



ARIS API

TECHNISCHE EINFÜHRUNG

Version 9.8 – Service Release 6

Oktober 2016

In diesem Dokument finden Sie zusätzliche Informationen zum Zugriff auf ein ARIS-Repository über die ARIS API.

Das Dokument stellt eine Ergänzung zu der offiziellen API-Dokumentation dar, die auf jedem ARIS Connect/Design Server 9.8.x unter <http://<servername:port>/apidocs> zur Verfügung steht.

INHALTSVERZEICHNIS

1	Allgemeine Grundsätze	3
2	Anmeldung	4
3	Informationen über verfügbare Datenbanken anfordern	5
4	Ein Element abrufen	6
5	Kind-Elemente abrufen	7
6	Inhalt eines Modells abrufen	8
7	Ein Element erstellen	9
8	Modellinhalt erstellen	11
9	Modellinhalt löschen	12
10	Attribute	13
10.1	Erstellung	13
10.2	Bestehendes Element aktualisieren oder Element neu anlegen	14
10.3	Ein Attribut löschen	14
11	Zuordnungen	15
12	Modellgrafik	15
13	Elemente in der Datenbank finden	16
14	Seitenformat	18
15	Die Suchergebnisse anordnen	19
16	Generische Abfragen	19

1 Allgemeine Grundsätze

- Die ARIS RESTful APIs wurden für den Zugriff auf das ARIS Repository über Apps auf mobilen Geräten entwickelt. Dies soll den Zugriff auf das ARIS Repository über andere Kanäle, wie z. B. die ARIS Report API, keinesfalls ersetzen. Wie Sie sich vielleicht vorstellen können, ist der Funktionsumfang daher in einigen Bereichen eingeschränkt.
- Jeder Aufruf an die API ist elementar, die Operation ist entweder erfolgreich, oder eben nicht.
- Für jeden Aufruf an die API ist ein gültiges API-Cookie bzw. eine gültige API-Marke der ARIS-Benutzerverwaltung erforderlich.
- Für jeden Aufruf gelten vorgeschriebene und/oder optionale URL-Parameter. Alle Parameterwerte müssen URL-kodiert sein, da sie möglicherweise Sonderzeichen enthalten.
- Werden URL-Parameter weitergegeben, die nicht bekannt oder für eine bestimmte Methode ungültig oder falsch geschrieben sind, werden sie stillschweigend übergangen. Die angeforderte Operation führt dann möglicherweise zu unerwarteten Ergebnissen.
- Viele Aufrufe benötigen auch eine Datenbanksprache als Parameter. Ist diese nicht verfügbar, wird die Ausweichsprache der aktuellen Datenbank verwendet.
- Für viele Operationen ist auch eine GUID für den ARIS-Methodenfilter als Parameter erforderlich. Wird er nicht angegeben, wählt der jeweilige Auswahlmechanismus den richtigen Filter aus. Dies ähnelt der Anmeldung beim Connect-Portal. Wir empfehlen, den erforderlichen Filter als Parameter weiterzugeben, was außerdem deutlich schneller ist.
- Die ausgegebenen Objekte können auch Methodendaten enthalten, wie z. B. `typename`. Methodendaten werden in der Sprache des HTTP-Headers des Kunden zur Verfügung gestellt („Annahmesprache“). Sie können auch den optionalen URL-Parameter `methodlanguage` weitergeben, um die Methodensprache manuell festzulegen.
- Datumsparameter müssen in UTC im RFC 3339 Internetformat YYYY-MM-DD angegeben werden und werden auch in diesem Format ausgegeben. Für Zeitstempel in UTC gilt das Zeitformat RFC 3339 Internet Zulu, also YYYY-MM-DD'T'HH:MM:SS'Z' (2014-01-22T08:22:55Z).

2 Anmeldung

Für alle API-Aufrufe ist eine gültige Marke der ARIS-Benutzerverwaltung (auch als UMC bekannt) erforderlich. Die Marke muss über die dafür vorgesehene Methode von der UMC-API abgerufen werden. Standard-UMC-Marken, die man bei der Anmeldung beim Connect-Portal erhält, sind dafür nicht zulässig.

Die Marke muss in allen folgenden API-Aufrufen als URL-Parameter (**umcsession**) enthalten sein oder als Cookie in die HTTP-Anforderung aufgenommen werden.

POST

[http:// <servername:port>/umc/api/tokens?tenant=default&name=system
&password=manager&key=lsjflskjfsfj](http://<servername:port>/umc/api/tokens?tenant=default&name=system&password=manager&key=lsjflskjfsfj)

Anmerkungen:

Ein häufig auftretendes Problem ist, dass der Programmierer des Kunden vergisst, für alle Parameter eine URL-Codierung vorzunehmen.

