# Denapp Bankdata Service

Beschreibung und Anleitung zur Nutzung des Web-Service ‚Denapp Bankdata Service‘


**Stand:** 22.02.2010

**Version:** **1.06.01 (gültig ab 01.03.2010)**

**Herausgeber:** Denapp



**Redaktion:** Denapp UG (haftungsbeschränkt)
Klarastraße 27, 04229 Leipzig
Tel.: 0341/39284280 – Fax: 0341/39284289
E-Mail: info@denapp.com


**Anmerkung:**

# Inhaltsverzeichnis

## Änderungshistorie

Diese Darstellung beschreibt die Änderungen zur jeweiligen Vorgängerversion und soll eine kurze Information über die geänderten Teile des Dokumentes geben.

| **Version** | **1.06.01** | **Gültig ab** | **01.03.2010** | **Datum** | **22.02.2010** |
|---|---|---|---|---|---|
| **Seite** | **Punkt** | **Art** (neu, geändert, gelöscht) | **Kurzbeschreibung der Änderung** | | |
| alle | alle | geändert | redaktionelle Überarbeitung der Dokumentation | | |

| **Version** | **1.06** | **Gültig ab** | **01.11.2009** | **Datum** | **06.10.2009** |
|---|---|---|---|---|---|
| **Seite** | **Punkt** | **Art** (neu, geändert, gelöscht) | **Kurzbeschreibung der Änderung** | | |
| alle | alle | neu | Erstellung der Dokumentation | | |

## Definitionen und Abkürzungen

**APP – A**tom **P**ublishing **P**rototocol
Definiert Standards für das Erstellen und Bearbeiten von Web-Ressourcen.

**BIC – B**ank **I**dentifier **C**ode
Eine internationale Bankleitzahl.

**DBS –** Denapp Bankdata Service

**Entität**
Bezeichnet ein einzelnes Objekt, ähnlich einem Datensatz in einer Datenbanktabelle.

**Entitätenmenge**
Bezeichnet eine Ansammlung von Objekten, ähnlich einer Datenbanktabelle.

**HTTP** – **H**yper**T**ext **T**ransfer **P**rotocol
Beschreibt die Übertragung von Daten über ein Netzwerk.

**IBAN** – **I**nternational **B**ank **A**ccount **N**umber
Eine internationale Kontonummer.

**JSON – J**ava**S**cript **O**bject **N**otation
Definiert ein Format für den Datenaustausch zwischen Schnittstellen.

**REST** – **RE**presentational **S**tate **T**ransfer
Beschreibt eine Anwendungsarchitektur.

**SEPA** – **S**ingle **E**uro **P**ayments **A**rea
Ein Gebiet mit einheitlichem Zahlungsverkehr (Überweisung, Lastschrift), auf Basis der Euro-Währung.

**SOAP** – **S**imple **O**bject **A**ccess **P**rotocol (SOAP gilt seit Version 1.2 als eigenständiger Name)
Ein Protokoll für die Kommunikation verteilter Anwendungen.

**SSL** – **S**ecure **S**ockets **L**ayer
Bezeichnet die verschlüsselte Datenübertragung über ein Netzwerk.

**UTC** – **C**oordinated **U**niversal **T**ime
Die koordinierte Weltzeit.

# 1. Allgemeines

**E-Commerce** ist aus unserem Alltag nicht wegzudenken. Die Geschäftstätigkeit von Unternehmen findet heute in erheblichem Umfang in Online-Systemen statt. Und dieser Trend verstärkt sich weiter, schließlich steckt noch immer enormes Potential in der Integration neuer Einsatzgebiete.

Trotz der offensichtlichen Vorteile sollten die Schattenseiten nicht übersehen werden. Denapp Bankdata Service macht es möglich von den Vorteilen des e-Commerce zu profitieren und einhergehende Widrigkeiten abzuwenden.
Mit Denapp Bankdata Service machen Sie einen wichtigen Schritt in Richtung Sicherheit und Kundenfreundlichkeit.

# 2. Verwendung

Der Denapp Bankdata Service (DBS) ist im Internet unter der Adresse **https://www.denapp.com/bankdata/** erreichbar. Um die Leistungen des Dienstes zu nutzen, werden Anfragen an die genannte Internetadresse gesendet.

Der DBS akzeptiert nur sichere Verbindungen, das heißt die angeforderte URL muss mit **https://** beginnen, um die Verwendung von SSL zu aktivieren. Ungesicherte Verbindungen werden mit HTTP-Statuscode **403 - Forbidden** abgelehnt.

Wenn der Dienst die Berechtigung zum Zugriff auf die gewünschte Ressource erteilen kann antwortet er mit dem Ergebnis der Anfrage, andernfalls gibt er den HTTP-Statuscode **401 - Unauthorized** aus.
Zugriff auf den DBS erhalten Kunden der Denapp, die sich für den Dienst registriert und die fällige Nutzungsgebühr in vollem Umfang erbracht haben.

Der DBS ist im Stile eines **REST**ful-Services implementiert. Daraus folgend vereinfachen sich Abfragen im Gegensatz zu herkömmlichen Diensten auf Basis des SOAP-Protokolls merklich. Es ist möglich in einem Webbrowser Anfragen zu erstellen und das Ergebnis zu begutachten. Diese Option ist besonders während der Einführungsphase hilfreich.

> **Hinweis**: *Um das Ergebnis einer Abfrage im Atom-Format (dem Standardformat der Antwort) im Internet Explorer anzuzeigen, muss die Option 'Feedleseanzeige einschalten' deaktiviert sein! Zu finden ist diese Einstellung in 'Internetoptionen' unter 'Feeds und Web Slices'.*

## 2.1 Authentifizierung

Zur Steigerung der Sicherheit und zu Abrechnungszwecken ist es notwendig die Benutzer des Bankdaten-Dienstes zu authentifizieren. Für die Authentifizierung steht die Anmeldung mit einem Benutzer-Alias (Name und Kennwort) zur Verfügung. Die Anmeldedaten ihres Kundenkontos sind nicht für die Anmeldung am Bankdaten-Dienst verwendbar.

Die Einstellungen für die Benutzer-Aliase werden im Kundenbereich auf **http://www.denapp.com** vorgenommen.

Die Anmeldedaten müssen im ‚Basic-Authentication' genannten Verfahren übertragen werden. Dabei werden dem ‚Authorization' - Header der HTTP-Anfrage, Name und Kennwort im ‚base64' codierten Format übergeben. Entsprechende Funktionen sind in den gängigen, HTTP-Requests unterstützenden Programmiersprachen enthalten.

### 2.2 Die Anfrage

Für den DBS ist nur die ‚GET'-Requestmethode zulässig, das heißt, es besteht nur Leseberechtigung, sämtliche zugreifbaren Ressourcen können nur abgerufen aber nicht geändert werden. Anfragen mittels anderer Requestmethoden werden mit HTTP-Statuscode **401 - Unauthorized** abgelehnt.

