AERA Webservices

Inhalt

REST Paradigma

AERA Webservices sind nach dem REST Paradigma aufgebaut. Ressourcen werden als URL dargestellt. Mit den von AERA definierten Ressourcen kann wie folgt interagiert werden.

HTTP request method	**Description**
POST	Create (Erstellen von neuen Ressourcen)
GET	Read (Lesen einer Ressource)
PUT	Update (Aktualisieren einer vorhandenen Ressource)
DELETE	Delete (Löschen einer vorhandenen Ressource)

Unterstützte Versionen

Version designation	**Description**
V1	Version 1 des Webservices

Im Nachfolgenden wird hierfür **[Version]** verwendet.

Alle Ressourcen können ebenfalls eine separate Versionsnummer erhalten, um eine Erweiterung der Funktionalität und eine rückwärts Kompatibilität zu ermöglichen.

Unterstützte Lokalisierungen

IsoCode	**Definition**
de-de	Deutsch-Deutschland

Im Nachfolgenden wird hierfür **[Locale]** verwendet.

Server Adresse

Die API ist über folgende URL erreichbar. Alle Ressourcen werden dieser URL angehängt.

```
https://api.aera-online.com/[Version]/[Locale]/
```

Im Nachfolgenden wird hierfür [Server] verwendet.

Unterstützte HTTP Statuscodes

HTTP statuscode	Description
200	OK
401	Unauthorized (Sitzung benötigt, abgelaufen oder ungültig)
404	Resource not found
500	Interner Serverfehler (Anforderungsfehler)

Wenn der aufrufende Benutzer keine Berechtigung hat, eine Ressource aufzurufen, wird anstelle des Statuscodes 403 der Code 500 zurückgegeben.

Unterstützte Dokumententypen

Documenttype	Example
JSON (Default)	…/AnyResource1/AnyResource2
XML	…/AnyResource1/AnyResource2?documenttype=xml

Bestimmt das Format der Daten für den Request- und Response-Stream.
Der Dokumententyp kann über den Parameter „documenttype“ angegeben werden. Standardmäßig kommuniziert der Webservice mit dem Dokumenttyp JSON.

Manche Ressourcen unterstützten zusätzlich andere Dokumenttypen, dies wird entsprechend angegeben.

Kompatibilität

Änderungen an Objekten können bei einer verwendeten Schnittstelle auftauchen, ohne dass eine andere Version der Schnittstelle bereitgestellt wird. Daher sollte immer anhand des Element-Namens auf ein Element im Dokument zugegriffen werden. Niemals auf die Feldreihenfolge oder Anzahl verlassen! Diese kann sich ändern.

Beschreibung Datentypen

Aktuell werden folgende Datentypen von dem Webservice verwendet.

Data type	Description
Guid	Global eindeutige Nummer
Boolean	Einfacher Boolescher Wert
Byte	Ein ganzzahliger Wert im 8 Bit Wertebereich
Short	Ein ganzzahliger Wert mit Vorzeichen im 16 Bit Wertebereich
Integer	Ein ganzzahliger Wert mit Vorzeichen im 32 Bit Wertebereich
Long	Ein ganzzahliger Wert mit Vorzeichen im 64 Bit Wertebereich
Single	Eine 32-Bit-Gleitkommazahlen mit einfacher Genauigkeit
Double	Eine 64-Bit-Gleitkommazahlen mit doppelter Genauigkeit
Decimal (n,m)	Eine Dezimalzahl mit einer Genauigkeit von n und m Dezimalstellen
String	Eine Zeichenkette
String (n)	Eine Zeichenkette mit einer Maximallänge von n
Date	Ein Datumswert ohne Uhrzeit
DateTime	Ein Datumswert mit Uhrzeit

AERA Objekte

Die AERA Objekte stellen die Metadatenschicht des Webservices dar. Diese werden für den Datenaustausch verwendet.

Es gilt folgende Struktur für ExampleObject.

Name	Data type	Description
FieldInteger	Integer	Beispielinteger
FieldDecimal	Decimal (5,4)	Beispieldezimalwert
FieldString	String	Beispieltext
FieldDateTime	DateTime	Beispieldatum mit Uhrzeit

Das Objekt ExampleObjectList stellt eine AERA Objektliste von ExampleObject dar.

