungsinformationen in der Ta- belle <code>ALFA_ADIF_SESSION</code> pro- tokolliert.</p> <p>MITTEILUNG: Eine ausführliche Protokollierung ist in den meis- ten Fällen nicht erforderlich. Sie kann zu Leistungseinbußen führen.</p>

Rückgabewert:

Der Rückgabewert ist ein JSON-Objekt im folgenden Format:

```
{
  "Successful": true,
  "SessionToken": "BA7557F7581F46609BCC338E9DBBA96E",
  "EstimatedExecutionTime": 0,
  "ResultMessage": "ADIF scheme has been sent to the Alfabet Server for
asynchronous execution."
}
```

Folgende Informationen werden zurückgegeben:

- **SessionToken:** Kopieren Sie dieses Token, und verwenden Sie es in den Aufrufen an die Endpunkte `adifsessionresult` und `adiflog`, um Informationen darüber zu erhalten, ob die Ausführung des ADIF-Imports erfolgreich verlaufen ist.
- **EstimatedExecutionTime:** Gibt die geschätzte Ausführungsdauer in Minuten gemäß Definition im ADIF-Schema an. Ein Aufruf an die `adifsessionresult`- oder `adiflog`-Endpunkte sollte erst nach der geschätzten Ausführungsdauer geplant werden.
- **Successful:** Gibt `true` zurück, wenn der ADIF-Import erfolgreich zur asynchronen Ausführung geplant wurde. Um Informationen darüber zu erhalten, ob die Ausführung des ADIF-Imports erfolgreich verlaufen ist, müssen Sie eine eigene Anfrage an den Endpunkt `adifresult` senden.
- **ResultMessage:** Gibt eine Beschreibung der erfolgreich ausgeführten Aktion zurück.
- **ErrorMessage:** Wenn der Aufruf fehlschlägt, enthält der Rückgabewert in diesem Feld Informationen über den Grund.
- **ErrorCode:** Wenn der Aufruf fehlschlägt, enthält der Rückgabewert in diesem Feld den Fehlercode des Web-Servers.

Beachten Sie, dass der Versuch, ein ADIF-Exportschema mit dem Endpunkt `adifimport` zu starten, fehlschlagen würde. Die Fehlermeldung informiert nicht über einen falschen Schematyp, gibt aber an, dass das ADIF-Schema nicht gefunden wurde. Dies ist darauf zurückzuführen, dass die REST-API nur in der Teilmenge der ADIF-Importschemata nach dem ADIF-Schemanamen sucht und daher das ADIF-Exportschema nicht finden kann.

Auslösen des ADIF-Imports aus einem Dateistrom im Serviceaufruf

EndPoint-Name : `adifimport`

HTTP-Methode : `POST`

Serviceaufruf:

```
ServerAdress/api/v2/adifimport
```

Kopfzeilenfelder:

```
Authorization:TypeValue
```



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

```
Content-Type: multipart/form-data; charset=utf-8;
```

Nutzdaten:

Die Nutzdaten bestehen aus zwei Teilen:

- 1) Ein JSON-Objekt, das Details über den ADIF-Import bereitstellt.
- 2) Der Inhalt der ZIP-Importdatei.



Auf welche Art die Nutzdaten hinzugefügt werden, hängt vom Client-Typ oder vom Clientcode-Typ ab.

Für RESTful Service-Client-Applikationen und objektorientierte Programmiersprachen wird der Inhalt in Form von zwei separaten Dateien hinzugefügt: einer JSON-Datei, die das JSON-Objekt enthält, und der ZIP-Datei, die die Dateien für den Import enthält.

Für Clients, die in zeichenfolgenorientierten Programmiersprachen implementiert sind, müssen die Nutzdaten mit einem im Kopfzeilenfeld Inhaltstyp definierten Trennzeichen beginnen und enden (`Content-Type: multipart/form-data; charset=utf-8; delimiter: DelimiterStringNotPartOfAnyContent`), und das Trennzeichen muss auch verwendet werden, um die beiden Teile der Nutzdaten voneinander zu trennen, wobei die JSON direkt zwischen den Trennzeichen definiert und die ZIP-Importdatei als Dateistrom bereitgestellt wird. Die beiden Teile der Nutzdaten müssen in der angegebenen Reihenfolge wie folgt definiert sein:

```
---DelimiterStringNotPartOfAnyContent---
{
  "Scheme": "ADIFImportSchemeName",
  "ImportFileName": "FileName.zip",
  "Verbose": true
}
```

```
---DelimiterStringNotPartOfAnyContent--
file content stream data
---DelimiterStringNotPartOfAnyContent--
```

Die JSON kann folgende Felder enthalten:

Feld	Obligato- risch/Op- tional	Erforderlicher Wert
"Scheme": " ADIFImportSchemeName "	Obligato- risch	Der Name des ADIF-Import- schemas, das ausgeführt wer- den soll.
"UserArgs": { " arg1name ":" arg1value "," arg2name ":" " arg2value " }	Optional; nur erforderlich, wenn das ADIF- Schema Va- riablen ver- wendet.	Wenn das ADIF-Schema zur Verwendung von Variablen kon- figuriert ist, können die Variab- len im Feld UserArgs als JSON- Objekt mit einem Feld für jede Variable angegeben werden. Der Feldname muss mit dem Variab- lenamen übereinstimmen, und der Feldwert definiert den Vari- ablenwert für die aktuelle Aus- führung des ADIF-Importsche- mas.
"ImportFileName": " FileName.zip"	Obligato- risch	Der Name der Importdatei für den RESTful Service-Aufruf.
"Verbose": true/false	Optional, der Standard lautet false.	Wenn dieses Feld auf true ge- setzt ist, werden zusätzliche In- formationen über den laufenden Prozess in den Protokollierungs- informationen in der Tabelle ALFA_ADIF_SESSION protokol- liert. MITTEILUNG: Eine ausführliche Protokollierung ist in den meis- ten Fällen nicht erforderlich. Sie kann zu Leistungseinbußen füh- ren.

Rückgabewert:

Der Rückgabewert ist ein JSON-Objekt im folgenden Format:

```
{
  "Successful": true,
  "SessionToken": "BA7557F7581F46609BCC338E9DBBA96E",
```

```

    "EstimatedExecutionTime": 0,
    "ResultMessage": "ADIF scheme has been sent to the Alfabet Server for
asynchronous execution."
}

```

Folgende Informationen werden zurückgegeben:

- **SessionToken:** Kopieren Sie dieses Token, und verwenden Sie es in den Aufrufen an die Endpunkte `adifsessionresult` und `adiflog`, um Informationen darüber zu erhalten, ob die Ausführung des ADIF-Imports erfolgreich verlaufen ist.
- **EstimatedExecutionTime:** Gibt die geschätzte Ausführungsdauer in Minuten gemäß Definition im ADIF-Schema an. Ein Aufruf an die `adifsessionresult`- oder `adiflog`-Endpunkte sollte erst nach der geschätzten Ausführungsdauer geplant werden.
- **Successful:** Gibt `true` zurück, wenn der ADIF-Import erfolgreich zur asynchronen Ausführung geplant wurde. Um Informationen darüber zu erhalten, ob die Ausführung des ADIF-Imports erfolgreich verlaufen ist, müssen Sie eine eigene Anfrage an den Endpunkt `adifresult` senden.
- **ResultMessage:** Gibt eine Beschreibung der erfolgreich ausgeführten Aktion zurück.
- **ErrorMessage:** Wenn der Aufruf fehlschlägt, enthält der Rückgabewert in diesem Feld Informationen über den Grund.
- **ErrorCode:** Wenn der Aufruf fehlschlägt, enthält der Rückgabewert in diesem Feld den Fehlercode des Web-Servers.

Beachten Sie, dass der Versuch, ein ADIF-Exportschema mit dem Endpunkt `adifimport` zu starten, fehlschlagen würde. Die Fehlermeldung informiert nicht über einen falschen Schematyp, gibt aber an, dass das ADIF-Schema nicht gefunden wurde. Dies ist darauf zurückzuführen, dass die REST-API nur in der Teilmenge der ADIF-Importschemata nach dem ADIF-Schmanamen sucht und daher das ADIF-Exportschema nicht finden kann.

Überprüfen des Ergebnisstatus der ADIF-Ausführung

EndPoint-Name: `adifsessionstatus`

HTTP-Methode: `POST`

Serviceaufruf:

```
ServerAdress/api/v2/adifsessionstatus
```

Kopfzeilenfelder:

```
Authorization:TypeValue
```



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

```
Content-Type: application/json; charset=utf-8
```

Nutzdaten

Die Nutzdaten liegen als JSON-Objekt mit folgender Struktur vor:

```
{
  "SessionToken" : "Session Token Returned via the adifimport call"
}
```

Feld	Obligatorisch/Optional	Erforderlicher Wert
"SessionToken" : " Session Token Returned via the adifimport call "	Obligatorisch	Das Sitzungstoken vom Rückgabewert des Aufrufs an den adifimport-Endpunkt, der die Ausführung des ADIF-Imports auslöste.