Alle Ressourcen bzw. Entitäten, auf die zugegriffen werden kann, sind durch URI's adressierbar. DBS-URI's entsprechen der allgemeinen URI-Grammatik und bestehen aus drei Abschnitten:

1. Dienststamm
2. Ressourcenpfad
3. Abfrageoptionen.

#### 2.2.1 Dienststamm

Der URI-Abschnitt mit dem Dienststamm stellt den Speicherort des DBS dar. Die Ressource, die dem Stamm zugeordnet ist, beschreibt alle EntitySet, AssociationSet und/oder Vorgänge des Dienstes. Folgende gültige URI enthält das Dienstnamenelement:

**https://www.denapp.com/bankdata**

#### 2.2.2 Ressourcenpfad

Das Ressourcenpfadelement des URI legt fest, wie Entitätenmenge, Entitätstyp, Navigationseigenschaft, Navigationslink, Eigenschaft, komplexer Typ und Dienstvorgangsnamen zusammengesetzt werden, damit ein URI für die Ressourcenidentifizierung erstellt werden kann.

Das Ressourcenpfadelement des URI ermöglicht das Filtern und Durchsuchen des Entitätendiagramms von DBS. In diesem Abschnitt sind keine Konstrukte zum Ändern des Formats einer Anforderungsantwort enthalten, die an den angegebenen URI gesendet wird.

Folgende gültige URI's enthalten das Dienststamm- und Ressourcenpfadelement:

**https://www.denapp.com/bankdata/DE**

**https://www.denapp.com/bankdata/DE(28)**

**https://www.denapp.com/bankdata/PreviousDE(28)**

### 2.2.3 Abfrageoptionen

Für Abfrageoptionen gelten folgende Regeln:

- In einer einzigen URI-Abfragezeichenfolge kann eine beliebige Anzahl der unterstützten Abfrageoptionen festgelegt werden.
- Jede Abfrageoption kann in einer Abfragezeichenfolge enthalten sein, unabhängig davon, ob die Zeichenfolge bereits andere Optionen enthält.
- Die Reihenfolge der Abfrageoptionen in einem URI spielt keine Rolle.
- In Abfrageoptionsnamen wird die **Groß-/Kleinschreibung** berücksichtigt.
- In Abfrageoptionswerten wird die **Groß-/Kleinschreibung** berücksichtigt.
- In einer Abfragezeichenfolge eines Anforderungs-URI darf eine bestimmte Systemabfrageoption nur einmal enthalten sein.

Zu den Abfrageoptionen gehören die folgenden Direktiven:

| Option | Funktion |
|---|---|
| **$expand**[1] | Gibt zu den Basiselementen verwandte Inhalte zurück. |
| **$filter**[1] | Begrenzt die Menge der zurückgegebenen Elemente. |
| **$orderby**[1] | Legt die Reihenfolge der zurückgegebenen Elemente fest. |
| **$skip**[1] | Überspringt eine bestimmte Anzahl an Elementen. |
| **$top**[1] | Begrenzt die Anzahl der zurückgegebenen Elemente. |
| **$value** | Gibt den Wert einer Eigenschaft ohne umgebende Metadaten zurück. |

[1] Systemabfrageoption

Folgende gültige URI‘s enthalten das Dienststamm-, Ressourcenpfadelement und Abfrageoptionen:

**https://www.denapp.com/bankdata/DE?$filter=BankCode eq '12345678'**

**https://www.denapp.com/bankdata/DE?$filter=BankCode eq '12345678'&$skip=20&$top=10**

#### 2.2.3.1 $expand

Durch eine **Expand** Abfrageoption wird angezeigt, dass jedes Expandelement vollständig erweitert sein soll. Bei der Eigenschaft ganz links in der **Expand** Abfrageoption muss es sich um eine EntityType-Eigenschaft für das EntitySet des letzten Segments in der Abfrage-URI handeln.

Nach der Eigenschaft ganz links in einer **Expand** Abfrageoption (siehe vorherige Regel) müssen alle Eigenschaften in der Expandelement-Klausel Eigenschaften des Resource Identifiers des vorherigen Klauselsegments sein.

Redundante Elemente in der Erweiterungsklausel sind zulässig und werden ignoriert.

#### 2.2.3.2 $filter

Die **Filter** Abfrageoption gibt nur die Elemente aus dem Zielsatz der Ressourcen zurück, die den Filterausdruck erfüllen. Das Ergebnis des Ausdrucks muss ein boolescher Wert oder ein NULL-Werte zulassender boolescher Wert sein. Die zulässigen Operatoren und Funktionen sind unter Punkt **2.2.4** aufgeführt.

Für die **Filter** Syntax gelten die folgenden Regeln:

- Alle bei der Verarbeitung gefundenen syntaktischen und semantischen Fehler werden mit dem HTTP-Statuscode **400 - Bad Request** behandelt.
- Konstanten müssen nach denselben Syntaxregeln dargestellt werden, die für Konstanten im Schlüsselprädikat des URI-Pfads gelten.
- Von **$**-Operatoren werden keine Vergleiche und Operationen für Sätze von Ressourcen unterstützt.

#### 2.2.3.3 $orderby

Hinsichtlich der Syntax übernimmt die Abfrageoption **OrderBy** eine durch Trennzeichen getrennte Liste von ORDER-Klauseln, die jeweils dieselbe Syntax wie die **$filter**-Ausdrücke verwenden und diese optional mit dem Suffix **asc** oder **desc** für eine auf- oder absteigende Sortierung versehen.

Aufsteigend – **asc**
Absteigend – **desc**

### 2.2.4 Operatoren und Funktionen

#### 2.2.4.1 Konstante

Wahr – **true**
Falsch – **false**
Null – **null**

#### 2.2.4.2 Gruppierende Operatoren

Vorrang – **( )**

#### 2.2.4.3 Logische Operatoren

Und – **and**
Oder – **or**
Nicht – **not**
Gleich – **eq**
Ungleich – **ne**
Kleiner als – **lt**
Größer als – **gt**
Kleiner als oder Gleich – **le**
Größer als oder Gleich – **ge**

#### 2.2.4.4 Arithmetische Operatoren

Addition – **add**
Subtraktion – **sub**
Division – **div**
Multiplikation – **mul**
Modulo – **mod**

#### 2.2.4.5 String Funktionen

Boolean **substringof**(String p0, String p1)
Boolean **endswith**(String p0, String p1)
Boolean **startswith**(String p0, String p1)
Int32 **length**(String p0)
Int32 **indexof**(String arg)
String **insert**(String p0, Int32 pos, String p1)
String **remove**(String p0, Int32 pos)
String **remove**(String p0, Int32 pos, Int32 length)
String **replace**(String p0, String find, String replace)
String **substring**(String p0, Int32 pos)
String **substring**(String p0, Int32 pos, Int32 length)
String **tolower**(String p0)
String **toupper**(String p0)
String **trim**(String p0)
String **concat**(String p0, String p1)