Beispiel einfaches AERA Objekt: ExampleObject

JSON

```
{
     "Data": {
          "ExampleObject": {
               "FieldDateTime": "2020-12-31T23:59:59",
               "FieldDecimal": 1.2345,
               "FieldInteger": 12345,
               "FieldString": "Hello World"
          }
     }
}
```

XML

```
<?xml version="1.0"?>
<AERA>
     <Data>
          <ExampleObject>
               <FieldDateTime>2020-12-31T23:59:59</FieldDateTime>
               <FieldDecimal>1.2345</FieldDecimal>
               <FieldInteger>12345</FieldInteger>
               <FieldString>Hello World</FieldString>
          </ExampleObject>
     </Data>
</AERA>
```

Beispiel einfache AERA Objektliste: ExampleObjectList mit ExampleObject

JSON

```
{
     "Data": {
          "ExampleObjectList": {
               "Items": [
                    {
                         "FieldDateTime": "2020-12-31T23:59:59",
                         "FieldDecimal": 1.2345,
                         "FieldInteger": 12345,
                         "FieldString": "Hello World"
                    },
                    {
                         "FieldDateTime": "2020-01-31T00:00:00",
                         "FieldDecimal": 5.4321,
                         "FieldInteger": 54321,
                         "FieldString": "dlroW olleH"
                    }
               ]
          }
     }
}
```

XML

```
<?xml version="1.0"?>
<AERA>
     <Data>
          <ExampleObjectList>
               <Items type="list">
                    <ExampleObject>
                         <FieldDateTime>2020-12-31T23:59:59</FieldDateTime>
                         <FieldDecimal>1.2345</FieldDecimal>
                         <FieldInteger>12345</FieldInteger>
                         <FieldString>Hello World</FieldString>
                    </ExampleObject>
                    <ExampleObject>
                         <FieldDateTime>2020-01-31T00:00:00</FieldDateTime>
                         <FieldDecimal>5.4321</FieldDecimal>
                         <FieldInteger>54321</FieldInteger>
                         <FieldString>dlroW olleH</FieldString>
                    </ExampleObject>
               </Items>
          </ExampleObjectList>
     </Data>
</AERA>
```

Anfrage

Alle Ressourcen werden in den AERA Webservice Dokumentationen wie folgt angegeben.

Example Resource

Beschreibung und Funktionsweise der Ressource

URL	[Server]/anyresource/[Parameter1]?parameter2= [Parameter2]&…
HTTP Method	*HTTP Method*

Parameter

Name	Data type	Required	Description
Parameter1	*AnyType*	*X*	*Beschreibung und Definitionen*
Parameter2	*AnyType*		*Beschreibung und Definitionen*

Request data: *Datentyp* (mit *Typ des Einzelsatzes bei Listen*)

Die Struktur beschreibt den Aufbau des Einzelsatzes.

Name	Data type	Required	Description
FieldNameData	*AnyType*	*X*	*Beschreibung und Definitionen*

Result: *Rückgabetyp* (mit *Typ des Einzelsatzes bei Listen*)

Die Struktur beschreibt den Aufbau des Einzelsatzes.

Name	Data type	Description
FieldNameResult	*AnyType*	*Beschreibung und Definitionen*

Sonstige Hinweise zur Funktionalität.

Beschreibung Required

Felder können als *Required* gekennzeichnet sein. Dies bedeutet, dass diese nicht dem Standardwert des Datentyps entsprechen dürfen. Im Falle von Integer darf somit nicht der Wert 0 übergeben werden.
Dementsprechend müssen diese Felder angegeben werden, wenn nicht ein anderer Wert als Standard angegeben ist.

Beschreibung Parameter

Parameter sind ein Teil der URL oder werden im „QueryString“ angefügt.

Beschreibung Request data

Die Daten, meist Dokumente (JSON, XML, …), die im Request-Stream mitgeschickt werden.

Beschreibung Result

Die Daten, die von der Ressource zurückgegeben werden.

Antwort

Der Aufbau der Antwort folgt bei AERA Schnittstellen einem vordefinierten Schema mit drei Hauptelementen.