Rückgabewert:

Der Rückgabewert ist ein JSON-Objekt im folgenden Format:

```
{
  "SessionID": "RPC_20200624182254131",
  "SchemeName": "ADIF Scheme Name",
  "Status": "SUCCESS",
  "Start": "2020-06-24T18:22:54.183",
  "EstimatedExecutionTime": 0,
  "End": "2020-06-24T18:22:54.24",
  "IDOCPath": "",
  "IDOCFileName": "EXPORTTEST.ZIP",
  "Log": true,
  "Count": 0
}
```

Folgende Informationen werden zurückgegeben:

- **SessionID:** Die ID der ADIF-Ausführungssitzung.
- **SchemeName:** Der Name des ausgeführten ADIF-Exportschemas.
- **EstimatedExecutionTime:** Gibt die geschätzte Ausführungsdauer in Minuten gemäß Definition im ADIF-Importschema an. Ein Aufruf an die adifsessionresult- oder adiflog-Endpunkte sollte erst nach der geschätzten Ausführungsdauer geplant werden.
- **Status:** Gibt SUCCESS zurück, wenn die ADIF-Ausführung erfolgreich war; FAILED, wenn während der Ausführung ein Fehler auftrat; WARNING, wenn während der Ausführung eine Warnmeldung in die Protokolldatei geschrieben wurde; und STARTED während der Ausführung.
- **Start:** Gibt Datum und Uhrzeit für den Start der ADIF-Ausführung zurück.
- **End:** Gibt Datum und Uhrzeit für das Ende der ADIF-Ausführung zurück.
- **IDOCPath:** Dieses Feld ist nicht relevant.

- IDOCFileName: Wenn von einer im RESTful-Serviceaufruf gestreamten Datei Daten importiert werden, wird der Dateiname zurückgegeben. Wenn Daten von einer Datei in der **internen Dokumentenauswahl** importiert werden, wird der Dateiname samt Pfad zur Datei in der **internen Dokumentenauswahl** zurückgegeben.
- Log: Gibt `true` zurück, wenn eine Protokolldatei verfügbar ist. Die Protokolldatei kann über einen Aufruf an den `adiflog`-Endpunkt heruntergeladen werden.
- Count: Dieses Feld ist nicht relevant.

Herunterladen der Protokolldatei zur ADIF-Ausführung

EndPoint-Name : `adiflog`

HTTP-Methode : `POST`

Serviceaufruf:

`ServerAdress/api/v2/adiflog`

Kopfzeilenfelder:

`Authorization: TypWert`



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

`Content-Type: application/json; charset=utf-8`

Nutzdaten

Die Nutzdaten liegen als JSON-Objekt mit folgender Struktur vor:

```
{
  "SessionToken" : "Session Token Returned via the adifimport call"
}
```

Feld	Obligatorisch/Optional	Erforderlicher Wert
"SessionToken" : " Session Token Returned via the adifimport call "	Obligatorisch	Das Sitzungstoken vom Rückgabewert des Aufrufs an den <code>adifimport</code> -Endpunkt, der die Ausführung des ADIF-Imports auslöste.

Rückgabewert:

Rückgabewert ist der Inhalt der Protokolldatei.

ADIF-Export über einen RESTful-Serviceaufruf starten

Ein ADIF-Export auf Basis eines in der Alfabet-Datenbank gespeicherten ADIF-Exportschemas kann entweder synchron oder asynchron über die Alfabet RESTful Services gestartet werden.

Wenn das ADIF-Exportschema für den Export in eine Datei konfiguriert ist, werden die generierten Dateien in der **internen Dokumentenauswahl** der Alfabet-Datenbank gespeichert.



Informationen zur **internen Dokumentenauswahl** finden Sie unter *Hochladen von Dokumenten und Verwalten von Anwenderberechtigungen für Dokumentverzeichnisse im Selektor für interne Dokumente* im Referenzhandbuch *Anwender- und Lösungsverwaltung* und unter *Dokumente und Dateien der Alfabet -Anwendergemeinschaft zur Verfügung stellen* im Referenzhandbuch *Systemadministration*.

Folgende Informationen sind verfügbar:

- [Voraussetzungen für das Ausführen eines ADIF-Exports](#)
- [Synchrone Ausführung eines ADIF-Exports](#)
- [Asynchrone Ausführung eines ADIF-Exports](#)
 - [Auslösen der asynchronen Ausführung eines ADIF-Exports](#)
 - [Überprüfen des Ergebnisstatus der ADIF-Ausführung](#)
 - [Herunterladen der Protokolldatei zur ADIF-Ausführung](#)

Voraussetzungen für das Ausführen eines ADIF-Exports

Die folgenden Voraussetzungen müssen erfüllt sein, um einen ADIF-Export über Alfabet RESTful Services ausführen zu können:

- Die folgenden Attributeinstellungen werden im ADIF-Exportschema für eine Ausführung über die RESTful-Services benötigt:
 - **Für REST-API verwendbar** : Auf `True` setzen.
 - **Geschätzte Ausführungsdauer**: Wenn der ADIF-Prozess asynchron ausgeführt werden soll, geben Sie die geschätzte Ausführungsdauer für den ADIF-Prozess in Minuten ein. Die geschätzte Ausführungsdauer wird dem Rückgabewert des RESTful Service-Aufrufs hinzugefügt, der die Ausführung auslöst, um die Clientseite über die geschätzte Wartedauer zu informieren, bis ein Ergebnis für den ausgelösten ADIF-Export erwartet und über den RESTful Service-Aufruf angefordert werden kann.
- Die ADIF-Ausführung erfolgt ausschließlich durch Ausführen von Alfabet-Server, der mit derselben Datenbank wie die Alfabet-Webapplikation verbunden ist. In der Registerkarte **Applikationsserver** der Server-Alias-Konfiguration der Alfabet-Webapplikation ist eine der folgenden Einstellungen erforderlich, um Alfabet-Webapplikation zum Übergeben des ADIF-Prozesses an den Alfabet-Server zu befähigen:
 - Das Attribut **Ereigniswarteschlange für alle Prozesse verwenden** muss ausgewählt sein oder

- der Service **Applikationsserver und Net-Remoting-Service** muss ausgewählt sein, ein Remote-Alias für die Verbindung zum Alfabet-Server muss in das Attribut **Remote-Alias für Verbindung zum Applikationsserver** eingegeben sein und die Option **Server zur Ausführung von ADIF-Prozessen verwenden** muss ausgewählt sein.



Informationen zur erforderlichen Konfiguration für die Verbindung zwischen Alfabet-Webapplikation und Alfabet-Server, einschließlich der Definition eines Remote-Alias, finden Sie unter *Standardkonfiguration der Alfabet -Komponenten* im Referenzhandbuch *Systemadministration*.

Synchrone Ausführung eines ADIF-Exports

EndPoint-Name : adifexport

HTTP-Methode : POST

Serviceaufruf:

`ServerAdress/api/v2/adifexport`

Kopfzeilenfelder:

`Authorization:TypeValue`



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

`Content-Type: application/json; charset=utf-8`

Nutzdaten

Die Nutzdaten liegen als JSON-Objekt mit folgender Struktur vor:

```
{
  "Scheme": "ADIFExportSchemeName",
  "Synchron": true,
  "SynchronTimeout": 60,
  "UserArgs": {"@AppRef":"76-2760-0"},
  "Verbose": false,
}
```

Sie können folgende Felder enthalten:

Feld	Obligatorisch/Optional	Erforderlicher Wert
"Scheme": " ADIFExportSchemeName "	Obligatorisch	Der Name des ADIF-Exportschemas, das ausgeführt werden soll.