#### 2.2.4.6 DateTime Funktionen

Int32 **day**(DateTime p0)
Int32 **hour**(DateTime p0)
Int32 **minute**(DateTime p0)
Int32 **month**(DateTime p0)
Int32 **second**(DateTime p0)
Int32 **year**(DateTime p0)

#### 2.2.4.7 Mathematische Funktionen

Double **round**(Double p0)
Decimal **round**(Decimal p0)
Double **floor**(Double p0)
Decimal **floor**(Decimal p0)
Double **ceiling**(Double p0)
Decimal **ceiling**(Decimal p0)

#### 2.2.4.8 Typ Funktionen

Boolean **isof**(Type p0)
Boolean **isof**(Expression p0, Type p1)
<p0> **cast**(Type p0)
<p1> **cast**(Expression p0, Type p1)

### 2.2.5 Dienstvorgänge

Dienstvorgänge stellen spezielle Methoden des Datendienstes dar. Auf sie kann, wie bei allen anderen Ressourcen des Dienstes, mittels URI's zugegriffen werden. Parameter werden mit Hilfe der URI-Abfragezeichenfolge übergeben. Wie bei anderen Abfragen können Abfrageoptionen auch bei Dienstvorgängen eingesetzt werden. Syntax- oder andere Fehler werden mit dem HTTP-Statuscode **400 - Bad Request** behandelt.

Folgende gültige URI enthält Dienststammelement, Dienstvorgangsnamen und Parameter:

**https://www.denapp.com/bankdata/ValidityDE?bankCode='12345678'&account='1234567890'**

### 2.2.6 Beispiele

Nachfolgend sehen sie einige häufige Anfragen (die eingesetzten Kontonummern und Bankleitzahlen müssen durch reale Werte ersetzt werden):

Prüft die Bankverbindung (BLZ: ‚12345678‘, Kto-Nr.: ‘1234567890‘) auf rechnerische Gültigkeit -
**https://www.denapp.com/bankdata/ValidityDE?bankCode='12345678'&account='1234567890'**

Prüft die IBAN ‚DE12123456789012345678‘ auf rechnerische Gültigkeit -
**https://www.denapp.com/bankdata/ValidityIban?iban='DE12123456789012345678'**

Wie voriges Beispiel mit dem Unterschied, dass das Ergebnis als Klartext zurückgegeben wird -
**https://www.denapp.com/bankdata/ValidityIban/$value?iban='DE121234567890123456778'**

Liefert aus der nach Bankleitzahlen sortierten Liste der Kreditinstitute die Positionen 11 bis 30 -
**https://www.denapp.com/bankdata/DE?$orderby=BankCode&$skip=10&$top=20**

Liefert das als Hauptsitz fungierende Kreditinstitut mit der Bankleitzahl '12345678' -
**https://www.denapp.com/bankdata/DE?$filter=BankCode eq '12345678' and Distinction eq 1**

Liefert eine Sepa Entität mit den BIC und IBAN Werten der konvertierten deutschen Bankverbindung.
**https://www.denapp.com/bankdata/SepaDE?bankCode='12345678'&account='1234567890'**

# 3. Datenstruktur/Schema

Damit sie ihre Anfragen an den DBS erstellen können, benötigen sie einen umfassenden Überblick der Datenstruktur. Der DBS bietet dafür die Möglichkeit das zugrunde liegende Entitätsdatenmodell als XML-Schema unter folgendem URI abzurufen:
**https://www.denapp.com/bankdata/$metadata**.

Unter Verwendung dieses Schemas können Sie für die von Ihnen bevorzugte Programmiersprache ein Datenmodell erstellen lassen.

## 3.1. Beschreibung des Entitätsdatenmodells

Das Datenmodell des DBS ist sehr einfach aufgebaut und besteht aus wenigen Entitätenmengen und keinen Relationen.
Die wichtigste Entitätenmenge ist ‚Validities'. Sie enthält die ‚Validity' Entitäten, in denen die Gültigkeit der in den anderen Entitätenmengen enthaltenen Daten hinterlegt ist. In ‚Validities' wird das jeweilige Datum für das Inkrafttreten der Daten in den anderen Entitätenmengen, in UTC (Coordinated Universal Time) festgehalten.

### 3.1.1 Die DE - Entitätenmengen

Die grundlegenden Daten sind in der Entitätenmenge ‚DE' enthalten, welche die aktuell gültigen ‚DE' Entitäten mit den Datensätzen der deutschen Kreditinstitute zur Verfügung stellt. Weitere Entitätenmengen sind ‚PreviousDE' und ‚NextDE', welche den gleichen Aufbau haben. Während ‚PreviousDE' die Datensätze der deutschen Kreditinstitute des zuletzt gültigen Zeitraums enthält, werden in ‚NextDE' diejenigen des kommenden Zeitabschnitts hinterlegt. Die in ‚NextDE' gespeicherten Daten sind in der Regel mehrere Wochen vor dem Inkrafttreten verfügbar und können zur Vorbereitung auf kommende Änderungen genutzt werden.
In Validities' existiert eine Entität die in der Eigenschaft ‚Name' den Wert ‚DE' enthält und in der Eigenschaft ‚UtcValidFrom' das Datum des Inkrafttretens der in ‚DE' enthaltenen Daten. Wenn die Entität ‚NextDE' keine Daten enthält, ist in der Entitätenmenge ‚Validities' auch kein entsprechender Eintrag mit dem Namen ‚NextDE' vorhanden.

### 3.1.2 Die SepaBanks - Entitätenmenge

Die ‚SepaBanks' Entitätenmenge enthält die aktuell gültigen ‚SepaBank' Entitäten mit den Datensätzen der im SEPA erreichbaren Kreditinstitute.
In Validities' existiert eine Entität die in der Eigenschaft ‚Name' den Wert ‚SepaBanks' enthält und in der Eigenschaft ‚UtcValidFrom' das Datum des Inkrafttretens der in ‚SepaBanks' enthaltenen Daten.