Data Element

Behälter für die eigentlichen Daten in Form von AERA Objekten.

DistributedException Element

Enthält im Fehlerfall für die Anzeige aufbereitete Fehlerinformationen.

Name	Data type	Description
Subject	String	Fehlerbezeichnung
SubjectId	Integer	Eindeutige Fehlernummer
Message	String	Fehlerbeschreibung
MessageId	Integer	Eindeutige Fehlernummer
Source	String	Fehlerquelle

Incident Element

Enthält IncidentItem Listen, welche optionale Informationen, Warnungen und Fehler darstellen. Jede dieser Listen kann 0 bis n Einträge enthalten. Wenn keine Ereignisse während der Verarbeitung aufgetreten sind, ist das Incident Element leer und es wird nicht mitgeschickt.

Incident Item

Repräsentiert ein einzelnes Ereignis in einem Incident Element. Es wird in die Gruppen *Error*, *Warnings* und *Informations* einsortiert.

Name	Data type	Description
Subject	String	Fehlerbezeichnung
Message	String	Fehlerbeschreibung
MessageId	Integer	Eindeutige Fehlernummer

Mögliche Kombinationen

Aufgrund dessen, dass entweder ein Fehler auftritt oder die Anfrage erfolgreich ausgeführt wurde, ergeben sich folgende mögliche Kombinationen.

```
Data
Data + Incident
DistributedException
DistributedException + Incident
```

Beispiel JSON

Data + Incident

```
{
     "Data": {…},
     "Incident": {
          "Errors": [
               {
                    "MessageId": …,
                    "Message": "…",
                    "Subject": "…"
               },
               {…}
          ],
          "Warnings": [
               {…}
          ],
          "Informations": []
     }
}
```

DistributedException + Incident

```
{
     "DistributedException": {
          "MessageId": …,
          "Message": "…",
          "SubjectId": …,
          "Subject": "…",
          "Source": "…"
     },
     "Incident": {
          "Errors": […],
          "Warnings": […],
          "Informations": […]
     }
}
```

Beispiel XML

Data + Incident

```
<?xml version="1.0"?>
<Aera>
     <Data>…</Data>
     <Incident>
          <Errors type="list">
               <Item>
                    <MessageId>…</MessageId>
                    <Message>…</Message>
                    <Subject>…</Subject>
               </Item>
               <Item>
                    <MessageId>…</MessageId>
                    <Message>…</Message>
                    <Subject/>
               </Item>
               <Item>…</Item>
          </Errors>
          <Warnings type="list">
               <Item>…</Item>
          </Warnings>
          <Informations type="list"></Informations>
     </Incident>
</Aera>
```

DistributedException + Incident

```
<?xml version="1.0"?>
<Aera>
     <DistributedException>
          <MessageId>…</MessageId>
          <Message>…</Message>
          <SubjectId>…</SubjectId>
          <Subject>…</Subject>
          <Source>…</Source>
     </DistributedException>
     <Incident>
          <Errors type="list">…</Errors>
          <Warnings type="list">…</Warnings>
          <Informations type="list">…</Informations>
     </Incident>
</Aera>
```

Beispiel: Failed Login

In diesem Beispiel wurde versucht sich auf dem Webservice anzumelden. Es wurde ein gültiger LoginName übergeben, jedoch kein Passwort.

JSON

```
{
     "DistributedException": {
          "MessageId": 0,
          "Message": "Validation failed.",
          "SubjectId": 0,
          "Subject": "ValidationFailed",
          "Source": "CreateUserSessionV1"
     },
     "Incident": {
          "Errors": [
               {
                    "MessageId": 0,
                    "Message": "Incorrect value in StringLegacySlot.",
                    "Source": "CreateUserSessionData.Password"
               }
          ],
          "Warnings": [],
          "Informations": []
     }
}
```

XML

```
<?xml version="1.0"?>
<Aera>
     <DistributedException>
          <MessageId>0</MessageId>
          <Message>Validation failed.</Message>
          <SubjectId>0</SubjectId>
          <Subject>ValidationFailed</Subject>
          <Source>CreateUserSessionV1</Source>
     </DistributedException>
     <Incident>
          <Errors type="list">
               <Item>
                    <MessageId>0</MessageId>
                    <Message>Incorrect value in StringLegacySlot.</Message>
                    <Source>CreateUserSessionData.Password</Source>
               </Item>
          </Errors>
          <Warnings type="list"/>
          <Informations type="list"/>
     </Incident>
</Aera>
```

Erste Schritte

Anmelden

Durch das Anmelden erhält man eine Sitzungsnummer die als Authentifizierung gegenüber dem Webservice zu verwenden ist. Die meisten Funktionen setzen eine gültige Anmeldung voraus.