Feld	Obligatorisch/Optional	Erforderlicher Wert
<pre>"UserArgs": {" arg1name ":" arg1value "," arg2name ":" arg2value "}</pre>	<p>Optional; nur erforderlich, wenn das ADIF-Schema Variablen verwendet.</p>	<p>Wenn das ADIF-Schema zur Verwendung von Variablen konfiguriert ist, können die Variablen im Feld <code>UserArgs</code> als JSON-Objekt mit einem Feld für jede Variable angegeben werden. Der Feldname muss mit dem Variablennamen übereinstimmen, und der Feldwert definiert den Variablenwert für die aktuelle Ausführung des ADIF-Exportschemas.</p>
<pre>"Verbose": true/false</pre>	<p>Optional, Standard ist <code>false</code></p>	<p>Wenn dieses Feld auf <code>true</code> gesetzt ist, werden zusätzliche Informationen über den laufenden Prozess in den Protokollierungsinformationen in der Tabelle <code>ALFA_ADIF_SESSION</code> protokolliert.</p> <p>Mitteilung: Eine ausführliche Protokollierung ist in den meisten Fällen nicht erforderlich. Sie kann zu Leistungseinbußen führen.</p>
<pre>„Verzeichnispfad“:„IDOC:\\Verzeichnisname“</pre>	<p>Optional, für den Export in eine Datei lautet der Standard <code>IDOC:\\ADIF_SYS</code></p>	<p>Wenn das ausgeführte ADIF-Schema in eine Datei exportiert wird, wird das Exportergebnis in der internen Dokumentenauswahl in dem Ordner gespeichert, der mit diesem Feld definiert wurde, und zwar mit dem Dateinamen, der im Feld <code>ExportFileName</code> definiert ist.</p> <p>Der Pfad muss mit <code>IDOC:\</code> anfangen, und es müssen umgekehrte Schrägstriche zwischen den Verzeichnisnamen verwendet werden.</p>

Feld	Obligatorisch/Optional	Erforderlicher Wert
		<p>Beachten Sie, dass umgekehrte Schrägstriche in JSON mit einem weiteren umgekehrten Schrägstrich geschützt werden müssen. Wenn Sie die JSON bei der Implementierung eines RESTful-Clients in den Code einbeziehen, müssen die für JSON erforderlichen umgekehrten Schrägstriche möglicherweise geschützt werden, um den Code ausführen zu können. Möglicherweise müssen Sie vier umgekehrte Schrägstriche zwischen Ordernamen setzen.</p> <p>Dokumente können nicht direkt in die Datei IDOC:\root.</p>
<p>„Export-Dateiname“:„Dateiname.zip“ hochgeladen werden.</p>	<p>Optional, der Standard lautet <ADIF-Schemaname>_<Zeitstempel>.ZIP</p>	<p>Wenn das ausgeführte ADIF-Schema in eine Datei exportiert wird, wird das Exportergebnis in der internen Dokumentenauswahl in dem Ordner gespeichert, der im Feld <code>ExportFileName</code> definiert wurde, und zwar mit dem Dateinamen, der in diesem Feld definiert ist.</p> <p>Wenn die Datei in der internen Dokumentenauswahl bereits vorhanden ist, wird sie überschrieben.</p>

Rückgabewert:

Der Rückgabewert ist ein JSON-Objekt im folgenden Format:

```
{
  "ExportFile": "",
  "Successful": true,
  "SessionToken": "BA7557F7581F46609BCC338E9DBBA96E",
  "EstimatedExecutionTime": 0,
```

```
"ResultMessage": "ADIF scheme has been sent to the Alfabet Server for asynchronous execution."  
}
```

Folgende Informationen werden zurückgegeben:

- **SessionToken:** Kopieren Sie dieses Token, und verwenden Sie es in den Aufrufen an die `adifsessionresult`- und `adiflog`-Endpunkte, um Informationen darüber zu erhalten, ob die Ausführung des ADIF-Exports erfolgreich verlaufen ist.
- **EstimatedExecutionTime:** Gibt die geschätzte Ausführungsdauer in Minuten gemäß Definition im ADIF-Schema an. Ein Aufruf an die `adifsessionresult`- oder `adiflog`-Endpunkte sollte erst nach der geschätzten Ausführungsdauer geplant werden.
- **Successful:** Gibt `true` zurück, wenn der ADIF-Export erfolgreich zur asynchronen Ausführung geplant wurde. Um Informationen darüber zu erhalten, ob die Ausführung des ADIF-Exports erfolgreich verlaufen ist, müssen Sie eine eigene Anfrage an den Endpunkt `adifresult` senden.
- **ResultMessage:** Gibt eine Beschreibung der erfolgreich ausgeführten Aktion zurück.
- **ExportFile:** Dieses Feld ist nicht relevant.
- **ErrorMessage:** Wenn der Aufruf fehlschlägt, enthält der Rückgabewert in diesem Feld Informationen über den Grund.
- **ErrorCode:** Wenn der Aufruf fehlschlägt, enthält der Rückgabewert in diesem Feld den Fehlercode des Web-Servers.

Beachten Sie, dass der Versuch, ein ADIF-Importschema mit dem Endpunkt `adifexport` zu starten, fehlschlagen würde. Die Fehlermeldung informiert nicht über einen falschen Schematyp, gibt aber an, dass das ADIF-Schema nicht gefunden wurde. Dies ist darauf zurückzuführen, dass die REST-API nur in der Teilmenge der ADIF-Exportschemata nach dem ADIF-Schemanamen sucht und daher das ADIF-Importschema nicht finden kann.

Asynchrone Ausführung eines ADIF-Exports

Asynchrone ADIF-Exporte über RESTful Service-Aufrufe erfordern eine Reihe aufeinanderfolgender Aufrufe an den Endpunkt in der vorgegebenen Reihenfolge. In den folgenden Abschnitten erhalten Sie zu den Endpunkten ausführliche Informationen:

- 1) Planen Sie die asynchrone Ausführung des ADIF-Exports über einen Aufruf an den `adifexport`-Endpunkt. Ein Sitzungstoken wird zurückgegeben.
- 2) Überprüfen Sie den Ausführungsstatus des ADIF-Imports über den `adifsessionresult`-Endpunkt mithilfe des zurückgegebenen Sitzungstokens. Wenn ein ADIF-Export in eine Datei ausgeführt wird, enthält der Rückgabewert des Aufrufs Informationen über den Speicherort der generierten Datei in der **internen Dokumentenauswahl**.
- 3) Wenn der vom Aufruf an den `adifsessionresult`-Endpunkt zurückgegebene Status `FAILED` lautet, überprüfen Sie die Protokolldatei der ADIF-Ausführung über den `adiflog`-Endpunkt.
- 4) Wenn der ADIF-Export in die Datei erfolgreich ausgeführt wurde, können Sie die Datei über den `idocdownload`-Endpunkt herunterladen.

In den folgenden Abschnitten finden Sie weitere Informationen über die erforderlichen Aufrufe:

- [Auslösen der asynchronen Ausführung eines ADIF-Exports](#)
- [Überprüfen des Ergebnisstatus der ADIF-Ausführung](#)
- [Herunterladen der Protokolldatei zur ADIF-Ausführung](#)
- [Herunterladen von Dokumenten aus der internen Dokumentenauswahl](#)

Auslösen der asynchronen Ausführung eines ADIF-Exports

EndPoint-Name : adifexport

HTTP-Methode : POST

Serviceaufruf:

`ServerAdress/api/v2/adifexport`

Kopfzeilenfelder:

`Authorization:TypeValue`



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

`Content-Type: application/json; charset=utf-8`

Nutzdaten

Die Nutzdaten liegen als JSON-Objekt mit folgender Struktur vor:

```
{
  "Scheme": "ADIFExportSchemeName",
  "UserArgs": {"@AppRef": "76-2760-0"},
  "Verbose": false,
}
```

Sie können folgende Felder enthalten:

Feld	Obligatorisch/Optional	Erforderlicher Wert
"Scheme": " ADIFExportSchemeName "	Obligatorisch	Der Name des ADIF-Exportschemas, das ausgeführt werden soll.
"UserArgs": {" arg1name ":" arg1value "," arg2name ":" arg2value "}	Optional; nur erforderlich, wenn das ADIF-Schema Variablen verwendet.	Wenn das ADIF-Schema zur Verwendung von Variablen konfiguriert ist, können die Variablen im Feld UserArgs als JSON-

Feld	Obligatorisch/Optional	Erforderlicher Wert
		<p>Objekt mit einem Feld für jede Variable angegeben werden. Der Feldname muss mit dem Variablennamen übereinstimmen, und der Feldwert definiert den Variablenwert für die aktuelle Ausführung des ADIF-Exportschemas.</p>
<p>"Verbose": true/false</p>	<p>Optional, Standard ist false</p>	<p>Wenn dieses Feld auf true gesetzt ist, werden zusätzliche Informationen über den laufenden Prozess in den Protokollierungsinformationen in der Tabelle ALFA_ADIF_SESSION protokolliert.</p> <p>Mitteilung: Eine ausführliche Protokollierung ist in den meisten Fällen nicht erforderlich. Sie kann zu Leistungseinbußen führen.</p>
<p>„Verzeichnispfad“: „IDOC: \\Verzeichnisname“</p>	<p>Optional, für den Export in eine Datei lautet der Standard IDOC: \\ADIF_SYS</p>	<p>Wenn das ausgeführte ADIF-Schema in eine Datei exportiert wird, wird das Exportergebnis in der internen Dokumentenauswahl in dem Ordner gespeichert, der mit diesem Feld definiert wurde, und zwar mit dem Dateinamen, der im Feld <code>ExportFileName</code> definiert ist.</p> <p>Der Pfad muss mit IDOC: \ anfangen, und es müssen umgekehrte Schrägstriche zwischen den Verzeichnisnamen verwendet werden. Beachten Sie, dass umgekehrte Schrägstriche in JSON mit einem weiteren umgekehrten Schrägstrich geschützt werden müssen. Wenn Sie die</p>