## 3.2. Grafische Darstellung des Entitätsdatenmodells

Die im Datenmodell dargestellten Entitäten werden in folgenden Mengen zusammengefasst:

| Entität | Entitätenmenge |
|---|---|
| **PreviousDE** | PreviousDE |
| **DE** | DE |
| **NextDE** | NextDE |
| **Validity** | Validities |
| **SepaBank** | SepaBanks |
| **Sepa** | Keine Menge |

### 3.3. Beschreibung der Daten für deutsche Kreditinstitute und Filialen

Wie bereits erläutert, enthält die ‚DE' Entitätenmenge die aktuell gültigen Daten der deutschen Kreditinstitute und die Entitätenmengen ‚PreviousDE' und ‚NextDE' diejenigen der letzten und der kommenden Periode.
Der Aufbau der in diesen 3 Entitätenmengen enthaltenen Entitäten (‚PreviousDE', ‚DE', ‚NextDE' – die Bezeichnungen der Entitäten gleichen den Bezeichnungen für die Entitätenmengen) ist identisch und setzt sich folgendermaßen zusammen:

Eigenschaft ‚**BankCode**' (8 nummerische Zeichen):

Enthält die Bankleitzahl zur eindeutigen Identifizierung eines Kreditinstitutes.

Eigenschaft ‚**Distinction**' (8 Bit Integer):

Enthält einen Wert der angibt, ob es sich um ein Bankleitzahl-führendes Kreditinstitut handelt.

Für jede, bei der Deutschen Bundesbank gemeldete Bankleitzahl wird genau eine Entität mit dem Wert ‚1' in der Eigenschaft ‚Distinction' angelegt. Diese Entitäten sind im Zahlungsverkehr zu verwenden.

Wenn die gleiche Bankleitzahl für weitere Filialen eines Kreditinstitutes verwendet wird, werden diese Entitäten mit dem Wert ‚2' in der Eigenschaft ‚Distinction' angelegt. Entitäten mit dem Wert ‚2' werden nicht im Zahlungsverkehr verwendet (Ausnahme siehe Eigenschaft ‚Pan'), sie dienen der ortsbezogenen Suche von Filialen eines Kreditinstitutes.

Eigenschaft ‚**City**' (maximal 35 Zeichen):

Enthält den Ort des Sitzes eines Kreditinstitutes oder einer Filiale.

Eigenschaft ‚**Designation**' (maximal 58 Zeichen):

Enthält einen Wert der die Bezeichnung eines Kreditinstitutes oder einer Filiale darstellt.

Eigenschaft ‚**Pan**‘ (5 Zeichen):

Enthält die Institutsnummer für PAN (Primary Account Number) eines Kreditinstitutes oder einer Filiale.
Für den internationalen Kartenzahlungsverkehr mittels Bankkundenkarten haben die Spitzenverbände des Kreditgewerbes und die Deutsche Bundesbank eine gesonderte Institutsnummerierung festgelegt. Demnach erhält das kartenausgebende Kreditinstitut eine fünfstellige Institutsnummer.

**Zusätzliche Institutsnummer(n) für PAN**
Sofern ein Kreditinstitut weitere Institutsnummern für PAN zu einer Bankleitzahl führt, werden zu der Entität mit dem Wert ‚1‘ in der Eigenschaft ‚Distinction‘ unter dem gleichen Ort zusätzliche Entitäten mit dem Wert ‚2‘ in der Eigenschaft ‚Distinction‘ aufgenommen. Die Entitäten sind bis auf den Wert in der Eigenschaft ‚Distinction‘ und die Institutsnummern in der Eigenschaft ‚Pan‘ identisch.

Anwender (z. B. Netzbetreiber des electronic cash-Systems sowie kreditwirtschaftliche Kopfstellen), die in ihren Anwendungen die Eigenschaft ‚Pan‘ auswerten, müssen daher auch die Entitäten mit dem Wert ‚2‘ in der Eigenschaft ‚Distinction‘ verwenden.

Eigenschaft ‚**Bic**‘ (11 Zeichen):

Enthält den BIC (Bank Identifier Code) eines Kreditinstitutes. Es ist zu beachten, dass durch die unterschiedlichen Aktualisierungsintervalle der Bankleitzahlen durch die Deutsche Bundesbank (vierteljährlich) und der BIC‘s durch das BIC-Directory (monatlich) Unterschiede zwischen den beiden Listen bestehen können.

Eigenschaft ‚**RowId**‘ (32 Bit Integer):

Enthält einen Wert der die eindeutige Nummer der Entität darstellt.

Eigenschaft ‚**Name**‘ (maximal 27 Zeichen):

Enthält die Kurzbezeichnung des Kreditinstitutes unter Hinzufügung des Ortes.

Eigenschaft ‚**Zip**‘ (5 nummerische Zeichen):

Enthält die Postleitzahl des Sitzes eines Kreditinstitutes oder einer Filiale.

Eigenschaft ‚**CheckId**‘ (maximal 2 Zeichen):

Enthält einen Wert der die durch das Kreditinstitut verwendete Prüfzifferberechnungsmethode darstellt. Die Werte dieser Eigenschaft sind maximal 2 Zeichen lang.

Eigenschaft ‚**Deletion**‘ (Boolean):

Enthält einen Wert der angibt, ob die Löschung der Bankleitzahl geplant ist. Für diesen Hinweis gibt es keine zeitliche Festlegung. Die Angabe der beabsichtigten Löschung kann einen oder mehrere Aktualisierungsintervalle vor der Durchführung gesetzt werden. Die Löschung einer Bankleitzahl kann auch ohne eine vorherige Ankündigung vorgenommen werden.
Der Boolesche Wert ‚true‘ bestimmt, dass die Löschung der Bankleitzahl geplant ist.

Eigenschaft ‚**Replacing**‘ (8 nummerische Zeichen):

Enthält eine Nachfolge-Bankleitzahl. Die Eigenschaft enthält entweder den Wert ‚00000000' (Bankleitzahl ist nicht zur Löschung vorgesehen bzw. das Kreditinstitut hat keine Nachfolge-Bankleitzahl veröffentlicht) oder die Angabe einer Bankleitzahl. Eine Bankleitzahl kann angegeben sein, wenn die Eigenschaft ‚Distinction' den Wert ‚1' enthält und die bevorstehende Löschung der Bankleitzahl angekündigt wurde (Eigenschaft ‚Deletion' = ‚true').

### 3.4. Beschreibung der Daten für SEPA Kreditinstitute und Filialen

Wie bereits erläutert, enthält die ‚SepaBanks' Entitätenmenge die aktuell gültigen Daten der im SEPA erreichbaren Kreditinstitute.
Der Aufbau der in dieser Entitätenmenge enthaltenen Entitäten (‚SepaBank') setzt sich folgendermaßen zusammen:

Eigenschaft ‚**Bic**' (11 Zeichen):

Enthält den BIC (Bank Identifier Code) des Kreditinstitutes.

Eigenschaft ‚**Name**' (maximal 105 Zeichen):

Enthält den Namen des Kreditinstitutes.

### 3.5 Beschreibung der Dienstvorgänge

#### 3.5.1 DesignationDE

**Beschreibung**:

Enthält die Bezeichnung des bankleitzahlführenden (Eigenschaft ‚Distinction' = ‚1') deutschen Kreditinstitutes mit der in ‚bankCode' enthaltenen Bankleitzahl der aktuell gültigen Entitäten.