Der Parameter **key** ist die Zeichenfolgen-Darstellung eines X.502-Zertifikats. Er muss von der Software AG zum Zertifizieren der App unter folgender Adresse angefordert werden:

<https://www.ariscommunity.com/aris-access-certification-inquiry-form>

3 Informationen über verfügbare Datenbanken anfordern

GET

[http:// <servername:port>/abs/api/databases](http://<servername:port>/abs/api/databases)

Dieser Aufruf gibt eine Liste aller verfügbaren Datenbanken für den aktuellen Benutzer aus. Die gleiche Liste wird für Benutzer beim Öffnen des ARIS Architect angezeigt.

Für jede Datenbank werden deren Name, Hauptgruppen-GUID und die isversioned-Kennzeichnung angezeigt.

Beispiel:

```
{
  "kind": "DATABASE",
  "name": "United Motor Group",
  "isversioned": true,
  "maingroup_guid": "4a713de0-5d02-11e3-0fda-fd81e986d7e2"
}
```

Hinweis: Für diese Operation ist der Lizenztyp „ARIS Access Full“ oder „ARIS Access Read-only“ erforderlich.

GET

[http:// <servername:port>/abs/api/databases/United%20Motor%20Group](http://<servername:port>/abs/api/databases/United%20Motor%20Group)

Dieser Aufruf gibt detailliertere Informationen zu einer Datenbank aus: Hauptgruppen-GUID, isversioned-Kennzeichnung sowie eine Liste aller zulässigen Methodenfilter und Datenbanksprachen. Genau eine Datenbanksprache wird die Kennzeichnung `isalternative = true` haben, sie wird also als Ausweichsprache für die Datenbank dienen, wenn eine Anforderung nicht den URL-Parameter `language` ausgibt.

Bei versionierbaren Datenbanken wird darüber hinaus eine Liste der bestehenden Änderungen ausgegeben.

Beispiel für ein Methodenfilter-Element

```
{
  "kind": "METHODFILTER",
  "guid": "dd838074-ac29-11d4-85b8-00005a4053ff",
  "name": "Entire method",
  "description": "All method content is available."
}
```

Beispiel für ein Datenbanksprachen-Element

```
{
  "kind": "DBLANGUAGE",
  "language": "en_US",
  "isalternative": true
}
```

Beispiel für ein changelist-Element

```
{
  "kind": "CHANGELIST",
  "changelist_number": 1,
  "user": "internal",
  "description": "initial revision after restore from basic archive file.",
  "submit_time": "2015-04-07T16:11:13Z"
}
```

4 Ein Element abrufen

Die ARIS API umfasst die Möglichkeit, Gruppen, Objekte (Definitionen) und Modelle abzurufen. Das gewünschte Element muss durch seine GUID identifiziert werden. Eine Identifizierung über den Gruppenpfad + Name wird nicht unterstützt, da dies möglicherweise zu Verwechslungen führen kann.

Beispiel:

GET

<http://<servername:port>/abs/api/groups/United%20Motor%20Group/1191ae90-02f7-11dc-2729-000bcd0cce4e>

Standardmäßig ist nur der Attributname (AT_NAME) in der Antwort enthalten. Benötigen Sie mehr Attribute, muss der URL-Parameter **attributes** angegeben werden. Mögliche Werte sind

all: alle nicht leeren Attribute

Beispiel:

attributes = 1, AT_DESC, AT_AUTH // Attributname, Beschreibung, Autor

Anmerkung: Kanten werden nicht unterstützt.

Hinweis: Man kann auch HTTP-Portallinks abrufen, die dann direkt in den Browser kopiert werden, um das Element über das CONNECT-Portal zu öffnen. Fügen Sie den URL-Parameter **withportallinks** = **true** ein.

Beispiele für Portallinks (für ein Modell):

```
{
  "kind": "LINK",
  "method": "GET",
  "href": "http://<servername:port>/#default/item/c.process.United Motor
Group.CibrcPlSEdsnKQALzQzOTg",
  "rel": "ITEM_MODEL"           // CONNECT Item View
},

{
  "kind": "LINK",
  "method": "GET",
  "href": "http://<servername:port>/#default/repository/a.model.United Motor
Group.CibrcPlSEdsnKQALzQzOTg",
  "rel": "REPO_MODEL"           // CONNECT Repository View
},

{
  "kind": "LINK",
  "method": "GET",
  "href": "http://<servername:port>/#default/thinclient/c.process.United Motor
Group.CibrcPlSEdsnKQALzQzOTg",
  "rel": "TC_MODEL"             // CONNECT Designer View
},

{
  "kind": "LINK",
  "method": "GET",
  "href": "http://
<servername:port>/abs/downloadClient/aris_database.jsp?configuration=ARIS&apps
erver=myserver&database=United%20Motor%20Group&guid=0a26eb70-fd52-11db-2729-
000bcd0cce4e&language=en_US&tenant=default",
  "rel": "DC_MODEL"             // CONNECT Download client
}
```

5 Kind-Elemente abrufen

Eine Gruppe enthält in der Regel Kinder, d. h. Untergruppen, Modelle und Objekte.