Feld	Obligatorisch/Optional	Erforderlicher Wert
		<p>JSON bei der Implementierung eines RESTful-Clients in den Code einbeziehen, müssen die für JSON erforderlichen umgekehrten Schrägstriche möglicherweise geschützt werden, um den Code ausführen zu können. Möglicherweise müssen Sie vier umgekehrte Schrägstriche zwischen Ordernamen setzen.</p> <p>Dokumente können nicht direkt in die Datei IDOC:\root.</p>
<p>„Export-Dateiname“: „Dateiname.zip“ hochgeladen werden.</p>	<p>Optional, der Standard lautet <ADIF-Schema>_<Zeitstempel>.ZIP</p>	<p>Wenn das ausgeführte ADIF-Schema in eine Datei exportiert wird, wird das Exportergebnis in der internen Dokumentenauswahl in dem Ordner gespeichert, der im Feld <code>ExportFileName</code> definiert wurde, und zwar mit dem Dateinamen, der in diesem Feld definiert ist.</p> <p>Wenn die Datei in der internen Dokumentenauswahl bereits vorhanden ist, wird sie überschrieben.</p>

Rückgabewert:

Der Rückgabewert ist ein JSON-Objekt im folgenden Format:

```
{
  "ExportFile": "",
  "Successful": true,
  "SessionToken": "BA7557F7581F46609BCC338E9DBBA96E",
  "EstimatedExecutionTime": 0,
  "ResultMessage": "ADIF scheme has been sent to the Alfabet Server for asynchronous execution."
}
```

Folgende Informationen werden zurückgegeben:

- **SessionToken:** Kopieren Sie dieses Token, und verwenden Sie es in den Aufrufen an die `adifsessionresult`- und `adiflog`-Endpunkte, um Informationen darüber zu erhalten, ob die Ausführung des ADIF-Exports erfolgreich verlaufen ist.
- **EstimatedExecutionTime:** Gibt die geschätzte Ausführungsdauer in Minuten gemäß Definition im ADIF-Schema an. Ein Aufruf an die `adifsessionresult`- oder `adiflog`-Endpunkte sollte erst nach der geschätzten Ausführungsdauer geplant werden.
- **Successful:** Gibt `true` zurück, wenn der ADIF-Export erfolgreich zur asynchronen Ausführung geplant wurde. Um Informationen darüber zu erhalten, ob die Ausführung des ADIF-Exports erfolgreich verlaufen ist, müssen Sie eine eigene Anfrage an den Endpunkt `adifresult` senden.
- **ResultMessage:** Gibt eine Beschreibung der erfolgreich ausgeführten Aktion zurück.
- **ExportFile:** Dieses Feld ist nicht relevant.
- **ErrorMessage:** Wenn der Aufruf fehlschlägt, enthält der Rückgabewert in diesem Feld Informationen über den Grund.
- **ErrorCode:** Wenn der Aufruf fehlschlägt, enthält der Rückgabewert in diesem Feld den Fehlercode des Web-Servers.

Beachten Sie, dass der Versuch, ein ADIF-Importschema mit dem Endpunkt `adifexport` zu starten, fehlschlagen würde. Die Fehlermeldung informiert nicht über einen falschen Schematyp, gibt aber an, dass das ADIF-Schema nicht gefunden wurde. Dies ist darauf zurückzuführen, dass die REST-API nur in der Teilmenge der ADIF-Exportschemata nach dem ADIF-Schemanamen sucht und daher das ADIF-Importschema nicht finden kann.

Überprüfen des Ergebnisstatus der ADIF-Ausführung

EndPoint-Name: `adifsessionstatus`

HTTP-Methode: `POST`

Serviceaufruf:

`ServerAdress/api/v2/adifsessionstatus`

Kopfzeilenfelder:

`Authorization:TypeValue`



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

`Content-Type: application/json; charset=utf-8`

Nutzdaten

Die Nutzdaten liegen als JSON-Objekt mit folgender Struktur vor:

```
{
  "SessionToken" : "Session Token Returned via the adifexport call"
}
```

Feld	Obligatorisch/Optional	Erforderlicher Wert
"SessionToken" : " Session Token Returned via the adifexport call "	Obligatorisch	Das Sitzungstoken vom Rückgabewert des Aufrufs an den adifexport-Endpunkt, der die Ausführung des ADIF-Exports auslöste.

Rückgabewert:

Der Rückgabewert ist ein JSON-Objekt im folgenden Format:

```
{
  "SessionID": "RPC_20200624182254131",
  "SchemeName": "ADIFExportSchemeName",
  "Status": "SUCCESS",
  "Start": "2020-06-24T18:22:54.183",
  "EstimatedExecutionTime": 0,
  "End": "2020-06-24T18:22:54.24",
  "IDOCPath": "IDOC:\\DOCUMENTS",
  "IDOCFileName": "EXPORTTEST.ZIP",
  "Log": true,
  "Count": 0
}
```

Folgende Informationen sind optional:

- **SessionID:** Die ID der ADIF-Ausführungssitzung.
- **SchemeName:** Der Name des ausgeführten ADIF-Exportschemas.
- **EstimatedExecutionTime:** Gibt die geschätzte Ausführungsdauer in Minuten gemäß Definition im ADIF-Schema an. Ein Aufruf an die adifsessionresult- oder adiflog-Endpunkte sollte erst nach der geschätzten Ausführungsdauer geplant werden.
- **Status:** Gibt SUCCESS zurück, wenn die ADIF-Ausführung erfolgreich war; FAILED, wenn während der Ausführung ein Fehler auftrat; WARNING, wenn während der Ausführung eine Warnmeldung in die Protokolldatei geschrieben wurde; und STARTED während der Ausführung.
- **Start:** Gibt Datum und Uhrzeit für den Start der ADIF-Ausführung zurück.
- **End:** Gibt Datum und Uhrzeit für das Ende der ADIF-Ausführung zurück.
- **IDOCPath:** Wenn das ADIF-Schema für den Export in eine Datei konfiguriert ist, wird die Datei in der **internen Dokumentenauswahl** der Alfabet-Datenbank gespeichert. Anschließend wird in diesem Feld der Pfad innerhalb der **internen Dokumentenauswahl** zurückgegeben. Diese Informationen können zusammen mit den im IDOCFileName-Feld zurückgegebenen Informationen verwendet werden, um die Ergebnisse über einen Aufruf an den idocdownload-Endpunkt herunterzuladen. Mehr Informationen zum Download des Dokuments finden Sie unter [Herunterladen von Dokumenten aus der internen Dokumentenauswahl](#).

- **IDOCFileName:** Wenn das ADIF-Schema für den Export in eine Datei konfiguriert ist, wird die Datei in der **internen Dokumentenauswahl** der Alfabet-Datenbank gespeichert. Der Name der Datei wird in diesem Feld zurückgegeben. Diese Informationen können zusammen mit den im `IDOCPath`-Feld zurückgegebenen Informationen verwendet werden, um die Ergebnisse über einen Aufruf an den `idocdownload`-Endpunkt herunterzuladen. Mehr Informationen zum Download des Dokuments finden Sie unter [Herunterladen von Dokumenten aus der internen Dokumentenauswahl](#).
- **Log:** Gibt `true` zurück, wenn eine Protokolldatei verfügbar ist. Die Protokolldatei kann über einen Aufruf an den `adiflog`-Endpunkt heruntergeladen werden.
- **Count:** Dieses Feld ist nicht relevant.

Herunterladen der Protokolldatei zur ADIF-Ausführung

EndPoint-Name: `adiflog`

HTTP-Methode: `POST`

Serviceaufruf:

`ServerAdress/api/v2/adiflog`

Kopfzeilenfelder:

`Authorization:TypeValue`



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

`Content-Type: application/json; charset=utf-8`

Nutzdaten

Die Nutzdaten liegen als JSON-Objekt mit folgender Struktur vor:

```
{
  "SessionToken" : "Session Token Returned via the adifexport call"
}
```

Feld	Obligato- risch/Op- tional	Erforderlicher Wert
"SessionToken" : " Über den Aufruf des Endpunkts adifexport zurückgegebener Token "	Obligatorisch	Das Sitzungstoken vom Rückgabewert des Aufrufs an den <code>adifexport</code> -Endpunkt, der die Ausführung des ADIF-Exports auslöste.

Rückgabewert:

Rückgabewert ist der Inhalt der Protokolldatei.