**Parameter**:

- **bankCode:**
  (String) Die Bankleitzahl des deutschen Kreditinstitutes.

**Return**:

(String) Die Bezeichnung des Kreditinstitutes.

### 3.5.2 ValidityDateUtcPreviousDE

**Beschreibung**:

Enthält das Datum und die Zeit des Inkrafttretens der im vergangenen Zeitintervall gültigen Entitäten in UTC.

**Parameter**:

keine

**Return**:

(DateTime) Der Zeitpunkt des Inkrafttretens.

### 3.5.3 ValidityDateUtcDE

**Beschreibung**:

Enthält das Datum und die Zeit des Inkrafttretens der im aktuellen Zeitintervall gültigen Entitäten in UTC.

**Parameter**:

keine

**Return**:

(DateTime) Der Zeitpunkt des Inkrafttretens.

### 3.5.4 ValidityDateUtcNextDE

**Beschreibung**:

Enthält das Datum und die Zeit des Inkrafttretens der im kommenden Zeitintervall gültigen Entitäten in UTC.

**Parameter**:

keine

**Return**:

(DateTime) Der Zeitpunkt des Inkrafttretens.

### 3.5.5 ValidityDE

**Beschreibung**:

Enthält die Gültigkeit, als 32 Bit Integer, der durch die Parameter ‚bankCode' und ‚accountNumber' definierten deutschen Bankverbindung.

> **Achtung**: *Die Bankverbindung wird nach aktuell gültigen, durch die deutsche Bundesbank herausgegebenen Verfahren überprüft. Es kann jedoch nicht festgestellt werden, dass die Bankverbindung auch tatsächlich existent und in Gebrauch ist. Die Verwendung einer als gültig geprüften Bankverbindung geschieht daher auf eigene Gefahr.*

**Parameter**:

- **bankCode:**
  (String) Die Bankleitzahl der Bankverbindung.
- **account:**
  (String) Die Kontonummer der Bankverbindung.

**Return**:

(Int32) Eine codierte Darstellung der Gültigkeit der deutschen Bankverbindung.
Möglich sind die folgenden Werte:

- **0 – Success**: Die Bankleitzahl ist an ein Kreditinstitut vergeben und die Kontonummer ist rechnerisch gültig.
- **1 – Error**: Während der Prüfung ist ein unerwarteter Fehler aufgetreten. Bankleitzahl und Kontonummer sind nicht prüfbar.
- **2 – NotTestable**: Die Bankleitzahl ist an ein Kreditinstitut vergeben. Über die rechnerische Gültigkeit der Kontonummer kann, bedingt durch den Prüfalgorithmus, keine abschließende Aussage getroffen werden.
- **3 – NoBankCode**: Es wurde keine Bankleitzahl übergeben. Daraus folgend konnte die Kontonummer nicht geprüft werden.
- **4 – InvalidBankCode**: Die übergebene Bankleitzahl ist nicht vergeben. Daraus folgend konnte die Kontonummer nicht geprüft werden.
- **5 – BankCodeTooLong**: Die Länge der Bankleitzahl ist größer 8. Daraus folgend konnte die Kontonummer nicht geprüft werden.
- **6 – BankCodeTooShort**: Die Länge der BLZ ist kleiner 8. Daraus folgend konnte die Kontonummer nicht geprüft werden.
- **7 – InvalidBankCodeCharacter**: Die BLZ enthält ungültige Zeichen. Daraus folgend konnte die Kontonummer nicht geprüft werden.
- **8 – NoAccountNumber**: Es wurde keine Kontonummer übergeben.
- **9 – AccountNumberTooLong**: Die Länge der Kontonummer ist größer 10.
- **10 – InvalidAccountNumberCharacter**: Die Kontonummer enthält ungültige Zeichen.
- **11 – InvalidAccountNumberRestriction**: Die Kontonummer erfüllt eine Einschränkung nicht.
- **12 - InvalidAccountNumberCheckSum** - Die in der Kontonummer enthaltene Prüfsumme entspricht nicht der rechnerisch ermittelten Prüfsumme.

### 3.5.6 Validitylban

**Beschreibung**:

Enthält die Gültigkeit, als 32 Bit Integer, der im Parameter ‚iban' enthaltenen IBAN (International Bank Account Number).

> **Achtung**: *Die IBAN wird nach aktuell gültigen Verfahren überprüft. Eine evtl. zugrunde liegende deutsche Bankverbindung wird nach aktuell gültigen, durch die deutsche Bundesbank herausgegebenen Verfahren überprüft. Es kann jedoch nicht festgestellt werden, dass die Bankverbindung auch tatsächlich existent und in Gebrauch ist. Die Verwendung einer als gültig geprüften Bankverbindung geschieht daher auf eigene Gefahr.*

**Parameter**:

- **iban**:
  (String) Die IBAN.

**Return**:

(Int32) Eine codierte Darstellung der Gültigkeit der IBAN.
Möglich sind die folgenden Werte:

- **0 – Success**: Die IBAN ist rechnerisch gültig. Wenn es sich um die IBAN einer deutschen Bankverbindung handelt, so ist auch diese rechnerisch gültig.
- **1 – Error**: Während der IBAN-Prüfung ist ein unerwarteter Fehler aufgetreten.
- **2 – NotTestable**: Über die Gültigkeit der IBAN kann, bedingt durch den Prüfalgorithmus keine abschließende Aussage getroffen werden.
- **3 – NoIBAN**: Es wurde keine IBAN übergeben.
- **4 – IBANTooLong**: Die Länge der IBAN ist größer 34.
- **5 – IBANTooShort**: Die Länge der IBAN ist kleiner 5.
- **6 – InvalidCharacter**: Die IBAN enthält ungültige Zeichen.
- **7 – InvalidCountryCode**: Die IBAN enthält einen ungültigen Ländercode.
- **8 – InvalidRegionalLength**: Die Länge der IBAN entspricht nicht dem durch den Ländercode vorgegebenen Wert.
- **9 – InvalidRestriction**: Die IBAN erfüllt eine Einschränkung nicht.
- **10 – InvalidCheckSum**: Die in der IBAN enthaltene Prüfsumme entspricht nicht der rechnerisch ermittelten Prüfsumme.

Wenn der IBAN eine deutsche Bankverbindung zugrunde liegt, kann einer der folgenden Werte enthalten sein (entspricht den Werten des Dienstvorgangs ‚ValidityDE' um 16 Bit verschoben, das heißt alle Werte sind ein Vielfaches von 65536):

- **65536 – Error**: Während der Prüfung ist ein unerwarteter Fehler aufgetreten. Bankleitzahl und Kontonummer sind nicht prüfbar.
- **131072 – NotTestable**: Die Bankleitzahl ist an ein Kreditinstitut vergeben. Über die rechnerische Gültigkeit der Kontonummer kann, bedingt durch den Prüfalgorithmus, keine abschließende Aussage getroffen werden.

