



**software** AG

# Web-Services für Alfabet

Alfabet-Referenzhandbuch

---

Dokumentationsversion Alfabet 10.13.0

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
<b>Fett</b>	<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 <b>Fertigstellen</b>.</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>&lt;XmlElement XmlAttribute="Anwendername"/&gt;</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 <b>Datei &gt; Beenden</b></p>
< >	<p>Steht für Variablen, die vom Anwender eingegeben werden.</p> <p>Beispiel: Erzeugen Sie einen neuen Anwender und geben Sie &lt;Anwendername&gt; 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

<b>Kapitel 1: Arbeiten mit den Alfabet -Webdiensten</b>	<b>5</b>
Verwenden der Alfabet -Webdienste für die Entwicklung einer Schnittstelle zum Lesen aus der oder zum Schreiben in die Alfabet-Datenbank	5
Erläuterungen zur WSDL der Alfabet -Webdienste	7
<b>Kapitel 2: Authentifizierung der Alfabet -Webdienste</b>	<b>9</b>
Authentifizierung der Webdienst-Clients	9
Integrierte Windows-Authentifizierung	10
<b>Kapitel 3: Für die Alfabet -Webdienste implementierte Methoden</b>	<b>12</b>
GetInstancesByQuery	13
GetInstancesByGuid	15
GetClassInstances	16
GetDataSetByQuery	17
GetDataSetByNativeQuery	19
UpdateData	20
UpdateInstance	22
CreateInstance	25
DeleteInstance	27
CreateRelation	28
DeleteRelation	30
<b>Index</b>	<b>33</b>

## Kapitel 1: Arbeiten mit den Alfabet -Webdiensten

Mit den Alfabet -Webdiensten kann man aus externen Programmen, die auch auf anderen Plattformen laufen können, direkt auf die in der Alfabet-Datenbank gespeicherten Daten zugreifen. Die Plattforminteroperabilität basiert auf dem SOAP-Protokoll und bietet die Möglichkeit, von Programmen, die in anderen Programmiersprachen geschrieben sind und auf anderen Betriebssystemen laufen, über HTTP Remoteprozeduraufrufe an die Alfabet-Datenbank zu senden.

Es wird empfohlen, die Alfabet -Webdienste nur für die kontinuierliche Datenintegration in oder von einer Drittapplikation zu verwenden. Einmalige oder seltene Batch-Datenexporte bzw. -importe zur Alfabet-Datenbank sollten über das Alfabet Data Integration Framework (ADIF) durchgeführt werden.

Alfabet bietet zwei Webdienste

- **AlfaQueryWebService** zum Lesen von Daten aus der Alfabet-Datenbank
- **AlfaUpdateDataWebService** zur Anwendung von Änderungen auf die Alfabet-Datenbank

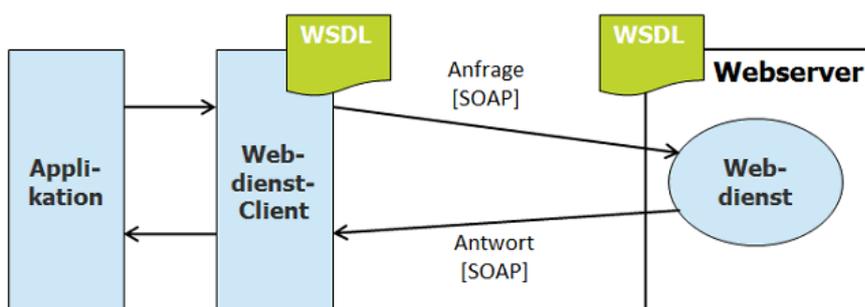
## Verwenden der Alfabet -Webdienste für die Entwicklung einer Schnittstelle zum Lesen aus der oder zum Schreiben in die Alfabet-Datenbank

Die Alfabet -Webdienste müssen auf einem Web-Server implementiert werden, damit sie als XML-basierte Schnittstelle für den Zugriff auf die Alfabet-Datenbank über externe Applikationen dienen können.

Der "Vertrag" zwischen Client und Server wird durch die Web Service Description Language (WSDL) repräsentiert, die die Methoden beschreibt, die zum Lesen oder Schreiben von Daten in die Alfabet-Datenbank verwendet werden können, sowie das XML-Format, das von den Methoden zurückgegeben wird.

Aufrufe zu den Methoden der Dienste von anderen Programmen oder HTML-Formularen über die Alfabet Webdienste basieren auf dem SOAP-Format. Auf Serverseite leiten die Alfabet -Webdienste Aufrufe zum laufenden Alfabet -Applikationsserver um und geben als Ergebnis XML-Dokumente zurück.

Clientseitig ist die Implementierung eines Webdienst-Clients erforderlich, um eine Applikation auf einem Webdienst zu erstellen. Der Webdienst-Client schließt die Lücke zwischen den Webdienstverfahren und der tatsächlichen Programmiersprache der externen Applikation. Er assembliert und disassembliert die SOAP-Nachrichten und dient als Proxy für die Webdienstmethoden.





Sie müssen wie folgt vorgehen, um die Alfabet -Webdienste für den Zugriff auf die Alfabet-Datenbank zu verwenden:

- Richten Sie die Alfabet -Webdienste auf einem Web-Server als Web-Applikation ein.  
  
Das Verfahren für die Einrichtung hängt vom verwendeten Web-Server ab. In der Dokumentation zu Ihrem Web-Server erhalten Sie Informationen darüber, wie Sie einen Webdienst einrichten.
- Erstellen Sie einen Webdienst-Client basierend auf der WSDL der Alfabet -Webdienste.  
  
Der Webdienst-Client wird in der Regel mithilfe einer Webdienst-Client-Engine erstellt. Informationen dazu finden Sie in der Dokumentation zu der Webdienst-Client-Engine, die implementiert wird.
- Implementieren Sie die Kommunikation mit dem Webdienst-Client und die Verarbeitung der Daten, die über die Methoden der Alfabet -Webdienste zurückgegeben oder gesendet werden.

Dieses Dokument enthält Informationen zur Webdienst-WSDL, die für die Implementierung einer externen Applikation basierend auf den Alfabet -Webdiensten erforderlich ist, und beschreibt die von den Alfabet -Webdiensten verwendeten Methoden sowie die XML-Ausgabe, die aus den Aufrufen zu den Alfabet -Webdiensten resultiert.

Wenn Sie eine Client-Applikation mithilfe der Alfabet -Webdienste erstellen, berücksichtigen Sie bei der Implementierung der externen Applikation Folgendes:

- Das Aufrufprogramm muss Ausnahmen handhaben, die im Dienst auftreten können, um den Erfolg des Methodenaufrufs zu steuern.
- Die Methoden geben als Antwort auf die Aufrufe Nur-XML zurück. Das XML muss zu In-Memory-Datenmodellen innerhalb der Applikation auf Clientseite übertragen werden.



Sonderzeichen werden in der XML-Ausgabe ignoriert. Weitere Informationen zu den verfügbaren Methoden und der resultierenden XML-Ausgabe finden Sie unter [Für die Alfabet -Webdienste implementierte Methoden](#).

- Die Alfabet -Webdienste überprüfen nicht die Datenintegrität auf Client- und Serverseite. Es ist Aufgabe des Architekten der Applikation, für Datenintegrität zu sorgen.
- Die Alfabet -Webdienste führen eine einfache Authentifizierung basierend auf dem Datenbankzugriffsmodus entweder über die Windows-Authentifizierung oder die Authentifizierung einzelner Anwender anhand des Anwernamens und Passworts für die Datenbank durch. Die Sicherheit lässt sich mit folgenden Methoden verbessern:
  - Zusätzliche Sicherheitsmechanismen des Web-Servers (z. B. kann eine Verschlüsselung via HTTPS implementiert werden).
  - Die Implementierung der Applikation auf Clientseite kann zusätzliche Sicherheitsmechanismen beinhalten, wie z. B. eine Kontrolle des Kundenzugriffs via LDAP.



Weitere Informationen zu den Sicherheitsmechanismen, die von den Alfabet -Webdiensten bereitgestellt werden, finden Sie unter [Authentifizierung der Alfabet -Webdienste](#).

## Erläuterungen zur WSDL der Alfabet -Webdienste

WSDL ist die Standardmodellierungssprache zur Beschreibung von Webdiensten.