Exportieren von Informationen über den Inhalt der internen Dokumentenauswahl

Dieser Endpunkt kann verwendet werden, um Informationen über den Dateiinhalt von Ordnern in der **internen Dokumentenauswahl** der Alfabet-Datenbank zu exportieren. Über einen Aufruf an den Endpunkt können nur Dateien in einem einzigen ausgewählten Ordner aufgelistet werden. Unterordner des ausgewählten Ordners und ihre Dateiinhalte sind in der Liste nicht enthalten.



Informationen zur **internen Dokumentenauswahl** finden Sie unter *Hochladen von Dokumenten und Verwalten von Anwenderberechtigungen für Dokumentverzeichnisse im Selektor für interne Dokumente* im Referenzhandbuch *Anwender- und Lösungsverwaltung* und unter *Dokumente und Dateien der Alfabet -Anwendergemeinschaft zur Verfügung stellen* im Referenzhandbuch *Systemadministration*.

Endpunktname: idocfilelist

HTTP-Methode: POST

Serviceaufruf:

`ServerAdress/api/v2/idocfilelist`

Kopfzeilenfelder:

`Authorization: TypWert`



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

`Content-Type: application/json; charset=utf-8`

Nutzdaten:

Die Nutzdaten liegen als JSON-Objekt mit folgender Struktur vor:

```
{
  "Path": "IDOC:\\Folder\\Subfolder"
}
```

Sie können folgende Felder enthalten:

Feld	Obligatorisch/Optional	Erforderlicher Wert
"Path": " Path to folder in IDOC "	Obligatorisch	Der Pfad zum IDOC-Ordner, für den die Liste erstellt werden soll. Der Pfad muss mit IDOC:\ anfangen, und es müssen umgekehrte Schrägstriche zwischen den Verzeichnisnamen verwendet werden. Beachten Sie, dass umgekehrte Schrägstriche in JSON mit einem weiteren umgekehrten Schrägstrich geschützt werden müssen.

Feld	Obligato- risch/Op- tional	Erforderlicher Wert
		Wenn Sie die JSON bei der Implementierung eines RESTful-Clients in den Code einbeziehen, müssen die für JSON erforderlichen umgekehrten Schrägstriche möglicherweise geschützt werden, um den Code ausführen zu können. Möglicherweise müssen Sie vier umgekehrte Schrägstriche zwischen Ordnernamen setzen.

Rückgabewert:

Der Rückgabewert ist eine JSON-Liste von Objekten mit einem Objekt pro Dokument, die mit den folgenden Feldern über das Dokument informiert:

- **Path:** Der vollständige Pfad des Dokuments, beginnend mit dem Root-Verzeichnis der internen Dokumentenauswahl.
- **Name:** Der Name des Dokuments.
- **CreationDate:** Das Datum und die Uhrzeit, zu der das Dokument in die interne Dokumentenauswahl hochgeladen wurde.
- **DateModified:** Das Datum und die Uhrzeit, zu der das Dokument in der internen Dokumentenauswahl zuletzt geändert wurde.
- **Size:** Die Größe des Dokuments in Byte.



```
{
  "Files": [
    {
      "Path": "IDOC:\\Documents\\CC_Corporate_FI-CO.pdf",
      "Name": "CC_Corporate_FI-CO.pdf",
      "CreationDate": "2008-10-03T13:45:48.5",
      "DateModified": "2010-04-04T13:45:48",
      "Size": 24117
    },
    {
      "Path": "IDOC:\\Documents\\CC_TradeWeb.pdf",
      "Name": "CC_TradeWeb.pdf",
      "CreationDate": "2007-08-30T13:46:09.14",
      "DateModified": "2009-02-28T13:46:09",
      "Size": 24892
    }
  ]
}
```

Herunterladen von Dokumenten aus der internen Dokumentenauswahl

Dieser Endpunkt kann verwendet werden, um Dateien aus der internen Dokumentenauswahl der Alfabet-Datenbank oder einem ihrer Unterordner herunterzuladen.



Informationen zur internen Dokumentenauswahl finden Sie unter *Hochladen von Dokumenten und Verwalten von Anwenderberechtigungen für Dokumentverzeichnisse im Selektor für interne Dokumente* im Referenzhandbuch *Anwender- und Lösungsverwaltung* und unter *Dokumente und Dateien der Alfabet -Anwendergemeinschaft zur Verfügung stellen* im Referenzhandbuch *Systemadministration*.

Endpunktname: `idocdownload`

HTTP-Methode: `POST`

Serviceaufruf:

`ServerAdress/api/v2/idocdownload`

Kopfzeilenfelder:

`Authorization: TypWert`



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

`Content-Type: application/json; charset=utf-8`

Nutzdaten:

Die Nutzdaten liegen als JSON-Objekt mit folgender Struktur vor:

```
{
  "Path": "IDOC:\\Folder\\Subfolder\\FileName.FileExtension"
}
```

Sie können folgende Felder enthalten:

Feld	Obligatorisch/Optional	Erforderlicher Wert
<code>"Path": " Path to file in IDOC "</code>	Obligatorisch	<p>Der Pfad zur IDOC-Datei, die heruntergeladen werden soll, einschließlich des Dateinamens. Der Pfad muss mit <code>IDOC:\</code> anfangen, und es müssen umgekehrte Schrägstriche zwischen den Verzeichnisnamen verwendet werden.</p> <p>Beachten Sie, dass umgekehrte Schrägstriche in JSON mit einem weiteren umgekehrten Schrägstrich geschützt werden müssen.</p> <p>Wenn Sie die JSON bei der Implementierung eines RESTful-Clients in den Code einbeziehen, müssen die für JSON erforderlichen umgekehrten Schrägstriche möglicherweise geschützt werden, um</p>




Feld	Obligato- risch/Op- tional	Erforderlicher Wert
		den Code ausführen zu können. Möglicherweise müssen Sie vier umgekehrte Schrägstriche zwischen Ordnernamen setzen.

Rückgabewert:

Der Rückgabewert des erfolgreichen Serviceaufrufs stellt die heruntergeladene Datei in einem Datenstrom bereit.

Hochladen von Dokumenten in die interne Dokumentenauswahl

Dieser Endpunkt kann verwendet werden, um Dateien in die **interne Dokumentenauswahl** der Alfabet-Datenbank oder einem ihrer Unterordner hochzuladen.

-  Informationen zur **internen Dokumentenauswahl** finden Sie unter *Hochladen von Dokumenten und Verwalten von Anwenderberechtigungen für Dokumentverzeichnisse im Selektor für interne Dokumente* im Referenzhandbuch *Anwender- und Lösungsverwaltung* und unter *Dokumente und Dateien der Alfabet -Anwendergemeinschaft zur Verfügung stellen* im Referenzhandbuch *Systemadministration*.
-  Das Hochladen von Dateien in die **interne Dokumentenauswahl** kann auch direkt über einen ADIF-Export in eine Datei erfolgen, der über den RESTful Service-Endpunkt `adifexport` ausgeführt wird. Der exportierte Inhalt wird dann nicht zum Herunterladen bereitgestellt, sondern direkt in der **internen Dokumentenauswahl** gespeichert. Informationen hierzu finden Sie unter [ADIF-Export über einen RESTful-Serviceaufruf starten](#).
-  Es gibt keinen spezifischen Endpunkt für das Löschen von Dateien aus der **internen Dokumentenauswahl**. Dateien in der internen Dokumentenauswahl sind Objekte der Objektklasse `ALFA_IDOCUMENT` und können wie jedes Objekt über den Endpunkt `delete` gelöscht werden. Sie können den Endpunkt `object` verwenden, um den `REFSTR` eines Dokuments mit einem bekannten Namen zurückzugeben. Der `REFSTR` wird benötigt, um das Objekt über den Endpunkt `delete` zu löschen.

Endpunktname : `idocupload`

HTTP-Methode : `POST`

Serviceaufruf:

`ServerAdress/api/v2/idocupload`

Kopfzeilenfelder:

`Authorization:TypeValue`

-  Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

Content-Type: multipart/form-data; charset=utf-8;

Nutzdaten:

Die Nutzdaten bestehen aus zwei Teilen in der angegebenen Reihenfolge:

- 1) Ein JSON-Objekt, das Details zur Speicherung der Datei in der internen Dokumentenauswahl bereitstellt.
- 2) Die hochzuladende Datei.



Auf welche Art die Nutzdaten hinzugefügt werden, hängt vom Client-Typ oder vom Clientcode-Typ ab.

Für RESTful Service-Client-Applikationen und objektorientierte Programmiersprachen wird der Inhalt in Form von zwei separaten Dateien hinzugefügt: einer JSON-Datei, die das JSON-Objekt enthält, und der Datei für den Upload.