- **196608 – NoBankCode**: Es wurde keine Bankleitzahl übergeben. Daraus folgend konnte die Kontonummer nicht geprüft werden.
- **262144 – InvalidBankCode**: Die übergebene Bankleitzahl ist nicht vergeben. Daraus folgend konnte die Kontonummer nicht geprüft werden.
- **327680 – BankCodeTooLong**: Die Länge der Bankleitzahl ist größer 8. Daraus folgend konnte die Kontonummer nicht geprüft werden.
- **393216 – BankCodeTooShort**: Die Länge der BLZ ist kleiner 8. Daraus folgend konnte die Kontonummer nicht geprüft werden.
- **458752 – InvalidBankCodeCharacter**: Die BLZ enthält ungültige Zeichen. Daraus folgend konnte die Kontonummer nicht geprüft werden.
- **524288 – NoAccountNumber**: Es wurde keine Kontonummer übergeben.
- **589824 – AccountNumberTooLong**: Die Länge der Kontonummer ist größer 10.
- **655360 – InvalidAccountNumberCharacter**: Die Kontonummer enthält ungültige Zeichen.
- **720896 – InvalidAccountNumberRestriction**: Die Kontonummer erfüllt eine Einschränkung nicht.

### 3.5.7 ValiditySepa

**Beschreibung**:

Enthält die Gültigkeit, als 32 Bit Integer, der durch die Parameter ‚bic' und ‚iban' definierten SEPA-Bankverbindung (SEPA – Single Euro Payments Area).

> **Achtung**: *Der BIC und die IBAN der SEPA-Bankverbindung werden nach aktuell gültigen Verfahren überprüft. Eine evtl. zugrunde liegende deutsche Bankverbindung wird nach aktuell gültigen, durch die deutsche Bundesbank herausgegebenen Verfahren überprüft und der BIC wird mit dem Kreditinstitut der deutschen Bankdaten verglichen. Es kann jedoch nicht festgestellt werden, dass die Bankverbindung auch tatsächlich existent und in Gebrauch ist. Die Verwendung einer als gültig geprüften Bankverbindung geschieht daher auf eigene Gefahr.*

**Parameter**:

- **bic**:
  (String) Der BIC.
- **iban**:
  (String) Die IBAN.

**Return**:

(Int32) Eine codierte Darstellung der Gültigkeit der SEPA-Bankverbindung.
Möglich sind die folgenden Werte (entspricht den Werten des Dienstvorgangs ‚ValidityIban'):

- **0 – Success**: Der BIC ist an ein SEPA Kreditinstitut vergeben und die IBAN ist rechnerisch gültig. Wenn es sich um die IBAN einer deutschen Bankverbindung handelt, so ist auch diese rechnerisch gültig und der BIC entspricht dem Kreditinstitut der deutschen Bankverbindung.
- **1 – Error**: Während der IBAN-Prüfung ist ein unerwarteter Fehler aufgetreten.

- **2 – NotTestable**: Über die Gültigkeit der IBAN kann, bedingt durch den Prüfalgorithmus keine abschließende Aussage getroffen werden.
- **3 – NoIBAN**: Es wurde keine IBAN übergeben.
- **4 – IBANTooLong**: Die Länge der IBAN ist größer 34.
- **5 – IBANTooShort**: Die Länge der IBAN ist kleiner 5.
- **6 – InvalidCharacter**: Die IBAN enthält ungültige Zeichen.
- **7 – InvalidCountryCode**: Die IBAN enthält einen ungültigen Ländercode.
- **8 – InvalidRegionalLength**: Die Länge der IBAN entspricht nicht dem durch den Ländercode vorgegebenen Wert.
- **9 – InvalidRestriction**: Die IBAN erfüllt eine Einschränkung nicht.
- **10 – InvalidCheckSum**: Die in der IBAN enthaltene Prüfsumme entspricht nicht der rechnerisch ermittelten Prüfsumme.

Wenn der IBAN eine deutsche Bankverbindung zugrunde liegt, kann einer der folgenden Werte enthalten sein (entspricht den Werten des Dienstvorgangs ‚ValidityDE' um 16 Bit verschoben, das heißt alle Werte sind ein Vielfaches von 65536):

- **65536 – Error**: Während der Prüfung ist ein unerwarteter Fehler aufgetreten. Bankleitzahl und Kontonummer sind nicht prüfbar.
- **131072 – NotTestable**: Die Bankleitzahl ist an ein Kreditinstitut vergeben. Über die rechnerische Gültigkeit der Kontonummer kann, bedingt durch den Prüfalgorithmus, keine abschließende Aussage getroffen werden.
- **196608 – NoBankCode**: Es wurde keine Bankleitzahl übergeben. Daraus folgend konnte die Kontonummer nicht geprüft werden.
- **262144 – InvalidBankCode**: Die übergebene Bankleitzahl ist nicht vergeben. Daraus folgend konnte die Kontonummer nicht geprüft werden.
- **327680 – BankCodeTooLong**: Die Länge der Bankleitzahl ist größer 8. Daraus folgend konnte die Kontonummer nicht geprüft werden.
- **393216 – BankCodeTooShort**: Die Länge der BLZ ist kleiner 8. Daraus folgend konnte die Kontonummer nicht geprüft werden.
- **458752 – InvalidBankCodeCharacter**: Die BLZ enthält ungültige Zeichen. Daraus folgend konnte die Kontonummer nicht geprüft werden.
- **524288 – NoAccountNumber**: Es wurde keine Kontonummer übergeben.
- **589824 – AccountNumberTooLong**: Die Länge der Kontonummer ist größer 10.
- **655360 – InvalidAccountNumberCharacter**: Die Kontonummer enthält ungültige Zeichen.
- **720896 – InvalidAccountNumberRestriction**: Die Kontonummer erfüllt eine Einschränkung nicht.

Zusätzlich können folgende Werte enthalten sein:

- **16 – Error**: Während der BIC-Prüfung ist ein unerwarteter Fehler aufgetreten.
- **32 - NotTestable**: Über die Gültigkeit des BIC kann, bedingt durch den Prüfalgorithmus keine abschließende Aussage getroffen werden.
- **48 – NoBIC**: Es wurde kein BIC übergeben.
- **64 – InvalidLength**: Die Länge des BIC ist ungleich 8 oder 11.
- **80 – InvalidCharacter**: Der BIC enthält ungültige Zeichen.
- **96 – InvalidCountryCode**: Der BIC enthält einen ungültigen Ländercode.
- **112 – InvalidRestriction**: Der BIC erfüllt eine Einschränkung nicht.
- **256 – Error**: Während der SEPA-Prüfung ist ein unerwarteter Fehler aufgetreten.