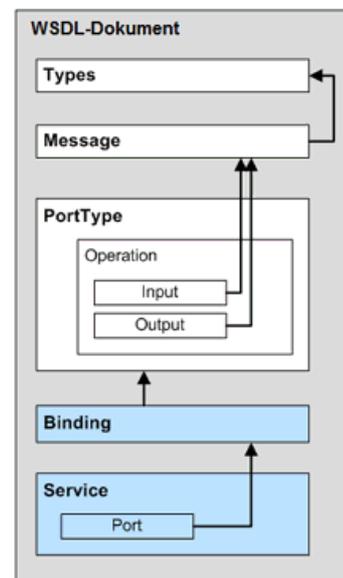
WSDL-Beschreibungen für die Alfabet -Webdienste stehen in der Standardschnittstelle der Alfabet -Webdienste zur Verfügung:

- ***AlfaQueryWebService.asmx?wsdl*** und
- ***AlfaUpdateDataWebService.asmx?wsdl***

Die WSDL-Beschreibung wird in einem XML-Dokument bereitgestellt, das mit dem W3C-WSDL-Standard konform ist. Das Dokument enthält Informationen zu:

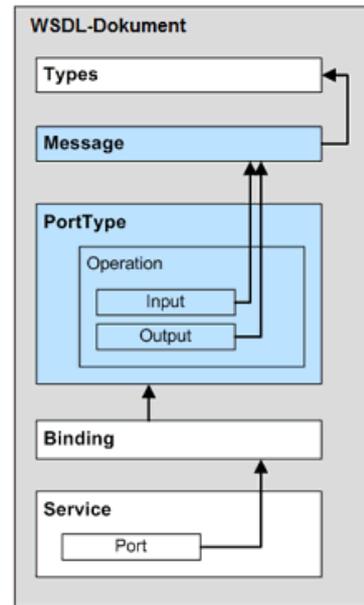
- dem Ort des Dienstes (*Service, Binding*), der für die SOAP-basierte Kommunikation relevant ist.

```
<?xml version="1.0" encoding="utf-8" ?>
- <wsdl:definitions xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:tns="http://alfabet.com/webservices/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  targetNamespace="http://alfabet.com/webservices/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
+ <wsdl:types>
+ <wsdl:message name="GetClassInstancesSoapIn">
+ <wsdl:message name="GetClassInstancesSoapOut">
+ <wsdl:message name="GetDataSetByQuerySoapIn">
+ <wsdl:message name="GetDataSetByQuerySoapOut">
+ <wsdl:message name="GetInstancesByQuerySoapIn">
+ <wsdl:message name="GetInstancesByQuerySoapOut">
+ <wsdl:message name="GetInstancesByGuidSoapIn">
+ <wsdl:message name="GetInstancesByGuidSoapOut">
+ <wsdl:message name="GetInstancesByReferenceSoapIn">
+ <wsdl:message name="GetInstancesByReferenceSoapOut">
+ <wsdl:portType name="AlfaQueryWebServiceSoap">
- <wsdl:binding name="AlfaQueryWebServiceSoap" type="tns:AlfaQueryWebServiceSoap">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
+ <wsdl:operation name="GetClassInstances">
+ <wsdl:operation name="GetDataSetByQuery">
+ <wsdl:operation name="GetInstancesByQuery">
+ <wsdl:operation name="GetInstancesByGuid">
+ <wsdl:operation name="GetInstancesByReference">
</wsdl:binding>
+ <wsdl:binding name="AlfaQueryWebServiceSoap12" type="tns:AlfaQueryWebServiceSoap">
- <wsdl:service name="AlfaQueryWebService">
  <wsdl:port name="AlfaQueryWebServiceSoap" binding="tns:AlfaQueryWebServiceSoap">
    <soap:address location="http://localhost:8080/pit61WS/AlfaQueryWebService.asmx" />
  </wsdl:port>
+ <wsdl:port name="AlfaQueryWebServiceSoap12" binding="tns:AlfaQueryWebServiceSoap12">
</wsdl:service>
</wsdl:definitions>
```



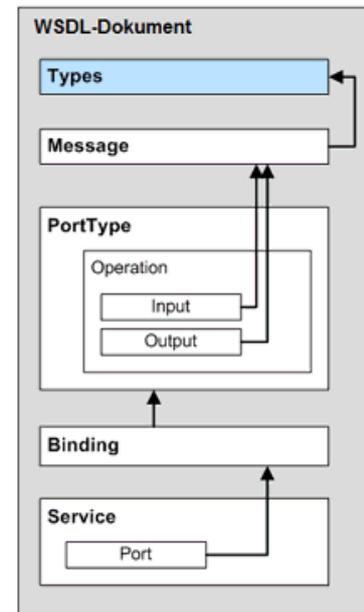
- den unterstützten Operationen und Nachrichten (*PortType, Messages*), die für die Definition der Anfragen und Antworten, die über SOAP bereitgestellt werden, relevant sind.

```
<?xml version="1.0" encoding="utf-8" ?>
- <wsdl:definitions xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:tns="http://alfabet.com/webservices/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  targetNamespace="http://alfabet.com/webservices/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
+ <wsdl:types>
- <wsdl:message name="GetClassInstancesSoapIn">
  <wsdl:part name="parameters" element="tns:GetClassInstances" />
</wsdl:message>
- <wsdl:message name="GetClassInstancesSoapOut">
  <wsdl:part name="parameters" element="tns:GetClassInstancesResponse" />
</wsdl:message>
+ <wsdl:message name="GetDataSetByQuerySoapIn">
+ <wsdl:message name="GetDataSetByQuerySoapOut">
+ <wsdl:message name="GetInstancesByQuerySoapIn">
+ <wsdl:message name="GetInstancesByQuerySoapOut">
+ <wsdl:message name="GetInstancesByGuidSoapIn">
+ <wsdl:message name="GetInstancesByGuidSoapOut">
+ <wsdl:message name="GetInstancesByReferenceSoapIn">
+ <wsdl:message name="GetInstancesByReferenceSoapOut">
- <wsdl:portType name="AlfaQueryWebServiceSoap">
  <wsdl:operation name="GetClassInstances">
    <wsdl:input message="tns:GetClassInstancesSoapIn" />
    <wsdl:output message="tns:GetClassInstancesSoapOut" />
  </wsdl:operation>
+ <wsdl:operation name="GetDataSetByQuery">
+ <wsdl:operation name="GetInstancesByQuery">
+ <wsdl:operation name="GetInstancesByGuid">
+ <wsdl:operation name="GetInstancesByReference">
</wsdl:portType>
+ <wsdl:binding name="AlfaQueryWebServiceSoap" type="tns:AlfaQueryWebServiceSoap">
+ <wsdl:binding name="AlfaQueryWebServiceSoap12" type="tns:AlfaQueryWebServiceSoap">
+ <wsdl:service name="AlfaQueryWebService">
  <wsdl:definitions>
```



- den Methodentypen (Type), die die Methoden beschreiben, die mit den Webdiensten implementiert werden.

```
<?xml version="1.0" encoding="utf-8" ?>
- <wsdl:definitions xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:tns="http://alfabet.com/webservices/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  targetNamespace="http://alfabet.com/webservices/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
- <wsdl:types>
  <xs:schema elementFormDefault="qualified" targetNamespace="http://alfabet.com/webservices/">
    <xs:element name="GetClassInstances">
      <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="1" name="aServerName" type="s:string" />
          <xs:element minOccurs="1" maxOccurs="1" name="bEnsureSecurity" type="s:boolean" />
          <xs:element minOccurs="0" maxOccurs="1" name="aUserName" type="s:string" />
          <xs:element minOccurs="0" maxOccurs="1" name="aPassword" type="s:string" />
          <xs:element minOccurs="0" maxOccurs="1" name="aProfileName" type="s:string" />
          <xs:element minOccurs="0" maxOccurs="1" name="aClassName" type="s:string" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="GetClassInstancesResponse">
    <xs:element name="GetDataSetByQuery">
    <xs:element name="GetDataSetByQueryResponse">
    <xs:element name="GetInstancesByQuery">
    <xs:element name="GetInstancesByQueryResponse">
    <xs:element name="GetInstancesByGuid">
    <xs:element name="GetInstancesByGuidResponse">
    <xs:element name="GetInstancesByReference">
    <xs:element name="GetInstancesByReferenceResponse">
  </xs:schema>
</wsdl:types>
- <wsdl:message name="GetClassInstancesSoapIn">
  <wsdl:part name="parameters" element="tns:GetClassInstances" />
</wsdl:message>
- <wsdl:message name="GetClassInstancesSoapOut">
  <wsdl:part name="parameters" element="tns:GetClassInstancesResponse" />
```



Die meisten der Informationen in dem XML-Dokument werden von der Webdienst-Client-Engine beim Erstellen des Webdienst-Clients interpretiert. Die Informationen zu den Methodenaufrufen sind für die Implementierung der Applikation auf Clientseite relevant und werden ausführlich im Abschnitt [Für die Alfabet - Webdienste implementierte Methoden](#) beschrieben. Die Informationen umfassen die unterstützten Methodenaufrufe, die Argumente, die mit den Methodenaufrufen übermittelt werden müssen, sowie die Struktur des zurückgegebenen XML.

## Kapitel 2: Authentifizierung der Alfabet-Webdienste

Die Alfabet -Webdienste verbinden sich mit der Alfabet-Datenbank über den Alfabet-Server. Dennoch erfolgt die Authentifizierung direkt auf Datenbankebene. Dazu werden ein Anwendername und ein Passwort für die Anmeldung bei der Datenbank auf Ebene des Datenbank-Servers bereitgestellt.

Sollte die Authentifizierung des Webdienst-Clients bei der Datenbank nicht wünschenswert oder verfügbar sein, steht ein zweiter Mechanismus zur Verfügung, um sicherzustellen, dass vorhandene Clients weiter ausgeführt werden können. Dieser Mechanismus basiert auf der integrierten Windows-Authentifizierung zwischen Applikationsserver und Datenbank-Server. Im Szenario II finden Sie eine ausführliche Beschreibung der an diesem Authentifizierungsszenario beteiligten Mechanismen.

Die folgenden Abschnitte bieten eine detaillierte Übersicht über diese Prozesse und die beteiligten Mechanismen:

- [Authentifizierung der Webdienst-Clients](#)
- [Integrierte Windows-Authentifizierung](#)

### Authentifizierung der Webdienst-Clients

In diesem Szenario ist die Authentifizierung bei der Datenbank und somit die Autorisierung der übermittelten Anfrage in Webdienst-Clientanfragen eingeschlossen. Jeder Webdienst-Clienttyp kann mit unterschiedlichen Anmeldeinformationen arbeiten. Folglich können ihm verschiedene Berechtigungen hinsichtlich verfügbarer Tabellen, Ansichten oder Statements bereitgestellt werden.