URL	[Server]/login
HTTP Method	POST

Parameter

Keine Parameter benötigt.

Request data: CreateUserSessionData

Name	Data type	Required	Description
LoginName	String (100)	X	Benutzername oder Kundennummer
Password	String (50)	X	Kennwort für das entsprechende Konto

Result: UserSessionForInfo

Name	Data type	Description
Id	Guid	Sitzungsnummer
UserId	Integer	Kundennummer
UserName	String	Benutzername
Email	String	E-Mail Adresse des Kontos
UserCompanies	UserCompanyList	Siehe UserCompanyList mit UserCompany (Liste von Firmen auf die der User Zugriff hat)

UserCompanyList mit UserCompany (Liste von Firmen auf die der User Zugriff hat)

Name	Data type	Description
CompanyId	Long	Firmennummer
CompanyDisplayName	String	Name der Firma
CompanyRoles	CompanyRoleIndexList	Siehe CompanyRoleIndexList mit CompanyRoleIndex (Rollen, die die Firma besitzt)

CompanyRoleIndexList mit CompanyRoleIndex (Rollen, die die Firma besitzt)

Name	Data type	Description
Id	Integer	ID der Firmenrolle 1 = Hersteller 2 = Lieferant 4 = Kunde

Daten Beispiel JSON

```
{
     "Data": {
          "CreateUserSessionData": {
               "LoginName": "100000",
               "Password": "MyPassword1"
          }
     }
}
```

Daten Beispiel XML

```
<?xml version="1.0"?>
<Aera>
     <Data>
          <CreateUserSessionData>
               <LoginName>100000</LoginName>
               <Password>MyPassword1</Password>
          </CreateUserSessionData>
     </Data>
</Aera>
```

Status der Sitzung überprüfen

Das Abrufen der Sitzung soll als Beispiel für eine Funktion dienen, die eine Anmeldung voraussetzt.

URL	[Server]/[SessionId]
HTTP Method	GET

Parameter

Name	Data type	Required	Description
SessionId	Guid	X	Sitzungsnummer für AERA Webservices

Request data:

Keine Daten benötigt.

Result: UserSessionForInfo

Name	Data type	Description
Id	Guid	Sitzungsnummer
UserId	Integer	Kundennummer
UserName	String	Benutzername
Email	String	E-Mail Adresse des Kontos
UserCompanies	UserCompanyList	Siehe UserCompanyList mit UserCompany (Liste von Firmen auf die der User Zugriff hat)

UserCompanyList mit UserCompany (Liste von Firmen auf die der User Zugriff hat)

Name	Data type	Description
CompanyId	Long	Firmennummer
CompanyDisplayName	String	Name der Firma
CompanyRoles	CompanyRoleIndexList	Siehe CompanyRoleIndexList mit CompanyRoleIndex (Rollen, die die Firma besitzt)

CompanyRoleIndexList mit CompanyRoleIndex (Rollen, die die Firma besitzt)

Name	Data type	Description
Id	Integer	ID der Firmenrolle 1 = Hersteller 2 = Lieferant 4 = Kunde

Abmelden

Das Abmelden macht die Sitzungsnummer ungültig. Nach dem Abmelden kann die Sitzungsnummer nicht mehr verwendet werden.

URL	[Server]/[SessionId]
HTTP Method	DELETE

Parameter

Name	Data type	Required	Description
SessionId	Guid	X	Sitzungsnummer für AERA Webservices

Request data:

Keine Daten benötigt.

Result: ExecutionResult

Name	Data type	Description
Result	Boolean	Gibt an ob die Anfrage erfolgreich war.