- **512 – NotTestable**: Über die Gültigkeit der SEPA-Bankverbindung kann keine abschließende Aussage getroffen werden.
- **768 – InvalidRestriction**: Die SEPA-Bankverbindung erfüllt eine Einschränkung nicht.

### 3.5.8 IbanDE

**Beschreibung**:

Enthält die IBAN-Entsprechung der durch Bankleitzahl und Kontonummer definierten deutschen Bankverbindung.

> **Achtung**: *Die IBAN wird unter Verwendung der aktuell gültigen Verfahren ermittelt. Zur Herausgabe von IBAN's sind jedoch nur die Kreditinstitute berechtigt. Für die Gültigkeit der abgerufenen IBAN übernimmt die Denapp keine Gewähr.*

Die übergebenen Werte für Bankleitzahl und Kontonummer werden vor der Auswertung auf rechnerische Gültigkeit geprüft. Abweichende Schreibweisen und Abkürzungen der Kontonummer werden erkannt und entsprechend korrigiert.

**Parameter**:

- **bankCode:**
  (String) Die Bankleitzahl der Bankverbindung.
- **account:**
  (String) Die Kontonummer der Bankverbindung.

**Return**:

(String) Die IBAN der durch die Parameter definierten deutschen Bankverbindung.

### 3.5.9 SepaDE

**Beschreibung**:

Enthält die SEPA-Entsprechung der durch Bankleitzahl und Kontonummer definierten deutschen Bankverbindung.

Die übergebenen Werte für Bankleitzahl und Kontonummer werden vor der Auswertung auf rechnerische Gültigkeit geprüft. Abweichende Schreibweisen und Abkürzungen der Kontonummer werden erkannt und entsprechend korrigiert.

> **Achtung**: *Die SEPA-Bankverbindung wird unter Verwendung der aktuell gültigen Verfahren ermittelt. Zur Herausgabe von BIC's und IBAN's sind jedoch nur die Kreditinstitute berechtigt. Für die Gültigkeit der abgerufenen SEPA-Bankverbindung übernimmt die Denapp keine Gewähr.*

**Parameter**:

- **bankCode:**
  (String) Die Bankleitzahl der Bankverbindung.
- **account:**
  (String) Die Kontonummer der Bankverbindung.

**Return**:

(Sepa) BIC und IBAN der durch die Parameter definierten deutschen Bankverbindung.

# 4. Beispiele

Weitere Informationen finden Sie im Internet unter **http://www.denapp.com**.
Bitte setzen Sie sich mit uns in Verbindung wenn Sie Fragen haben, wir beraten Sie gern!

## 4.1 Code-Beispiele

Nachfolgend sehen Sie einige einfache Beispiele für das Erstellen einer Anfrage in verschiedenen Programmiersprachen:

**C#**

```
HttpWebRequest request =
(HttpWebRequest)WebRequest.Create("https://www.denapp.com/bankdata/DE(12345)");
request.Method = "GET";
request.Accept = "application/json";

HttpWebResponse response = (HttpWebResponse) request.GetResponse();
StreamReader reader = new StreamReader(response.GetResponseStream());
StringBuilder output = new StringBuilder();
output.Append(reader.ReadToEnd());

response.Close();

//'output' enthält das Ergebnis der Anfrage im JSON-Format
```

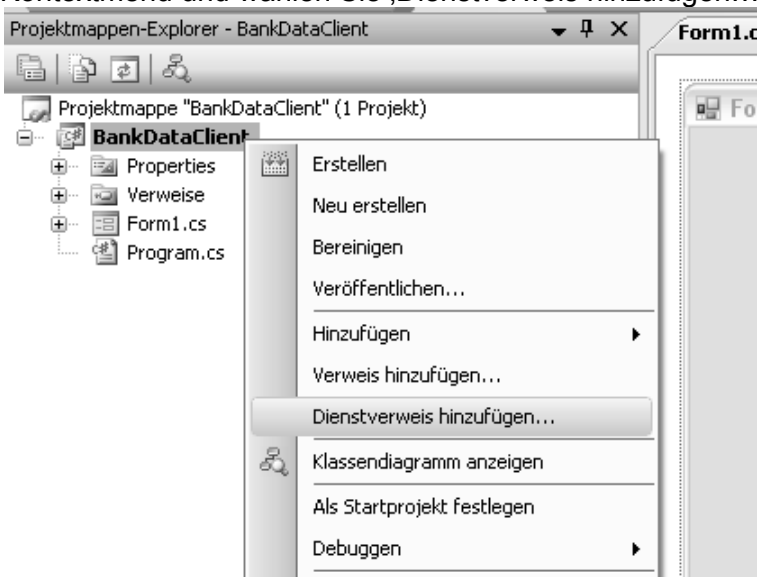
**VB**

```
Dim request As HttpWebRequest =
CType(WebRequest.Create("https://www.denapp.com/bankdata/DE(12345)"), HttpWebRequest)
request.Method = "GET"
request.Accept = "application/json"

Dim response As HttpWebResponse = CType(request.GetResponse(), HttpWebResponse)
Dim reader As StreamReader = New StreamReader(response.GetResponseStream())
Dim output As StringBuilder = New StringBuilder()
output.Append(reader.ReadToEnd())


response.Close()


REM 'output' enthält das Ergebnis der Anfrage im JSON-Format
```

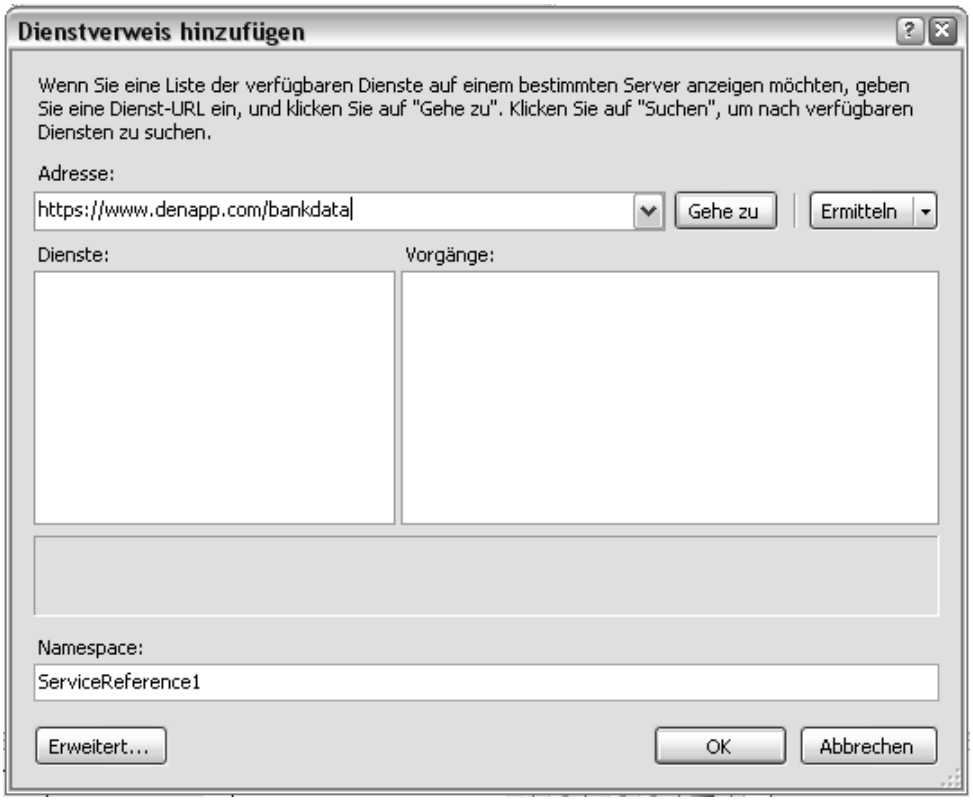
### 4.2 Code-Generierung unter Microsoft Visual Studio 2008 (.Net Framework 3.5)