Das SOAP-Protokoll, das verwendet wird, um Nachrichten zwischen Webdienst-Clients und dem Web-Server, auf dem der Alfabet -Webdienst gehostet wird, auszutauschen, enthält keine Möglichkeiten für die Verschlüsselung sensibler Informationen. Diese Aufgabe übernimmt das Protokoll, das auf untergeordneten Layers verwendet wird (z. B. in der Applikations- oder Transportschicht). Software AG empfiehlt, den Zugriff auf den Alfabet -Webdienst ausschließlich über HTTPS bereitzustellen, um eine sichere Kommunikation zu gewährleisten.

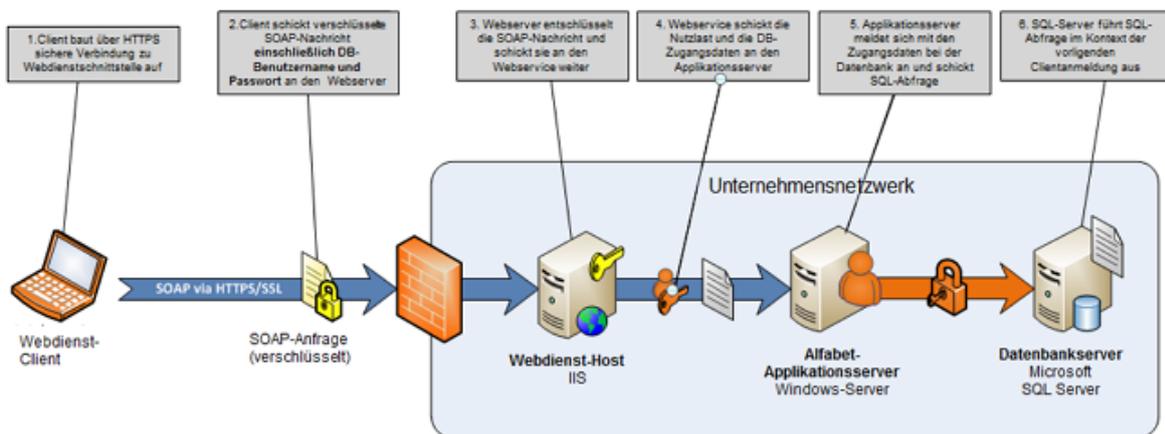
In der folgenden Liste wird die Verarbeitung von Webdienstanfragen mittels Clientauthentifizierung beschrieben (siehe Abbildung unten):

- 1) Die Implementierung des Webdienst-Clients umfasst eine SOAP-Nachricht, die alle für die angefragte Operation erforderlichen Parameter enthält. Hierzu gehören Anwendername und Passwort für das Datenbankkonto, das für die Authentifizierung und Autorisierung der Ausführung der erforderlichen Statements auf die Datenbank verwendet werden soll.
- 2) Über das Serverzertifikat, das vom Webdienst-Host bereitgestellt wird, verschlüsselt der Webdienst-Client die SOAP-Nachricht und übermittelt sie im Body der HTTP-Nachricht zum Webdienst-Host.
- 3) Der Webdienst-Host entschlüsselt den Body der Nachricht und leitet die SOAP-Nachricht zur Verarbeitung an die Microsoft® Internetinformationsdienste® im Webdienst-Anforderungshandler weiter.
- 4) Bei der Implementierung des Webdienstes wird die SOAP-Nachricht entpackt, und es wird ein Aufruf zur entsprechenden Schnittstelle des Applikationsservers vorbereitet, die die angeforderte Operation bereitstellt. Daraufhin werden die Parameter der angeforderten Operation sowie die Anmeldeinformationen an den Alfabet-Server übermittelt.

- 5) Der Alfabet-Server verwendet die bereitgestellten Anmeldeinformationen, um sich mit dem Datenbank-Server zu verbinden. Wenn der Verbindungsversuch erfolgreich war, werden die Statements, die zur Ausführung der angeforderten Operation erforderlich sind, an die Datenbank übermittelt. Die Datenbank führt diese Statements dann mit der bereitgestellten Identität und Autorisierung aus. Wenn kein Fehler auftritt, werden die Daten zurück an den aufrufenden Webdienst-Host gesendet und schließlich via HTTPS an den anfordernden Webdienst-Client zurückgegeben.



Bei diesem Szenario wird davon ausgegangen, dass der Alfabet-Server sich mit dem Datenbank-Server über die Clientauthentifizierung verbindet und die Windows-Authentifizierung auf dem Datenbank-Server deaktiviert ist. Wenn die Alfabet -Webdienste die Clientauthentifizierung, der Alfabet-Server aber die Windows-Authentifizierung für die Anmeldung beim Datenbank-Server verwenden, verbindet sich der Alfabet-Server zuerst mit dem Datenbank-Server unter Verwendung des Anwernamens und Passworts, die mit der Webdienst-Anforderung übermittelt wurden, um eine Anfrage von den Alfabet -Webdiensten zu verarbeiten. Wenn die Anmeldung jedoch fehlschlägt, wird die Windows-Anmeldung von Alfabet-Server als Auswechslösung verwendet und die Verbindung wird hergestellt, obwohl keine gültigen Parameter zur Clientauthentifizierung von der Webdienst-Anforderung bereitgestellt wurden.



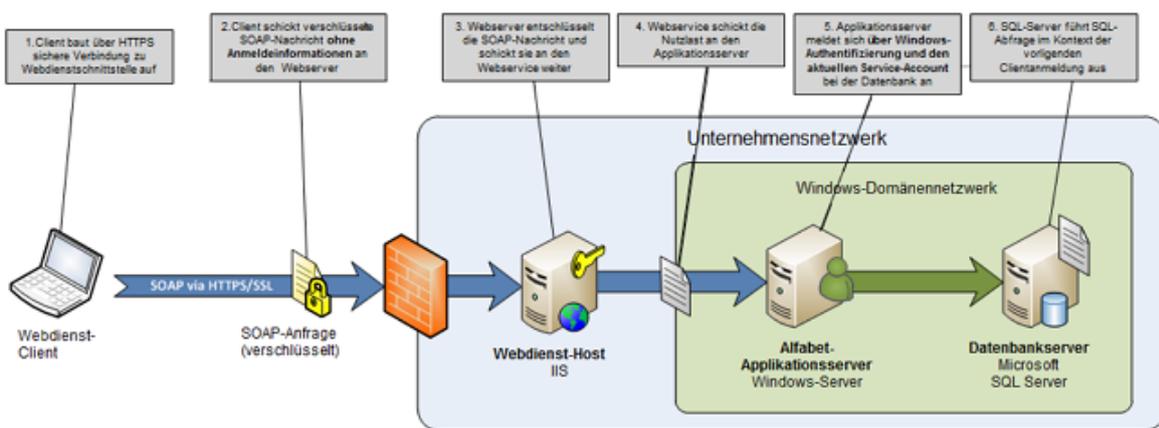
## Integrierte Windows-Authentifizierung

In diesem Szenario enthält der Webdienst-Client keine Authentifizierungsinformationen im Body der Anfrage, die an den Web-Server übermittelt wird. Stattdessen wird die Authentifizierung vom Applikationsserver ausgeführt, der die angeforderte Operation bereitstellt. Auf diese Weise werden keine sensiblen Datenbank-Authentifizierungsinformationen auf Webdienst-Clientseite des Systems offengelegt. Dieser Ansatz erlaubt jedoch keine eindeutige Definition der Berechtigungen, die verschiedenen Webdienst-Clienttypen bereitgestellt werden. Stattdessen werden alle Webdienst-Clientaufrufe im Kontext des Windows-Kontos ausgeführt, das zur Ausführung des Alfabet-Server -Service verwendet wird.

In der folgenden Liste wird die Verarbeitung von Webdienstanfragen mittels Windows-Authentifizierung beschrieben:

- 1) Die Implementierung des Webdienst-Clients umfasst eine SOAP-Nachricht, die alle für die angefragte Operation erforderlichen Parameter enthält. Die im Body der Nachricht enthaltenen Informationen beinhalten ausschließlich Informationen, die zur Ausführung der Operation erforderlich sind, z. B. Anfrageparameter. Für die Authentifizierung relevante Parameter der Operation können leer gelassen werden.

- 2) Über das Serverzertifikat, das vom Webdienst-Host bereitgestellt wird, verschlüsselt der Webdienst-Client die SOAP-Nachricht und übermittelt sie im Body der HTTP-Nachricht zum Webdienst-Host.
- 3) Der Webdienst-Host entschlüsselt den Body der Nachricht und leitet die SOAP-Nachricht zur Verarbeitung an die Microsoft® Internetinformationsdienste® im Webdienst-Anforderungshandler weiter.
- 4) Bei der Implementierung des Webdienstes wird die SOAP-Nachricht entpackt, und es wird ein Aufruf zur entsprechenden Schnittstelle des Applikationsserver vorbereitet, die die angeforderte Operation bereitstellt. Daraufhin werden die Parameter der angeforderten Operation an den Alfabet-Server übermittelt.
- 5) Nach Erhalt des Operationsaufrufs versucht der Alfabet-Server, sich unter Verwendung der in Windows integrierten Authentifizierung mit dem Datenbank-Server zu verbinden. Das für die Authentifizierung bei der Datenbank verwendete Konto ist das Domänenkonto, unter dem der Alfabet-Server -Service ausgeführt wird. Wenn der Verbindungsversuch erfolgreich war, werden die Statements, die zur Ausführung der angeforderten Operation erforderlich sind, an die Datenbank übermittelt. Die Datenbank führt diese Statements unter Verwendung des Alfabet-Server -Servicekontos und mit den zugehörigen Berechtigungen aus. Wenn kein Fehler auftritt, werden die Daten zurück an den aufrufenden Webdienst-Host gesendet und schließlich via HTTPS an den anfordernden Webdienst-Client zurückgegeben.