Für Clients, die in zeichenfolgenorientierten Programmiersprachen implementiert sind, müssen die Nutzdaten mit einem im Kopfzeilenfeld Inhaltstyp definierten Trennzeichen beginnen und enden (Content-Type: multipart/form-data; charset=utf-8; delimiter: DelimiterStringNotPartOfAnyContent), und das Trennzeichen muss auch verwendet werden, um die beiden Teile der Nutzdaten voneinander zu trennen, wobei die JSON direkt zwischen den Trennzeichen definiert wird und die Importdaten als Dateistrom bereitgestellt werden. Die beiden Teile der Nutzdaten müssen in der angegebenen Reihenfolge wie folgt definiert sein:

```

---DelimiterStringNotPartOfAnyContent--
{
  "FolderPath": "IDOC:\\ImportFolder",
  "Name": "UploadFile.pdf",
  "OverwriteExistingFile": true
}
---DelimiterStringNotPartOfAnyContent--
file content stream data
---DelimiterStringNotPartOfAnyContent--
    
```

Die JSON kann folgende Felder enthalten:

Feld	Obligatorisch/Optional	Erforderlicher Wert
"FolderPath": " Path to folder in IDOC "	Obligatorisch	<p>Der Pfad zum IDOC-Ordner, in den die Datei hochgeladen werden soll. Der Pfad muss mit IDOC:\ anfangen, und es müssen umgekehrte Schrägstriche zwischen den Verzeichnisnamen verwendet werden.</p> <p>Beachten Sie Folgendes:</p> <ul style="list-style-type: none"> Umgekehrte Schrägstriche in JSON müssen mit einem weiteren umgekehrten Schrägstrich geschützt werden.

Feld	Obligato- risch/Op- tional	Erforderlicher Wert
		<p>Wenn Sie die JSON bei der Implementierung eines RESTful-Clients in den Code einbeziehen, müssen die für JSON erforderlichen umgekehrten Schrägstriche möglicherweise geschützt werden, um den Code ausführen zu können. Möglicherweise müssen Sie vier umgekehrte Schrägstriche zwischen Ordernamen setzen.</p> <ul style="list-style-type: none"> • Es ist nicht möglich, Dateien in das IDOC:-Root-Verzeichnis hochzuladen. • Der Dateiname in der internen Dokumentenauswahl muss nicht mit dem Namen der hochgeladenen Datei identisch sein.
<code>"Name": " File- Name.FileExten- tion "</code>	Obligato- risch	<p>Der Name der Datei, die in IDOC erstellt werden soll. Der Name muss nicht mit dem Namen der Datei identisch sein, die den Inhalt zum Hochladen bereitstellt. Die Dateierweiterung muss mit dem hochgeladenen Inhalt übereinstimmen.</p> <p>Beachten Sie, dass der Upload fehlschlägt, wenn im definierten Ordner der internen Dokumentenauswahl bereits eine Datei mit demselben Namen vorhanden ist und das Feld <code>OverwriteExistingFile</code> fehlt oder auf <code>false</code> gesetzt ist.</p>
<code>"OverwriteEx- isting- File":true false</code>	Optional	<p>Wenn dieses Feld auf <code>true</code> gesetzt ist, wird eine bereits vorhandene Datei mit demselben Namen im selben Ordner der internen Dokumentenauswahl durch die hochgeladene Datei überschrieben. Wenn diese Option auf <code>false</code> gesetzt ist, kann eine vorhandene Datei nicht überschrieben werden, und der Upload schlägt fehl, wenn im definierten Ordner der internen Dokumentenauswahl bereits eine Datei mit demselben Namen vorhanden ist. Der Standardwert ist <code>false</code>.</p>

Rückgabewert:

Wenn der Upload erfolgreich war, ist der Rückgabewert ein JSON-Objekt, das die IDOC-ID zurückgibt, wenn im hochgeladenen Dokument ein Feld `IDocID` enthalten ist.

Es wird geprüft, ob die Alfabet-Komponenten ausgeführt werden

Dieser Endpunkt kann verwendet werden, um zu prüfen, ob die Alfabet-Komponenten, die an der Ausführung einer RESTful-Serviceanfrage beteiligt sind, ausgeführt werden und ob auf sie zugegriffen werden kann. Der Serviceaufruf prüft die Verfügbarkeit der folgenden Komponenten:

- Die Alfabet-Webapplikation.

- den Datenbank-Server, auf dem die Alfabet-Datenbank ausgeführt wird.
- Wenn der Server-Alias der Alfabet-Webapplikation so konfiguriert ist, dass eine Verbindung zu einem Alfabet-Server hergestellt wird, wird die Verfügbarkeit dieses Alfabet-Servers geprüft.

Endpunktname: `monitor`

HTTP-Methode: `GET`

Serviceaufruf:

`ServerAdress/api/monitor`

Kopfzeilenfelder:

`Authorization: TypWert`



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

`Content-Type: application/json; charset=utf-8`

Nutzdaten

Keine Nutzdaten erforderlich.

Rückgabewert:

Der Rückgabewert ist ein JSON-Objekt, das Informationen zur Verfügbarkeit der Komponenten enthält.



Beispiel:

```
{
  "Name": "Alfabet WebApplication",
  "State": "Error",
  "Reason": "Alfabet WebApplication is running Database is running
Alfabet Application Server error: Unable to connect to the
application server. Check the network connections and resources.
Contact your administrator."
}
```

Aktualisierung des Metamodells

Dieser Endpunkt kann verwendet werden, um das Alfabet -Metamodell der Zieldatenbank mit der in einer AMM-Datei gespeicherten Metamodellkonfiguration zu aktualisieren.



Weitere Informationen über das Erzeugen von AMM-Dateien und das Aktualisieren des Metamodells und das Wiederherstellen des Metamodells finden Sie unter *Anwenden von Konfigurationsänderungen auf andere Datenbanken* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*.



Beachten Sie, dass die Alfabet-Webapplikation so konfiguriert werden muss, dass eine Verbindung zu einem ausgeführten Alfabet-Server hergestellt wird, um ADIF-Prozesse über die RESTful-Services ausführen zu können. Informationen hierzu finden Sie im Referenzhandbuch *Systemadministration*.

EndPoint-Name : updateMM

HTTP-Methode : POST

Serviceaufruf:

```
ServerAdress/api/v2/updateMM
```

Kopfzeilenfelder:

```
Authorization:TypeValue
```



Informationen zum erforderlichen Inhalt des Autorisierungsheaders finden Sie unter [Autorisierung](#).

```
Content-Type: multipart/form-data; charset=utf-8;
```

Nutzdaten:

Der Inhalt der AMM-Datei.



Auf welche Art die Nutzdaten hinzugefügt werden, hängt vom Client-Typ oder vom Clientcode-Typ ab.

Für RESTful Service-Client-Applikationen und objektorientierte Programmiersprachen wird der Inhalt in Form von zwei separaten Dateien hinzugefügt: einer JSON-Datei, die das JSON-Objekt enthält, und der AMM-Datei.

Für Clients, die in zeichenfolgenorientierten Programmiersprachen implementiert sind, müssen die Nutzdaten mit einem im Kopfzeilenfeld Inhaltstyp definierten Trennzeichen beginnen und enden (Content-Type: multipart/form-data; charset=utf-8; delimiter: DelimiterStringNotPartOfAnyContent), und das Trennzeichen muss auch verwendet werden, um die beiden Teile der Nutzdaten voneinander zu trennen, wobei die JSON direkt zwischen den Trennzeichen definiert und die AMM-Datei als Dateistrom bereitgestellt wird. Die beiden Teile der Nutzdaten müssen in der angegebenen Reihenfolge wie folgt definiert sein:

```
---DelimiterStringNotPartOfAnyContent---
{
  ???
  "Name": "FileName.amm",
  "Verbose": true
}
---DelimiterStringNotPartOfAnyContent---
file content stream data
---DelimiterStringNotPartOfAnyContent---
```

Rückgabewert:

Der Rückgabewert ist ein JSON-Objekt im folgenden Format:

```
{  
  "ResultMessage": "Configuration File has been sent to the Alfabet Server for  
  update.",  
  "Log": "",  
}
```

Folgende Informationen werden zurückgegeben:

- **ResultMessage:** Informiert über den Erfolg des Serviceaufrufs. Der Serviceaufruf übergibt die AMM-Datei an den Alfabet-Server. Im Anschluss an den RESTful-Serviceaufruf erfolgt dann die Aktualisierung des Metamodells durch den Alfabet-Server.
- **Log:** Dieses Feld ist nicht relevant.