Die folgende Beschreibung erläutert das Einbinden des Denapp Bankdata Service unter Verwendung von Microsoft Visual Studio.

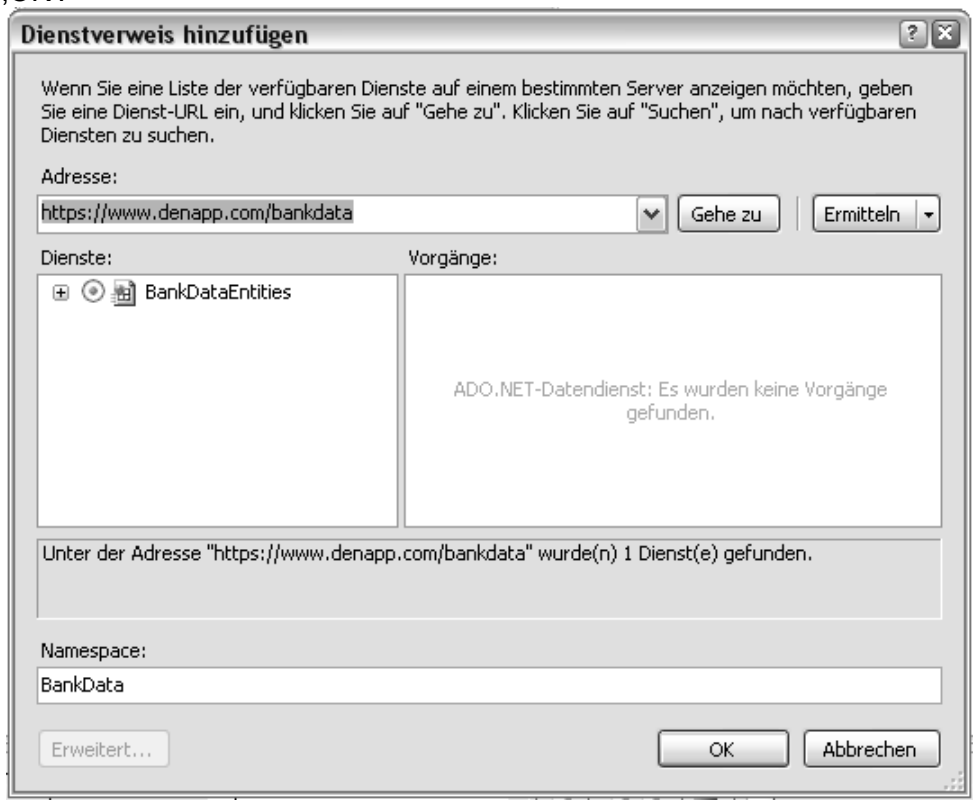
1. Öffnen Sie mit Rechtsklick auf den Projekt-Knoten im Projektmappen-Explorer das Kontextmenü und wählen Sie ‚Dienstverweis hinzufügen…'




2. Geben Sie die URL ‚https://www.denapp.com/bankdata' im Feld Adresse an und klicken Sie
   auf ‚Gehe zu'.
   Bestätigen Sie die Meldung, dass die Anforderung authentifiziert werden muss(die Meldung, dass die Daten unverschlüsselt übertragen werden ist irreführend, da es sich um eine SSL-Verbindung handelt).
   Geben Sie anschließend Benutzernamen und Kennwort ihres Kontos an(evtl. erscheint die Authentifizierungsmeldung zwei mal).




3. Geben Sie einen Namespace für den zu erstellenden Code an und bestätigen Sie mit ‚OK'.



Wenn alle Schritte erfolgreich durchgeführt wurden hat die IDE den Code für die Darstellung der BankData-Entitäten für Sie erstellt.

Nachfolgend sehen Sie ein C# Programm-Beispiel für die Nutzung des gerade erstellten Codes.

**C#**

```
Uri uri = new Uri("https://www.denapp.com/bankdata");
BankData.BankDataEntities entities = new BankData.BankDataEntities(uri);
entities.Credentials = new System.Net.NetworkCredential("Benutzername", "Kennwort");

var query = from de in entities.DE
            where de.BankCode == "12345678" && de.Distinction == 1
            select de;


//die Variable 'query' ist vom Typ IQueryable<DE> und enthält die gesuchte DE-Entität,
//wenn diese gefunden wurde
```

## 5. Rechtliche Hinweise

Diese Dokumentation, einschließlich der darin enthaltenen Beispielanwendungen, wird nur zu Informationszwecken zur Verfügung gestellt und Denapp übernimmt in dieser Dokumentation keine Gewährleistungen, weder ausdrücklich noch konkludent. Die in dieser Dokumentation enthaltenen Angaben und Daten, einschließlich URL‘s und andere Verweise auf Internetwebsites können ohne vorherige Ankündigung geändert werden. Das vollständige Risiko der Nutzung oder der Ergebnisse der Nutzung dieser Dokumentation liegt bei dem Benutzer.

Es ist möglich, dass Denapp Rechte an Patenten bzw. angemeldeten Patenten, an Marken, Urheberrechten oder sonstigem geistigen Eigentum besitzt, die sich auf den fachlichen Inhalt dieser Dokumentation beziehen. Das Bereitstellen dieser Dokumentation gibt Ihnen jedoch keinen Anspruch auf diese Patente, Marken, Urheberrechte oder auf sonstiges geistiges Eigentum, es sei denn, dies wird ausdrücklich in einem Lizenzvertrag von Denapp eingeräumt.



Denapp ist eine eingetragene Marke der Denapp UG (haftungsbeschränkt) in Deutschland und/oder anderen Ländern.

Microsoft ist eine eingetragene Marke der Microsoft Corporation in den USA und/oder anderen Ländern.

Alle anderen Marken sind Eigentum ihrer jeweiligen Inhaber.