## Kapitel 3: Für die Alfabet -Webdienste implementierte Methoden

**AlfaQueryWebService** verwendet zum Lesen der Alfabet -Daten folgende Methoden:

<a href="#">GetInstancesByQuery</a>	gibt unter Verwendung einer Alfabet -Abfrage Objekte mit vollständigen Daten zurück.
<a href="#">GetInstancesByGuid</a>	gibt unter Verwendung einer GUID Objekte mit vollständigen Daten zurück.
<a href="#">GetClassInstances</a>	gibt alle Objekte mit den vollständigen Daten einer Objektklasse zurück, die durch den Objektklassennamen definiert sind.
<a href="#">GetDataSetByQuery</a>	gibt unter Verwendung einer Alfabet -Abfrage Objekte (als Referenzen) zurück.
<a href="#">GetDataSetByNativeQuery</a>	gibt unter Verwendung einer Native-SQL-Abfrage Objekte (als Referenzen) zurück.

**AlfaUpdateDataWebService** verwendet zur Manipulation der Alfabet -Daten die folgende Methode:

<a href="#">UpdateData</a>	fügt Objekte und Beziehungen ein und aktualisiert und löscht diese
----------------------------	--

Alle Methoden haben fünf gemeinsame Argumente:

aServer-Name	Verbindungsstring (Protokoll, Server, Port und Name des Remote-Servers), der zur Verbindung der Webdienste mit dem relevanten Alfabet-Server verwendet wird, z. B. "tcp://localhost:8087/ Alfabet "
	Syntax: tcp://(Computer):(Port)/(Server)
	(1) (Computer): DNS-Name oder IP-Adresse des Alfabet-Server -Hosts
	(2) (Port): die Eigenschaft "Port", die in der Konfiguration des relevanten Remote-Alias definiert ist
	(3) (Server): das Attribut "Server", das in der Konfiguration des relevanten Remote-Alias definiert ist
aUserName	Der Anwendername für die Verbindung zum Datenbankserver.
aPassword	Das Passwort für die Verbindung zum Datenbankserver.
aProfile-Name	Dieses Argument ist obsolet, aber aus Gründen der Abwärtskompatibilität trotzdem verfügbar. Ein für dieses Argument definierter String wird ignoriert.

`bEnsureSecurity` Dieses Argument muss mit der Einstellung **Sicherheit gewährleisten** übereinstimmen, die im Server-Alias definiert ist. Sonst schlagen die Aufrufe zum Webdienst fehl. Setzen Sie es auf `true`, wenn **Sicherheit gewährleisten** in der Konfiguration des Server-Alias aktiviert ist. Setzen Sie es auf `false`, wenn **Sicherheit gewährleisten** in der Konfiguration des Server-Alias deaktiviert ist.

Für alle verfügbaren Methoden sind zusätzliche methodenspezifische Argumente erforderlich. Diese werden im Detail in den Abschnitten zu den verschiedenen Methoden definiert. Die ausführliche Beschreibung der allgemeinen Argumente wird in der Beschreibung der einzelnen Methoden nicht wiederholt.

## GetInstancesByQuery

```
GetInstancesByQuery (string aServerName, string aUserName, string aPassword,
string aQuery),
```

mit

`aQuery` muss eine gültige Alfabet -Abfrage sein.

Beispielsweise könnte die Alfabet -Abfrage wie folgt aussehen:

```
"FIND Application WHERE Name like 'A%'"
```



Eine detaillierte Beschreibung der Alfabet -Abfragesprache finden Sie im Kapitel *Definieren von Abfragen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*. Wenn Sie Abfragen in Native-SQL definieren möchten oder die Alfabet -Abfragesprache nicht die Abfragemechanismen bietet, die Sie implementieren möchten, wird empfohlen, stattdessen die Methode [GetDataSetByNativeQuery](#) oder die Webdienste, die von der zugrunde liegenden Datenbankplattform angeboten werden, anstelle der Webdienste von Alfabet zu verwenden.



Wenn mehr als eine Landeseinstellung für Alfabet definiert ist, muss die Abfrage Informationen zur Landeseinstellung enthalten, die für die Rückgabe der Ergebnisse verwendet werden soll. Zur Verwendung nur in Webdiensten: Die Alfabet -Abfragesprache bietet einen Parameter für die Rückgabe von Ergebnissen in einer definierten Sprache an. `LOCALE_ID <Sprachcode>` muss der Alfabet -Abfrage vor dem `FIND`-Statement hinzugefügt werden. Um beispielsweise Applikationsnamen in deutscher Sprache zurückzugeben, würden Sie Folgendes definieren:

```
ALFABET_QUERY_500 LOCALE_ID 1031 FIND Application WHERE Name like
'T%' QUERY_XML &lt;QueryDef&gt; &lt;ShowProperty Type='Property'
ClassName='Application' Name='Name' /&gt; &lt;/QueryDef&gt;
```

**Rückgabewert:** Die Methode gibt ein XML-Dokument zurück, das die Instanzen enthält, die von einer Alfabet -Abfrage gefunden wurden. Das Dokument bietet Informationen zu allen Eigenschaftswerten und Beziehungen der Instanz, die in der Alfabet-Datenbank gespeichert wird.

Das Root-Element **AlfaInstances** verfügt über die folgenden untergeordneten Elemente:

- Ein **AlfaInstance**-Element für jede Instanz, die von der Alfabet -Abfrage gefunden wird. Das **AlfaInstance**-Element verfügt für jede definierte Eigenschaft der Instanz über ein untergeordnetes **AlfaPropertyValue**-Element.

- Ein **AlfaRelations**-Element mit einem untergeordneten **AlfaRelation**-Element für jede Beziehung in der RELATIONS-Tabelle der Alfabet-Datenbank, die eine Beziehung von oder zu einer der von der Abfrage gefundenen Instanzen definiert.

Die XML-Elemente enthalten Informationen zu den Objektinstanzen, die von der Alfabet -Abfrage in den folgenden Attributen gefunden wurden:

Attribut	Beschreibung
<b>AlfaInstance</b>	
Guid	Die GUID der Objektklasseninstanz.
ClassGuid	Die GUID der Objektklasse.
ClassName	Der Name der Objektklasse.
Mandates	Die Mandantenzuordnung der Objektklasse.  Weitere Informationen zur Mandantenzuordnung für Objekte und zur Speicherung der Informationen finden Sie im Referenzhandbuch <i>Das Alfabet-Metamodell</i> .
Id	Die Id wird aus der REFSTR-Eigenschaft der Objektklasseninstanz extrahiert. Der REFSTR ist definiert als:  <Id>-<ClassId>-<CompanyId>
ClassId	Die ClassId wird aus der REFSTR-Eigenschaft der Objektklasseninstanz extrahiert. Der REFSTR ist definiert als:  <Id>-<ClassId>-<CompanyId>
CompanyId	Die CompanyId wird aus der REFSTR-Eigenschaft der Objektklasseninstanz extrahiert. Ein REFSTR ist definiert als:  <Id>-<ClassId>-<CompanyId>
<b>AlfaPropertyValue</b>	
Name	Der Name der Objektklasseneigenschaft.
Type	Der Datentyp der Objektklasseneigenschaft.

Attribut	Beschreibung
Datum	Der Wert der Objektklasseninstanz für die Objektklasseneigenschaft.
<b>AlfaRelation</b>	
From	Die GUID der Objektklasseninstanz, die die Beziehung zu einer ihrer Eigenschaften definiert.
To	Die GUID der Objektklasseninstanz, zu der die Beziehung erstellt wird.
Property	Der Name der Objektklasseneigenschaft, die die Beziehung herstellt.



```
<?xml version="1.0" encoding="utf-8" ?>
  <AlfaInstances>
    <AlfaInstance CompanyId="0"
      ClassName="ApplicationGroup"
      ClassGuid="F3790442ADFA41ACB153779CED6B24A3"
      Guid="10522BA133924E009D00B6C68AEF9001"
      ClassId="183"
      Id="5"
      Mandates="1">
      <AlfaPropertyValue Name="Name" Type="String"
        Datum="Group1" />
    </AlfaInstance>
    <AlfaRelations> <AlfaRelation
      From="F3790442ADFA41ACB153779CED6B24A3"
      To="841EBFD666654D21A02F6AF686416B15" Property="ConsistsOf"
    />
  </AlfaRelations>
</AlfaInstances>
```