Kapitel 7: Zugreifen auf die Alfabet -Benutzeroberfläche von externen Anwendungen

Die Alfabet RESTful API bietet Zugriff auf Daten in der Alfabet -Datenbank von externen Anwendungen aus, die diese Daten verarbeiten. Wenn die externe Anwendung für die Anwender Links zur Alfabet -Benutzeroberfläche enthält, können über eine spezielle Link-Syntax Links definiert werden, die definierte Ansichten der Alfabet -Benutzeroberfläche öffnen. Die Links zeigen entweder auf ein Objektprofil, ein Objekt-Cockpit oder eine grafische Ansicht für ein bestimmtes Objekt. In dem Link muss der `REFSTR`-Wert des Objekts angegeben werden, für das die Ansicht geöffnet wird. Der `REFSTR`-Wert für ein relevantes Objekt kann über die Alfabet RESTful API angefordert werden, um den Link zu erstellen. Der Endpunkt `objects` kann eine Liste der `REFSTR`-Werte für alle Objekte bereitstellen, die über eine Abfrage in einem konfigurierten Alfabet -Bericht gefunden werden.

Die erforderliche Link-Syntax sowie die Zugriffsrechte, die für Links von externen Anwendungen zur Alfabet -Benutzeroberfläche gelten, werden ausführlich im Abschnitt *Links zu Alfabet -Ansichten aus externen Applikationen* im Referenzhandbuch *Systemadministration* beschrieben.

Kapitel 8: Testen der Alfabet RESTful API

Für erste Tests des Alfabet RESTful API können Anfragen über einen marktüblichen REST-Client wie dem Advanced REST-Client von Google oder einem Swagger-Editor gesendet werden.

Sowohl für Tests mit einem marktüblichen REST-Client als auch für Tests kundenspezifischer REST-Clients gilt: Die Tests für Serviceaufrufe für die EndPoint-Aktualisierung sollten nur mit Alfabet -Testdatenbanken und nicht mit der produktiven Alfabet-Datenbank durchgeführt werden.

Dieses Kapitel enthält Informationen zur Durchführung eines einfachen Tests eines Serviceaufrufs und beschreibt, wie Fehlercodes zu interpretieren sind, die möglicherweise während des Tests angezeigt werden:


- [Testen der Alfabet RESTful API](#)
- [Konfigurationen, die für die Verwendung eines Swagger-Editors zum Testen erforderlich sind](#)

Testen der Alfabet RESTful API

Für einfache Tests von Serviceaufrufen für die Alfabet RESTful API können Sie marktübliche REST-Clients verwenden. Dazu sind keine zusätzlichen Konfigurationen erforderlich, außer bei Swagger-Editoren.

Konfigurationen, die für die Verwendung eines Swagger-Editors zum Testen erforderlich sind

Ein Swagger-Editor kann zum Testen verwendet werden. Um alle Aufrufe in einem gebrauchsfertigen Swagger-Format anzuzeigen, müssen die Alfabet RESTful Services und der Webbrowser wie folgt konfiguriert werden:

- 1) Öffnen Sie den Alfabet Administrator.
- 2) Klicken Sie im Explorer auf den Knoten **Alfabet-Aliase**. Ein Arbeitsbereich mit einer Symbolleiste wird geöffnet.
- 3) Klicken Sie in der Symbolleiste auf **Tools > alfabet.config konfigurieren**. Ein Editor wird geöffnet.
- 4) Klicken Sie auf die Schaltfläche „Durchsuchen“  rechts neben dem Feld **Webordner**, und wählen Sie im Verzeichnisbrowser das Hauptverzeichnis der Alfabet-Webapplikation aus. Die Datei `alfabet.config` im Unterverzeichnis **config** des ausgewählten Verzeichnisses wird im Editor geöffnet.
- 5) Geben Sie den folgenden Code als untergeordnetes Element des XML-Elements `alfaSection` ein, um den `WebApplicationFolderPath` durch den absoluten Pfad zum virtuellen Verzeichnis der Alfabet-Webapplikation zu ersetzen:

```
<add key="SwaggerSpecFileName"
value="WebApplicationFolderPath\SwaggerSpec\AlfabetWeb5_SwaggerSpec.json"
/>
```

- 6) Klicken Sie auf **Speichern**. Die Änderung wird gespeichert und der Editor geschlossen.
- 7) Schließen Sie den Alfabet Administrator.
- 8) Rufen Sie die URL `http:// URLOfTheAlfabetWebApplication /swagger/docs/v2` auf, und kopieren Sie den angezeigten JSON-Code.

- 9) Öffnen Sie Ihren Swagger-Editor, und fügen Sie den JSON-Code in das Code-Feld des Editors ein.
- 10) Vergewissern Sie sich, dass in Ihrem Browser **Cross-Origin Resource Sharing** aktiviert ist.



Die Kopfzeile: `Access-Control-Allow-Origin: *` muss in den vom Server generierten Aufrufen enthalten sein. Das Aktivieren von "Cross-Origin Resource Sharing" im Browser ist eine einfache Methode, dies zu erreichen, falls es nicht anderweitig im System implementiert ist.

Kapitel 9: Überprüfen des Erfolgs von Serviceaufrufen an die Alfabet-RESTful-API

Der Rückgabewert für den Serviceaufruf der RESTful-Services informiert über den erfolgreichen Abschluss des Serviceaufrufs. Wenn ein Fehler auftritt, informiert die entsprechende Fehlermeldung über den Grund für das Fehlschlagen des Aufrufs. Die Fehlermeldungen werden in einem JSON-Objekt mit zwei Feldern übergeben:

- `ErrorMessage`: Informationen über den Grund des Fehlschlags.
- `ErrorCode`: Der Fehlercode des Web-Servers.

Aus Sicherheitsgründen sind die an den REST-Client zurückgegebenen Fehlermeldungen nicht sehr detailliert. Zudem informieren Serviceaufrufe, die Aktivitäten wie ADIF-Ausführungen asynchron auslösen, lediglich über das erfolgreiche Auslösen des Prozesses, ohne Informationen über den Erfolg der Ausführung des ADIF-Prozesses zu liefern. Die Ausführung von Prozessen, die über die RESTful-Services ausgelöst werden, kann zudem im Falle synchroner Prozesse von der Alfabet-Webapplikation oder im Falle asynchroner Prozesse vom Alfabet-Server protokolliert werden.

Die Konfiguration der Alfabet-Komponenten, ob Protokollinformationen in eine zentrale Protokolldatei geschrieben oder an einen externen Protokollserver gesendet werden, ist im Abschnitt *Zentrale Protokollierung der Funktionalität für Alfabet-Komponenten* im Referenzhandbuch *Systemadministration* beschrieben.

Index

ADIF-Export	
Starten	106
Adifexport-Endpunkt	106
ADIF-Import	
Starten	98
Adifimport-Endpunkt	98
ADIF-Schema	
Für REST-API aktivieren	20
Aktivieren des REST API v2	13
Aktivierung	9
Aktualisieren von Objekten	
Update-Endpunkt	85
Alfabet.config	
JSON Web-Token	10
Max_api_requests_per_second	10
Alfabet-Anwender	
Anonymisieren	95
Erneutes Generieren des Passworts	93
Alfabet-Benutzeroberfläche	126
Alfabet-Komponenten	
Überwachung des Funktionszustands	123
Alfabet-Metamodell	
Lesen der Landeseinstellungen	39
Lesen des Objektklassenmodells	39, 44
Lesen von Aufzählungen	48
Alfabet-Objekt	
Aktualisieren von Eigenschaftswerten	85
Ändern der Daten	79
Archivieren	90
Archivieren und Löschen	90
Erzeugen	79, 82
Erzeugen einer Beziehung	89
Erzeugen einer Referenz	89
Lesen von Daten	50
Löschen	77
Ändern von Objektdaten	85
Anforderungen	
Siehe Technische Anforderungen	8
Anfrage	

Grenzwert	10
Anonymizeuser-Endpunkt	95
Ansichtsschema	
REST-Zugriffsberechtigung	22
Anwender	
Anonymisieren	95
Erneutes Generieren des Passworts	93
Zugriffsberechtigung	17
Anwenderbasierte Autorisierung	
Siehe Autorisierung	16
Anwendereditor	
REST-API-Einstellungen	17
Anwender-Passwort	16
Anwenderprofil	
REST-Zugriffsberechtigung	22
API-Landeseinstellung	33
API-Passwort generieren	17
API-Zugriffsoptionen	17
Archiveobject-Endpunkt	
HTTP-Methode	90
JSON-Text	90
Rückgabewert	90
Serviceaufruf	90
Aufzählung	
Lesen der Metamodell-Struktur	48
Authentifizierung	
Web-Server	12
Autorisierung	
Anfrage	16
Anwender-Passwort	16
Anwenderschlüssel	16
API-Zugriffsoptionen	17
Clientseitig	27
Token-Anfrage	27
Berechtigung	
Siehe Zugriffsberechtigung	22
Beziehung	
Erzeugen zwischen Objekten	89
Löschen	77
Blacklist	21
Classes-Endpunkt	