## GetInstancesByGuid

```
GetInstancesByGuid (string aServerName, string aUserName, string aPassword,
string aClassName, string guids)
```

mit

**aClassName** muss ein gültiger Alfabet -Klassenname sein.

**guids** Kommagetrennter String der GUIDs.

**Rückgabewert** : Die Methode gibt ein XML-Dokument zurück, das die von den GUIDs gefundenen Instanzen enthält. Eine Beschreibung der XML-Struktur und der Bedeutung der Attribute des XML-Elements in dem Dokument finden Sie im Abschnitt [GetInstancesByQuery](#).



```
<?xml version="1.0" encoding="utf-8" ?>
  <AlfaInstances>
    <AlfaInstance
      CompanyId="0"
      ClassName="ApplicationGroup"
      ClassGuid="F3790442ADFA41ACB153779CED6B24A3"
      Guid="10522BA133924E009D00B6C68AEF9001"
      ClassId="183"
      Id="5">
      <AlfaPropertyValue Name="Name" Type="String"
        Datum="Group1" />
    </AlfaInstance>
    <AlfaRelations>
      <AlfaRelation
        From="10522BA133924E009D00B6C68AEF9001"
        To="841EBFD666654D21A02F6AF686416B15"
        Property="ConsistsOf" />
    </AlfaRelations>
  </AlfaInstances>
```

## GetClassInstances

```
GetClassInstances (string aServerName, string aUserName, string aPassword,
string aClassName)
```

mit

**aClassName** muss ein gültiger Alfabet -Klassenname sein.

**Rückgabewert** : Die Methode gibt ein XML-Dokument zurück, das alle in der Objektklasse gefundenen Instanzen enthält, mit dem Namen, der im Argument **aClassName** definiert ist. Eine Beschreibung der XML-Struktur und der Bedeutung der Attribute des XML-Elements in dem Dokument finden Sie im Abschnitt [GetInstancesByQuery](#).



```
<?xml version="1.0" encoding="utf-8" ?>
  <AlfaInstances>
    <AlfaInstanceCompanyId="0"
      ClassName="ApplicationGroup"
      ClassGuid="F3790442ADFA41ACB153779CED6B24A3"
      Guid="10522BA133924E009D00B6C68AEF9001"
      ClassId="183"
      Id="5">
      <AlfaPropertyValue Name="Name" Type="String"
        Datum="Group1" />
    </AlfaInstance>
    <AlfaRelations>
      <AlfaRelation
        From="10522BA133924E009D00B6C68AEF9001"
        To="841EBFD666654D21A02F6AF686416B15"
        Property="ConsistsOf" />
    </AlfaRelations>
  </AlfaInstances>
```

## GetDataSetByQuery

```
GetDataSetByQuery (string aServerName, string aUserName, string aPassword,
string aQuery)
```

mit

**aQuery** muss eine gültige Alfabet -Abfrage sein.

Beispielsweise könnte eine Alfabet -Abfrage wie folgt aussehen:

```
"FIND Application WHERE Name like 'T%' QUERY_XML &lt;QueryDef&gt;
&lt;ShowProperty Type='Property' ClassName='Application' Name='Name' /&gt;
&lt;/QueryDef&gt;"
```



Eine detaillierte Beschreibung der Alfabet -Abfragesprache finden Sie im Kapitel *Definieren von Abfragen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*. Wenn Sie Abfragen in Native-SQL definieren möchten, können Sie stattdessen die Methode **GetDataSetByNativeQuery** verwenden. Informationen hierzu finden Sie unter [GetDataSetByNativeQuery](#).



Wenn mehr als eine Landeseinstellung für Alfabet definiert ist, muss die Abfrage Informationen zur Landeseinstellung enthalten, die für die Rückgabe der Ergebnisse verwendet werden soll. Zur Verwendung nur in Webdiensten: Die Alfabet -Abfragesprache bietet einen Parameter für die Rückgabe von Ergebnissen in einer definierten Sprache an. `LOCALE_ID <Sprachcode>` muss der

Alfabet -Abfrage vor dem `FIND`-Statement hinzugefügt werden. Um beispielsweise Applikationsnamen in deutscher Sprache zurückzugeben, würden Sie Folgendes definieren:

```
ALFABET_QUERY_500 LOCALE_ID 1031 FIND Application WHERE Name like
'T%' QUERY_XML <QueryDef> <ShowProperty Type='Property'
ClassName='Application' Name='Name' /> </QueryDef>
```

**Rückgabewert:** Die Methode gibt ein XML-Dokument zurück, das den Datensatz enthält, der von der Alfabet -Abfrage gefunden wurde.

Unter dem Root-Element **AlfaDataSet** sind folgende Elemente im Dokument vorhanden:

- Ein **AlfaDataSetCol**-Element für jede Eigenschaft, die in den SHOW-Eigenschaften der Alfabet -Abfrage enthalten sind. Das Element **AlfaDataSetCol** verfügt über zwei untergeordnete Elemente:
  - **Name:** Der Name der Eigenschaft, die in der Spalte angezeigt wird.
  - **Type:** Der Datentyp der Eigenschaft, die in der Spalte angezeigt wird.
- Ein **AlfaDataSetRow**-Element für jede Objektklasseninstanz, die von der Alfabet -Abfrage gefunden wurde. Das Element **AlfaDataSetRow** verfügt über mehrere untergeordnete Elemente:
  - **<Spaltenname>** : Für jede Spalte, die mit einem **AlfaDataSetCol**-Element definiert ist, wird ein Element hinzugefügt. Der Name des Elements ist mit dem Namen der Eigenschaft identisch. Der Inhalt des Elements ist der Wert der Instanz für die Eigenschaft.



```
<?xml version="1.0" encoding="utf-8" ?>
  <AlfaDataSet>
    <AlfaDataSetCol>
      <Name>Reference</Name>
      <Type>String</Type>
    </AlfaDataSetCol>
    <AlfaDataSetCol>
      <Name>Name</Name>
      <Type>String</Type>
    </AlfaDataSetCol>
    <AlfaDataSetRow>
      <Reference>183-5-0</Reference>
      <Name>Test1</Name>
    </AlfaDataSetRow>
    <AlfaDataSetRow>
      <Reference>183-6-0</Reference>
      <Name>Test2</Name>
    </AlfaDataSetRow>
    <AlfaDataSetRow>
      <Reference>183-12-0</Reference>
```

```

        <Name>Test4</Name>
    </AlfaDataSetRow>
</AlfaDataSet>

```

## GetDataSetByNativeQuery

```

GetDataSetByNativeQuery (string aServerName, string aUserName, string
aPassword, string aQuery)

```

mit

**aQuery** muss eine gültige Native-SQL-Abfrage sein.

Beispielsweise könnte eine Alfabet -Abfrage wie folgt aussehen:

```
"SELECT app.REFSTR, app.NAME FROM APPLICATION app WHERE app.NAME like 'A%'"
```



- Die allgemeinen Regeln für die Verwendung von Native-SQL im Kontext von Alfabet sind im Kapitel *Definieren von Abfragen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand* beschrieben.
- Das `SELECT`-Statement wird in Alfabet -Konfigurationen anders bewertet als der allgemeine Bewertungsmodus. Es ist nicht erforderlich, den `REFSTR` im `SELECT`-Statement als erstes Argument anzugeben. Falls definiert, wird der `REFSTR` in den Ergebnisdatensatz geschrieben.



Bei dieser Methode wird zur korrekten Ausführung ein `SELECT`-Statement erwartet. Beachten Sie jedoch, dass im Webdienst nicht geprüft werden kann, ob die Abfrage mit einem `SELECT`-Statement anstatt z. B. mit einem `UPDATE`-Statement definiert wurde. Um sicherzustellen, dass dies nicht zu Sicherheitsproblemen führt, sind die Alfabet -Webdienste durch die im Abschnitt [Authentifizierung der Alfabet -Webdienste](#) beschriebenen Authentifizierungsmechanismen geschützt.

**Rückgabewert:** Die Methode gibt ein XML-Dokument zurück, das den Datensatz enthält, der von der Native-SQL-Abfrage gefunden wurde.

Unter dem Root-Element **AlfaDataSet** sind folgende Elemente im Dokument vorhanden:

- Ein **AlfaDataSetCol**-Element für jedes Argument, das im `SELECT`-Statement der Native-SQL-Abfrage enthalten ist. Das Element **AlfaDataSetCol** verfügt über zwei untergeordnete Elemente:
  - **Name:** Der Name der Eigenschaft, die in der Spalte angezeigt wird.
  - **Type:** Der Datentyp der Eigenschaft, die in der Spalte angezeigt wird.
- Ein **AlfaDataSetRow**-Element für jede Objektklasseninstanz, die von der Alfabet -Abfrage gefunden wurde. Das Element **AlfaDataSetRow** verfügt über mehrere untergeordnete Elemente:
  - **<Spaltenname>:** Für jede Spalte, die mit einem **AlfaDataSetCol**-Element definiert ist, wird ein Element hinzugefügt. Der Name des Elements ist mit dem Namen der Eigenschaft identisch. Der Inhalt des Elements ist der Wert der Instanz für die Eigenschaft.



```
<?xml version="1.0" encoding="utf-8" ?>
  <AlfaDataSet>
    <AlfaDataSetCol>
      <Name>Reference</Name>
      <Type>String</Type>
    </AlfaDataSetCol>
    <AlfaDataSetCol>
      <Name>Name</Name>
      <Type>String</Type>
    </AlfaDataSetCol>
    <AlfaDataSetRow>
      <Reference>183-5-0</Reference>
      <Name>Test1</Name>
    </AlfaDataSetRow>
    <AlfaDataSetRow>
      <Reference>183-6-0</Reference>
      <Name>Test2</Name>
    </AlfaDataSetRow>
    <AlfaDataSetRow>
      <Reference>183-12-0</Reference>
      <Name>Test4</Name>
    </AlfaDataSetRow>
  </AlfaDataSet>
```

## UpdateData

UpdateData (string aServerName, string aUserName, string aPassword, string anXml)

mit

**anXml** muss ein gültiges XML-Dokument sein, das Aktualisierungsoperationen enthält. Die Details sind unten definiert.



Die Methode **UpdateData** führt keine Überprüfung der Alfabet-Datenbank durch. Der Alfabet-Server setzt automatisch die Eigenschaften `REFSTR`, `ID` und `GUID` neuer Objekte und identifiziert Objekte über die `GUID` und Objektklassen über deren Attribut "Name". Wenn Sie die Methode **UpdateData** verwenden, sind Sie selber dafür verantwortlich sicherzustellen, dass keine Probleme hinsichtlich Datenintegrität oder Zugriffsfähigkeit entstehen und das obligatorische Eigenschaften für Objekte während der Erstellung gesetzt werden. Es wird empfohlen, das

Referenzhandbuch *Das Alfabet-Metamodell* zu verwenden um zu verstehen, welche Eigenschaften für eine Objektklasse wie gesetzt sein müssen.

Die Daten in dem XML-Dokument, das mit dem Argument **anXML** definiert wurde, müssen XML-konform sein. Im XML-Inhalt sind keine Sonderzeichen zulässig.

Das mit dem Argument **anXML** definierte XML-Dokument muss über ein **AlfaDataUpdate**-Root-Element mit einem oder mehreren der folgenden untergeordneten Elemente verfügen, die unten ausführlich beschrieben werden:

Element	Durch Element ausgelöste Aktion
<a href="#">UpdateInstance</a>	Ändert die Eigenschaftswerte vorhandener Objekte in der Alfabet-Datenbank.
<a href="#">CreateInstance</a>	Erzeugt ein neues Objekt in der Alfabet-Datenbank.
<a href="#">DeleteInstance</a>	Löscht ein Objekt aus der Alfabet-Datenbank.
<a href="#">CreateRelation</a>	Erstellt eine Beziehung zwischen zwei vorhandenen Objekten in der Alfabet-Datenbank.
<a href="#">DeleteRelation</a>	Löscht eine Beziehung aus der Alfabet-Datenbank.

Ein **AlfaDataUpdate**-Element kann mehrere verschiedene untergeordnete Elemente enthalten. Wenn Sie ein XML-Dokument definieren, das Definitionen für die Manipulation sowohl der Instanzdaten als auch der Beziehungsdaten enthält, werden die Elemente in zwei Schritten verarbeitet:

- Alle untergeordneten Elemente, die die Erzeugung, Aktualisierung und Löschung von Instanzen auslösen, werden zuerst verarbeitet, unabhängig von ihrer Position vor oder nach Elementen, um Beziehungsdaten zu ändern.
- Alle untergeordneten Elemente, die die Erzeugung oder Löschung von Beziehungen auslösen, werden in einem zweiten Schritt verarbeitet.

Die beiden Aktualisierungsschritte werden als separate Aktualisierungsverfahren verarbeitet. Wenn die Änderung der Beziehung also fehlschlägt, wird die Verarbeitung der Beziehungen zurückgenommen, während die im ersten Schritt während der Instanzverarbeitung erzeugten Instanzen in der Alfabet-Datenbank verbleiben.



Das folgende Beispiel zeigt eine typische Aktualisierung eines XML-Dokuments:

```
<AlfaDataUpdate>
  <UpdateInstance
    ClassName="Application"
    Guid="841EBFD298654D21A02F6AF686416B15">
```

```

        <AlfaPropertyValue Name="Name" Type="String"
        Datum="Test1"/>

        <AlfaPropertyValue Name="Status" Type="String"
        Datum="Active"/>

    </UpdateInstance>

    <CreateInstance

        ClassName="ApplicationGroup">

        <AlfaPropertyValue Name="Name" Type="String"
        Datum="Test4"/>

    </CreateInstance>

    <DeleteInstance

        ClassName="ApplicationGroup"

        Guid="841EBFD298654D21A02F6AF686416B15"/>

    <CreateRelation

        From="C6F3EEBE4595414DAF359EF6FA79C192"

        To="BBD25FA714D84ACD896A95FCCFCA1410"

        Property="ConsistsOf" />

    <DeleteRelation

        From="C6F3EEBE4595414DAF359EF6FA79C192"

        To="BBD25FA714D84ACD896A95FCCFCA1410"

        Property="ConsistsOf" />

</AlfaDataUpdate>

```

**Rückgabewert:** Die Methode gibt ein XML-Dokument zurück, das über die Änderungen informiert, die in der Alfabet-Datenbank ausgeführt wurden.



```

<?xml version="1.0"?>
  <AlfaDataUpdateResult
    InstancesUpdated="1"
    InstancesAdded="20"
    InstancesDeleted="3"
    RelationsAdded="3"
    RelationsDeleted="1340"/>

```

## UpdateInstance

Das Element **UpdateInstance** definiert eines oder mehrere Objekte für die Eigenschaftswerte in der Alfabet-Datenbank aktualisiert werden sollen. Das Attribut "Name" der Objektklasse wird verwendet, um die Objektklasse zu identifizieren. Ein einzelnes Objekt wird über seine GUID identifiziert, während mehrere Objekte über eine Alfabet -Abfrage identifiziert werden.

Innerhalb des **UpdateInstance**-Elements werden die Werte für die Eigenschaften, die aktualisiert werden sollen, in untergeordneten **AlfaPropertyValue**-Elementen definiert.

Bitte beachten Sie hinsichtlich der Aktualisierung von Instanzdaten Folgendes:

- Die GUID-, REFSTR- und ID-Werte für Objekte in der Alfabet-Datenbank werden dem Objekt automatisch zugewiesen und dürfen nicht über ein **AlfaPropertyValue**-Element geändert werden.
- Datumswerte müssen im Format JJJJ-MM-TT angegeben werden.
- Es wird empfohlen, im Referenzhandbuch *Das Alfabet-Metamodell* nachzuschlagen, um die Anforderungen für die Eigenschaftseinstellungen sowie die Einschränkungen, die möglicherweise für die Festlegung des korrekten Werts gelten, zu verstehen. Zum Beispiel kann eine Eigenschaft auf einer Aufzählung basieren, was dazu führt, dass nur Werte, die in der Aufzählung definiert sind, gültig sind. Oder es ist ein Klassenschlüssel vorhanden, der erfordert, dass der Wert innerhalb der Datenbanktabelle eindeutig ist.
- Wenn das Element **UpdateInstance** über eine Alfabet -Abfrage mehrere Instanzen finde, dann werden alle diese Instanzen mit den Eigenschaftswerten aktualisiert, die in den untergeordneten Elementen **AlfaPropertyValue** definiert sind. Stellen Sie sicher, dass die Eigenschaftsdefinitionen im Element **UpdateInstance** keine Eigenschaften enthalten, die Bestandteil der Konfiguration eines eindeutigen Schlüssels sind..
- Eigenschaften des Typs `ReferenceArray`, dessen Attribut **ReferenceSupport** auf `true` gesetzt ist, definieren Beziehungen, die in der `RELATIONS`-Tabelle der Alfabet-Datenbank gespeichert sind. Diese Eigenschaften können nicht über das **AlfaPropertyValue**-Element des **UpdateInstance**-Elements definiert werden. Stattdessen müssen **CreateRelation**- und **DeleteRelation**-Elemente definiert werden, um die Beziehungen zu aktualisieren. Informationen hierzu finden Sie in den Abschnitten [CreateRelation](#) und [DeleteRelation](#).

#### UpdateInstance-Element, das Objekte über die GUID identifiziert

Die folgenden Attribute müssen für das **UpdateInstance**-Element und die untergeordneten **AlfaPropertyValue**-Elemente definiert werden:

Attribut	Beschreibung
<b>UpdateInstance</b>	
ClassName	Der Wert des Attributs <b>Name</b> der Objektklasse, für die die Instanz erzeugt werden soll.
Guid	Die GUID der Objektklasseninstanz, die aktualisiert werden soll.
<b>AlfaPropertyValue</b>	

Attribut	Beschreibung
Name	Der Wert des Attributs <b>Name</b> der Objektklasseneigenschaft, für die ein Wert mit diesem <b>AlfaPropertyValue</b> -Element definiert ist.
Type	Der Wert des Attributs <b>Typ</b> der Objektklasseneigenschaft, für die ein Wert mit diesem <b>AlfaPropertyValue</b> -Element definiert ist.
Datum	Der Wert, der für die Eigenschaft festgelegt werden soll.



```

<UpdateInstance
  ClassName="Application"
  Guid="841EBFD298654D21A02F6AF686416B15">
  <AlfaPropertyValue Name="Name" Type="String" Datum="Test1"/>
  <AlfaPropertyValue Name="Status" Type="String"
    Datum="Active"/>
</UpdateInstance>
    
```

Der Alfabet -Applikationsserver versucht, eine vorhandene Instanz mit einer angegebenen GUID zu finden. Wenn die Instanz vorhanden ist, werden die entsprechenden Werte aktualisiert und gespeichert. Andernfalls wird eine neue Instanz mit der angegebenen GUID erzeugt, und die Werte werden festgelegt und gespeichert.