HTTP-Methode	44
Parameter	44
Rückgabewert	44
Serviceaufruf	44
Dateiliste	
Interne Dokumentenauswahl	117
Datenbankobjekt	
Archivieren	90
Archivieren und Löschen	90
Lesen von Daten	50
Löschen	77
Datumsformat	33
Delete-Endpunkt	
HTTP-Methode	77
JSON-Text	77
Rückgabewert	77
Serviceaufruf	77
Dokument	
Herunterladen	119
Hochladen	120
Liste	117
Emptyvalues-Parameter	
Für Endpunkt-Aufzählungen	48
Für Endpunktklassen	44
Für metamodel-Endpunkt	39
Endpunkt	
Adifexport	106
Adifimport	98
Anonymizeuser	95
Archiveobject	90
Classes	44
Enums	48
Idocdownload	119
Idocfilelist	117
Idocupload	120
Löschen	77
Metamodell	39
Monitor	123
Objekte	50
Regeneratepassword	93
Token	27
Update	79
Versionen	8
Workflow	97
Enums-Endpunkt	

HTTP-Methode	48
Parameter	48
Rückgabewert	48
Serviceaufruf	48
Erzeugen von Beziehungen	
Update-Endpunkt	89
Erzeugen von Objekten	
Update-Endpunkt	82
Für REST-API verwendbar	20
Google Advanced REST-Client	127
Grenzwert	
Anfragen pro Sekunde	10
Hat API V2-Zugriff	17
Herunterladen	
Interne Dokumentenauswahl	119
Hochladen	
Interne Dokumentenauswahl	120
HTTP-Header	
Autorisierung	16
HTTP-Methode	
Archiveobject-Endpunkt	90
Classes-Endpunkt	44
Delete-Endpunkt	77
Enums-Endpunkt	48
Metamodel-Endpunkt	39
Objects-Endpunkt	50
Regeneratepassword-Endpunkt	93
Update-Endpunkt	79
IDOC	
Datei herunterladen	119
Datei hochladen	120
Dateiliste	117
Siehe	21
Idocdownload-Endpunkt	119
Idocfilelist-Endpunkt	117
Idocupload-Endpunkt	120
Interne Dokumentenauswahl	
Datei herunterladen	119
Datei hochladen	120
Lesen des Inhalts	117
Zugriffsberechtigungen	21
JSON Web-Token	10
JSON-Anfrage	

Für die Erzeugung neuer Objekte	82
JSON-Format	
In Serviceaufruf für update-Endpoint	82, 85, 89
Rückgabewert	36
Zum aktualisieren von Objektdaten	85
Zum Erzeugen von Beziehungen	89
Klasseneinstellung	
REST-Zugriffsberechtigung	22
Klassenschlüssel	
Lesen aus dem Metamodell	39
Lesen der Metamodell-Struktur	44
Konfigurierter Bericht	
Für object-Endpoint	63
Für REST-API aktivieren	20
Landeseinstellungen	
Lesen aus dem Metamodell	39
Lesen von Objektdaten für Landeseinstellungen	50
Leere Werte	
Einschließen in classes-Endpoint	44
Einschließen in enums-Endpoint	48
Einschließen in metamodel-Endpoint	39
Lese-/Schreibberechtigung	22
Leseberechtigung	22
Lesen über REST-API zulassen	22
Lizenzen	8
Mandant	25
Max_api_requests_per_second	10
Maximale Anzahl von Anfragen pro Sekunde	
ändern	10
Metamodel-Endpoint	
HTTP-Methode	39
Parameter	39
Rückgabewert	39
Serviceaufruf	39
Names-Parameter	
Für Endpoint-Aufzählungen	48
Für Endpunktklassen	44
Objects-Endpoint	

HTTP-Methode	50
Objekte nach Bericht	62
Objekte nach Filter	56
Objekte nach Referenzen	51
Parameter	50
Rückgabewert	50
Serviceaufruf	50
Objekt	
Archivieren	90
Archivieren und Löschen	90
Erzeugen	79
Lesen von Daten für Objekt	50
Löschen	77
Objekt in der Alfabet-Datenbank	
Aktualisieren von Eigenschaftswerten	85
Ändern der Daten	79
Archivieren	90
Archivieren und Löschen	90
Erzeugen	82
Erzeugen einer Beziehung	89
Erzeugen einer Referenz	89
Lesen von Daten	50
Löschen	77
Objektdaten	
Exportieren für Objekte nach REFSTR	51
Exportieren über Abfrage	63
Exportieren über Berichtsabfrage	62
Exportieren von Objekten nach Filter	56
Lesen	50
Lesen verschiedener Sprachversionen	50
Objektklasse	
Aktualisieren von Eigenschaftswerten	85
Ändern von Objektdaten	79
Erzeugen eines neuen Objekts	82
Lesen der Metamodell-Struktur	39, 44
Objekt erzeugen	79
Objektklasseneigenschaft	
Typ ReferenceArray	89
Objektklasseneigenschaften	
Ändern für ein Objekt	79
Lesen aus dem Metamodell	39
Lesen der Metamodell-Struktur	44
Obligatorisch für neues Objekt	82
Objekt-REFSTR	
Exportieren über Berichtsabfrage	62
Parameter	

Emptyvalues	39, 44, 48
Namen	44, 48
Passwort	16
Neugenerierung für Anwender	93
Protokollierung	
REST-Serviceaufrufe	129
ReferenceArray	89
Referenz	
Löschen	77
Regeneratepassword-Endpunkt	
HTTP-Methode	93
JSON-Text	93
Rückgabewert	93
Serviceaufruf	93
RELATIONS-Tabelle	
Löschen des Eintrags	77
REST-Client	
Zum Testen	127
RESTful-Serviceaufruf	
Grenzwert für eingehende	10
Return value	
Containing object property values	74
Containing REFSTR list	76
Rückgabewert	
Adifexport	106
Adifimport	98
Für classes-Endpunkt	44
Für Endpunkt	89
Für Endpunkt-Aufzählungen	48
Für metamodel-Endpunkt	39
Für objects-Endpunkt	50
Für update-Endpunkt	82
Idocdownload	119
Idocfilelist	117
Idocupload	120
Mit Berichtsinhalten	74
Mit Objekteigenschaftswerten	51, 56
Monitor	123
Objects-Endpunkt	50
Objekte nach Bericht	62
Objekte nach Filter	56
Objekte nach Referenzen	51
RESULT-Objekt	36
Workflow	97
Schlüssel	16

Schreibberechtigung	22
Schreiben über REST-API zulassen	22
Server-Alias	
Aktivieren der REST API	13
Serverrollen	11
Serviceaufruf	
Adifexport	106
Adifimport	98
Anonymizeuser-Endpunkt	95
Archiveobject-Endpunkt	90
Ausführen eines konfigurierten Berichts	64
Autorisierung	27
Classes-Endpunkt	44
Datumsformat	33
Delete-Endpunkt	77
Enums-Endpunkt	48
Idocdownload	119
Idocfilelist	117
Idocupload	120
Metamodel-Endpunkt	39
Monitor	123
Objects-Endpunkt	50
Objekte nach Bericht	62
Objekte nach Filter	56
Objekte nach Referenzen	51
Protokollierung	129
Regeneratepassword-Endpunkt	93
Rückgabewert	36
Syntax	36
Uhrzeitformat	33
Workflow	97
Service-Aufruf	
Update-Endpunkt	79
Sitzungs-Cookies	6
Swagger	127
Technische Anforderungen	
Konfiguration	9
Lizenzen	8
Web-Server	11
Token	16, 27
Uhrzeitformat	33
Update-Endpunkt	

Aktualisieren von Objektdaten	85
Erzeugen neuer Objekte	82
Erzeugen von Beziehungen	89
HTTP-Methode	79
JSON-Nutzdaten	79
Rückgabewert	79
Serviceaufruf	79
Verfügbarkeitsprüfung	
Alfabet-Komponenten	123
Verknüpfung	
Zur Alfabet-Benutzeroberfläche	126
Web.config	
Handler für RESTful-Dienste	9
WebDAV	11
Web-Server	
Anforderungen	11
Authentifizierung	12
Serverrollen	11
WebDAV	11
Whitelist	21
Workflow	
Starten	97
Workflow-Endpunkt	97
Workflow-Vorlage	
Für REST-API aktivieren	20
Zugriffsberechtigung	
Anwender	17
Objektklasse	22
Pro Mandant	25
Pro Objekt	26
Übersicht	28
Zugriffsberechtigungen	
ADIF-Schema	20
Interne Dokumentenauswahl	21
Konfigurierter Bericht	20
Workflow	20
Zurücksetzen	
Anwender-Passwort	93
Zustandsüberwachung	
Alfabet-Komponenten	123