#### UpdateInstance-Element, das Objekte über eine Alfabet -Abfrage identifiziert

Die folgenden Attribute müssen für das **UpdateInstance**-Element und die untergeordneten **AlfaPropertyValue**-Elemente definiert werden:

Attribut	Beschreibung
<b>UpdateInstance</b>	
Query	Eine Alfabet -Abfrage, die die zu aktualisierenden Objekte zurückgibt.
<b>AlfaPropertyValue</b>	

Attribut	Beschreibung
Name	Der Wert des Attributs "Name" der Objektklasseneigenschaft, für die ein Wert mit diesem <b>AlfaPropertyValue</b> -Element definiert ist.
Type	Der Wert des Attributs "Typ" der Objektklasseneigenschaft, für die ein Wert mit diesem <b>AlfaPropertyValue</b> -Element definiert ist.
Datum	Der Wert, der für die Eigenschaft festgelegt werden soll.



```
<UpdateInstance
  Query="FIND Application
  WHERE (AND Name like 'T%' Status = 'Retired')>
  <AlfaPropertyValue Name="Status" Type="String"
  Datum="Active"/>
</UpdateInstance>
```

Für alle von der Alfabet -Abfrage gefundenen Instanzen werden die entsprechenden Werte aktualisiert und gespeichert.

Beachten Sie bezüglich der Definition der Alfabet -Abfrage Folgendes:

- Die Definition der `SHOW`-Eigenschaften ist für die Alfabet -Abfrage nicht erforderlich.
- Die Objekte der im `FIND`-Statement definierten Klasse werden aktualisiert. Es können `JOINS` definiert werden, um `WHERE`-Bedingungen basierend auf den Referenzen zu anderen Objektklassen anzugeben. Objekte von Klassen, die den Ergebnissen über einen `JOIN` hinzugefügt wurden, werden jedoch nicht aktualisiert.
- Eine detaillierte Beschreibung der Alfabet -Abfragesprache finden Sie im Kapitel *Definieren von Abfragen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*. Wenn Sie Abfragen in Native-SQL definieren möchten oder die Alfabet -Abfragesprache nicht die Abfragemechanismen bietet, die Sie implementieren möchten, wird empfohlen, anstelle der Webdienste von Alfabet die Webdienste zu verwenden, die von der zugrunde liegenden Datenbankplattform angeboten werden.

## CreateInstance

Das **CreateInstance**-Element definiert die Objektklasse, für die ein neues Element zur Alfabet-Datenbank hinzugefügt werden soll. Das Attribut **Name** der Objektklasse wird verwendet, um die Objektklasse zu identifizieren.

Innerhalb des **CreateInstance**-Elements werden die Werte für die Eigenschaften des neuen Objekts im untergeordneten **AlfaPropertyValue**-Element definiert.

Die folgenden Attribute müssen für die Elemente **CreateInstance** und **AlfaPropertyValue** definiert werden:

Attribut	Beschreibung
<b>CreateInstance</b>	
ClassName	Der Wert des Attributs <b>Name</b> der Objektklasse, für die die Instanz erzeugt werden soll.
<b>AlfaPropertyValue</b>	
Name	Der Wert des Attributs <b>Name</b> der Objektklasseneigenschaft, für die ein Wert mit diesem <b>AlfaPropertyValue</b> -Element definiert ist.
Type	Der Wert des Attributs <b>Typ</b> der Objektklasseneigenschaft, für die ein Wert mit diesem <b>AlfaPropertyValue</b> -Element definiert ist.
Datum	Der Wert, der für die Eigenschaft festgelegt werden soll.



```
<CreateInstance
  ClassName="Application">
  <AlfaPropertyValue Name="Name" Type="String" Datum="Test1"/>
  <AlfaPropertyValue Name="Version" Type="String" Datum="2.1"/>
  <AlfaPropertyValue Name="StartDate" Type="Date" Datum="2012-01-28"/>
  <AlfaPropertyValue Name="EndDate" Type="Date" Datum="2014-01-28"/>
</CreateInstance>
```

Beachten Sie bezüglich der Erzeugung neuer Instanzen Folgendes:

- Die GUID-, REFSTR- und ID-Werte für die neuen Objekte werden dem Objekt automatisch zugewiesen und dürfen nicht in einem **AlfaPropertyValue**-Element angegeben werden.
- Datumswerte müssen im Format JJJJ-MM-TT angegeben werden.
- Einige der Eigenschaften für eine Objektklasse sind obligatorisch. Diese Eigenschaften müssen mit einem **AlfaPropertyValue**-Element festgelegt werden.
- Es wird empfohlen, im Referenzhandbuch *Das Alfabet-Metamodell* nachzuschlagen, um die Anforderungen für die Eigenschaftseinstellungen sowie die Einschränkungen, die möglicherweise

für die Festlegung des korrekten Werts gelten, zu verstehen. Zum Beispiel kann eine Eigenschaft auf einer Aufzählung basieren, was dazu führt, dass nur Werte, die in der Aufzählung definiert sind, gültig sind. Oder es ist ein Klassenschlüssel vorhanden, der erfordert, dass der Wert innerhalb der Datenbanktabelle eindeutig ist.

- Eigenschaften des Typs `ReferenceArray`, dessen Attribut **ReferenceSupport** auf `true` gesetzt ist, definieren Beziehungen, die in der `RELATIONS`-Tabelle der Alfabet-Datenbank gespeichert werden. Diese Eigenschaften können nicht über das **AlfaPropertyValue**-Element des **CreateInstance**-Elements definiert werden. Stattdessen müssen **CreateRelation**-Elemente definiert werden, um die Beziehungen zu erzeugen. Weitere Informationen hierzu finden Sie unter [CreateRelation](#).

## DeleteInstance

Das Element **DeleteInstance** definiert eines oder mehrere Objekte, die aus der Alfabet-Datenbank gelöscht werden sollen. Das Attribut **Name** der Objektklasse wird verwendet, um die Objektklasse zu identifizieren. Ein einzelnes Objekt wird über seine `GUID` identifiziert, während mehrere Objekte über eine Alfabet -Abfrage identifiziert werden.

Beachten Sie beim Löschen von Instanzen Folgendes:

- Wenn ein Objekt gelöscht wird, wendet der Alfabet-Server automatisch eine kaskadierende Löschung der Instanzen an, die über Eigenschaften miteinander verbunden sind, die im Attribut **Integritäts-Info** der Objektklasse des gelöschten Objekts definiert sind.
- Wenn die Instanz mit anderen Objekten in den Alfabet-Datenbank en über eine Beziehung verbunden ist, die in der Tabelle `RELATIONS` der Alfabet-Datenbank gespeichert ist, und nicht Teil des Attributs **Integritäts-Info** ist, müssen die Beziehungen ebenfalls gelöscht werden. Beziehungen werden über **DeleteRelation**-Element gelöscht. Weitere Informationen hierzu finden Sie unter [DeleteRelation](#).

### DeleteInstance-Element, das Objekte über die GUID identifiziert

Für das Element **DeleteInstance** müssen die folgenden Attribute definiert sein:

Attribut	Beschreibung
<code>ClassName</code>	Der Wert des Attributs <b>Name</b> der Objektklasse, für die die Instanz erzeugt werden soll.
<code>Guid</code>	Die <code>GUID</code> der Objektklasseninstanz, die gelöscht werden soll.



```
<DeleteInstance
  ClassName="Application"
```

```
Guid="841EBFD298654D21A02F6AF686416B15"/>
```

### DeleteInstance-Element, das Objekte über eine Alfabet -Abfrage identifiziert

Für das Element **DeleteInstance** müssen die folgenden Attribute definiert sein:

Attribut	Beschreibung
Query	Eine Alfabet -Abfrage, die die zu löschenden Objekte zurückgibt.



```
<DeleteInstance
  Query="FIND Application WHERE (AND Name like 'T%' Status =
  'Retired')"/>
```

Beachten Sie bezüglich der Definition der Alfabet -Abfrage Folgendes:

- Die Definition der `SHOW`-Eigenschaften ist für die Alfabet -Abfrage nicht erforderlich.
- Objekte der Klasse, die im `FIND`-Statement definiert sind oder gelöscht wurden. Es können `JOINS` definiert werden, um `WHERE`-Bedingungen basierend auf den Referenzen zu anderen Objektklassen anzugeben. Objekte von Klassen, die den Ergebnissen über einen `JOIN` hinzugefügt wurden, werden jedoch nicht gelöscht.
- Eine detaillierte Beschreibung der Alfabet -Abfragesprache finden Sie im Kapitel *Definieren von Abfragen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*. Wenn Sie Abfragen in Native-SQL definieren möchten oder die Alfabet -Abfragesprache nicht die Abfragemechanismen bietet, die Sie implementieren möchten, wird empfohlen, anstelle der Webdienste von Alfabet die Webdienste zu verwenden, die von der zugrunde liegenden Datenbankplattform angeboten werden.

### CreateRelation

Das Element **CreateRelation** löst die Erzeugung einer oder mehrerer Beziehungen in der Alfabet-Datenbank aus.



Das Element **CreateRelation** erzeugt Beziehungen, die in der Tabelle `RELATIONS` der Alfabet-Datenbank gespeichert werden. Dies sind Beziehungen, die auf einer Eigenschaft des Typs `ReferenceArray` basieren.

Das **CreateRelation**-Element kann Objekte entweder anhand ihrer `GUID`-Eigenschaft oder mittels einer Alfabet -Abfrage identifizieren. Die Identifizierung von Objekten über `GUIDS` erfordert ein separates **CreateRelation**-Element für jede zu erzeugende Beziehung. Wenn Objekte über eine Alfabet -Abfrage identifiziert werden, kann ein **CreateRelation**-Element die Erzeugung mehrerer Beziehungen auslösen.

### CreateRelation-Element, das Objekte über die Guid identifiziert

Das Element **CreateRelation** muss über folgende Attribute verfügen:

Attribut	Beschreibung
From	Die GUID der Objektklasseninstanz in der Alfabet-Datenbank, die die Beziehung über eine Eigenschaft erzeugt, die im Attribut <code>Property</code> definiert ist.
To	Die GUID der Objektklasseninstanz in der Alfabet-Datenbank, zu der die Beziehung erzeugt wird.
Property	Das Attribut <b>Name</b> der Objektklasseneigenschaft, die die Beziehung erzeugt.

Der Alfabet-Server sucht nach Instanzen mit den angegebenen GUIDS und erzeugt die Beziehung zwischen ihnen, wenn die Instanzen gefunden wurden.



```
<CreateRelation
  From="C6F3EEBE4595414DAF359EF6FA79C192"
  To="BBD25FA714D84ACD896A95FCCFCA1410"
  Property="ConsistsOf" />
```

### CreateRelation-Element, das Objekte über eine Alfabet-Abfrage identifiziert

Das Element **CreateRelation** muss über folgende Attribute verfügen:

Attribut	Beschreibung
QueryFrom	Eine Alfabet -Abfrage, die Objektklasseninstanzen in der Alfabet-Datenbank zurückgibt, die die Beziehung über die Eigenschaft erzeugen, die im Attribut <code>Property</code> definiert ist.
QueryTo	Eine Alfabet -Abfrage, die Objektklasseninstanzen in der Alfabet-Datenbank zurückgibt, zu denen die Beziehungen erzeugt werden.
Property	Das Attribut <b>Name</b> der Objektklasseneigenschaft, die die Beziehung erzeugt.



```
<CreateRelation
    QueryFrom="FIND Project WHERE Name like 'CMS_%'"
    QueryTo="FIND Project WHERE Name = 'CMSImplementation'"
    Property="Parent" />
```

Der Alfabet-Server sucht nach Instanzen über die definierten Abfragen und erzeugt ihre Beziehungen untereinander, wenn die Instanzen gefunden wurden.

Beachten Sie bezüglich der Definition der Alfabet -Abfrage Folgendes:

- Die Definition der `SHOW`-Eigenschaften ist für die Alfabet -Abfrage nicht erforderlich.
- Es werden die Objekte der Klasse berücksichtigt, die im `FIND`-Statement definiert sind. Es können `JOINS` definiert werden, um `WHERE`-Bedingungen basierend auf den Referenzen zu anderen Objektklassen anzugeben. Objekte von Klassen, die den Ergebnissen über einen `JOIN` hinzugefügt wurden, werden jedoch nicht berücksichtigt.
- Eine detaillierte Beschreibung der Alfabet -Abfragesprache finden Sie im Kapitel *Definieren von Abfragen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*. Wenn Sie Abfragen in Native-SQL definieren möchten oder die Alfabet -Abfragesprache nicht die Abfragemechanismen bietet, die Sie implementieren möchten, wird empfohlen, anstelle der Webdienste von Alfabet die Webdienste zu verwenden, die von der zugrunde liegenden Datenbankplattform angeboten werden.

## DeleteRelation

Das Element **DeleteRelation** löst die Löschung einer oder mehrerer Beziehungen aus der Alfabet-Datenbank aus.



Das Element **DeleteRelation** löscht die Beziehungen, die in der Tabelle `RELATIONS` der Alfabet-Datenbank gespeichert sind. Dies sind Beziehungen, die auf einer Eigenschaft des Typs `ReferenceArray` basieren.

Das **DeleteRelation**-Element kann Objekte entweder anhand ihrer `GUID`-Eigenschaft oder mittels einer Alfabet -Abfrage identifizieren. Die Identifizierung von Objekten über `GUIDs` erfordert ein separates **DeleteRelation**-Element für jede zu löschende Beziehung. Wenn Objekte über eine Alfabet -Abfrage identifiziert werden, kann ein **DeleteRelation**-Element die Löschung mehrerer Beziehungen auslösen.

### DeleteRelation-Element, das Objekte über die Guid identifiziert

Das Element **DeleteRelation** muss über folgende Attribute verfügen:

Attribut	Beschreibung
From	Die GUID der Objektklasseninstanz in der Alfabet-Datenbank, die die Beziehung über eine Eigenschaft erzeugt, die im Attribut <code>Property</code> definiert ist.

Attribut	Beschreibung
To	Die GUID der Objektklasseninstanz in der Alfabet-Datenbank, zu der die Beziehung erzeugt wird.
Property	Das Attribut <b>Name</b> der Objektklasseneigenschaft, die die Beziehung erzeugt.

Der Alfabet-Server sucht nach Instanzen mit den angegebenen GUIDs und entfernt die Beziehung zwischen ihnen, wenn die Instanzen gefunden wurden.



```
<DeleteRelation
  From="C6F3EEBE4595414DAF359EF6FA79C192"
  To="BBD25FA714D84ACD896A95FCCFCA1410"
  Property="ConsistsOf" />
```

**DeleteRelation-Element, das Objekte über eine Alfabet -Abfrage identifiziert**

Das Element **DeleteRelation** muss über folgende Attribute verfügen:

Attribut	Beschreibung
QueryFrom	Eine Alfabet -Abfrage, die Objektklasseninstanzen in der Alfabet-Datenbank zurückgibt, die die Beziehung über die Eigenschaft erzeugen, die im Attribut <code>Property</code> definiert ist.
QueryTo	Eine Alfabet -Abfrage, die Objektklasseninstanzen in der Alfabet-Datenbank zurückgibt, zu denen die Beziehungen erzeugt werden.
Property	Das Attribut <b>Name</b> der Objektklasseneigenschaft, die die Beziehung erzeugt.



```
<DeleteRelation
  QueryFrom="FIND Project WHERE Status = 'New'"
  QueryTo="FIND Project WHERE Status = 'Discarded'"
  Property="Parent" />
```

Der Alfabet-Server sucht nach Instanzen über die definierten Abfragen und entfernt ihre Beziehungen untereinander, wenn die Instanzen gefunden wurden.

Beachten Sie bezüglich der Definition der Alfabet -Abfrage Folgendes:

- Die Definition der `SHOW`-Eigenschaften ist für die Alfabet -Abfrage nicht erforderlich.
- Es werden die Objekte der Klasse berücksichtigt, die im `FIND`-Statement definiert sind. Es können `JOINS` definiert werden, um `WHERE`-Bedingungen basierend auf den Referenzen zu anderen Objektklassen anzugeben. Objekte von Klassen, die den Ergebnissen über einen `JOIN` hinzugefügt wurden, werden jedoch nicht berücksichtigt.
- Eine detaillierte Beschreibung der Alfabet -Abfragesprache finden Sie im Kapitel *Definieren von Abfragen* im Referenzhandbuch *Konfigurieren von Alfabet mit Alfabet Expand*. Wenn Sie Abfragen in Native-SQL definieren möchten oder die Alfabet -Abfragesprache nicht die Abfragemechanismen bietet, die Sie implementieren möchten, wird empfohlen, anstelle der Webdienste von Alfabet die Webdienste zu verwenden, die von der zugrunde liegenden Datenbankplattform angeboten werden.

## Index

Aktualisierung	
Objekt	22
Alfabet-Abfrage	
GetDataSetByQuery	17
GetInstancesByQuery	13
Spracheinstellungen	17
Anmeldung	
Datenbankserver	9
Authentifizierung	
Anwendernamen und Passwort	9
Client-Authentifizierung	9
Mechanismen	9
Beziehung	
Erzeugen	28
Löschen	30
Daten	
Methoden	12
ändern	12
Datenbank	
Anmelden	9
Erzeugen	
Objekt	25
GetClassInstances	
Rückgabewert	16
Syntax	16
GetDataSetByNativeQuery	
Rückgabewert	19
Syntax	19
GetDataSetByQuery	
Rückgabewert	17
Syntax	13, 17
GetInstancesByGuid	
Rückgabewert	15
Syntax	15
GetInstancesByQueryd	
Rückgabewert	13
GUID	15
Instanz	

---

Beziehung erzeugen	28
Beziehung löschen	30
Löschen	27
Klassenname	16
Lesen von Daten	
GetClassInstances	16
GetDataSetByNativeQuery method	19
GetDataSetByQuery method	17
GetInstancesByGuid	15
GetInstancesByQuery	13
Methoden	12
Methoden	
Daten lesen	12
Daten ändern	12
Gemeinsame Argumente	12
GetClassInstances	16
GetDataSetByNativeQuery	19
GetDataSetByQuery	17
GetInstancesByGuid	15
GetInstancesByQuery	13
Update Data	20
Native SQL-Abfrage	
GetDataSetByNativeQuery method	19
Ändern von Daten	
Beziehung erzeugen	28
Beziehung löschen	30
Methoden	12
Objekt löschen	27
Änderung von Daten	
Objekt erzeugen	25
Objekte aktualisieren	22
UpdateData-Methode	20
Objekt	
Aktualisieren	22
Beziehung erzeugen	28
Beziehung löschen	30
Erzeugen	25
Löschen	27
Rückgabewert	
GetClassInstances	16
GetDataSetByNativeQuery	19
GetDataSetByQuery	17
GetInstancesByGuid	15
GetInstancesByQuery	13
UpdateData	22
Sprache definieren	

---

Für Alfabet-Abfrage	17
UpdateData	
Rückgabewert	22
Syntax	20
Webdienste	
Daten lesen	5
Daten ändern	5