GET

[http:// <servername:port>/abs/api/groups/United%20Motor%20Group/4a713de0-5d02-11e3-0fda-fd81e986d7e2/children](http://<servername:port>/abs/api/groups/United%20Motor%20Group/4a713de0-5d02-11e3-0fda-fd81e986d7e2/children)

Dieser Aufruf gibt alle Untergruppen einer Gruppe für eine bestimmte GUID aus. Wenn Sie außerdem die Modelle und/oder Objekte in der Gruppe benötigen, können Sie die URL-Parameter **withmodels = true** oder **withobjects=true** hinzufügen.

GET

[http:// <servername:port>/abs/api/groups/United%20Motor%20Group/4a713de0-5d02-11e3-0fda-fd81e986d7e2/children?withmodels=true&withobjects=true](http://<servername:port>/abs/api/groups/United%20Motor%20Group/4a713de0-5d02-11e3-0fda-fd81e986d7e2/children?withmodels=true&withobjects=true)

Anmerkung: Sie können auch einen vollständigen Unterbaum abrufen, indem Sie den URL-Parameter **recursive=true** weitergeben. Beachten Sie, dass pro Ergebnis maximal 5000 Elemente angezeigt werden können (unabhängig vom gewählten Seitenformat).

Enthält eine Gruppe mehr als 5000 Kinder, können Sie sie nicht alle abrufen.

6 Inhalt eines Modells abrufen

Der Inhalt eines Modells entspricht seinen Ausprägungen, also den Modellobjekten und -kanten. Sie können sie im Aufruf `getModel` durch Weitergeben des URL-Parameters `withcontent=true` abrufen.

GET

<http://<servername:port>/abs/api/models/United%20Motor%20Group/88ba40a0-cfb6-11e0-2556-5c260a398437?withcontent=true>

Anmerkung: Modellobjekte/Modellkanten sind Daten aus der Ausprägungs- und der Definitionsebene. Sie enthalten keine graphischen Informationen, also keine Maßangaben (Breite, Höhe) und keine Koordinaten für Position und Kantenwege.

Beispiel für ein Modellobjekt-Element

```
{
  "kind": "MODELOBJECT",
  "occid": "(7wLHfY8btgy:u:L+6PmFOGfzQQ4:x:L+33+c)",
  "guid": "d734eb6f-cf14-11e0-2556-5c260a398437",
  "link": {
    "kind": "LINK",
    "method": "GET",
    "href": "http://<servername:port>/abs/api/objects/United Motor
Group/d734eb6f-cf14-11e0-2556-
5c260a398437?language=en_US&methodfilter=dd838074-ac29-11d4-85b8-
00005a4053ff",
    "rel": "OBJECT"
  },
  "type": 239,
  "typename": "Strategy",
  "apiname": "OT_STAT",
  "symbol": 1627,
  "symbolname": "Tactic",
  "symbol_apiname": "ST_TACTIC",
  "attributes": [
    {
      "kind": "ATTRIBUTE",
      "id": "6S4A4i43Hh0:p:L=1=1033:1:s",
      "typename": "Name",
      "type": 1,
      "apiname": "AT_NAME",
      "language": "en_US",
      "value": "Reduce operational costs in Supply Chain"
    }
  ]
}
```


Beispiel für ein Modellkanten-Element

```

{
  "kind": "MODELCONNECTION",
  "occid": "(7wLHfY8btgy:u:L+-6xMo0A_0oEz:y:L+34+c)",
  "type": 67,
  "typename": "encompasses",
  "apiname": "CT_SUBS_1",
  "source_guid": "c7ca78b0-abcf-11e0-7ee8-5c260a398437",
  "target_guid": "289b0560-ac64-11e0-7ee8-5c260a398437",
  "source_link": {
    "kind": "LINK",
    "method": "GET",
    "href": "http://<servername:port>/abs/api/objects/United Motor Group/c7ca78b0-abcf-11e0-7ee8-5c260a398437?language=en_US&methodfilter=dd838074-ac29-11d4-85b8-00005a4053ff",
    "rel": "OBJECT"
  },
  "target_link": {
    "kind": "LINK",
    "method": "GET",
    "href": "http://<servername:port>/abs/api/objects/United Motor Group/289b0560-ac64-11e0-7ee8-5c260a398437?language=en_US&methodfilter=dd838074-ac29-11d4-85b8-00005a4053ff",
    "rel": "OBJECT"
  },
  "source_occid": "(7wLHfY8btgy:u:L+-7cJgU4Td:x:L+33+c)",
  "target_occid": "(7wLHfY8btgy:u:L+-2apRUFiN0WT:x:L+33+c)"
}

```

7 Ein Element erstellen

Die API bietet die Möglichkeit, Gruppen und Objekte zu erstellen. Sie können für Objekte, Gruppen und Modelle Attribute erstellen (oder in ARIS-Terminologie ausgedrückt: sie können Attribute mit einem Wert führen). Sie können innerhalb eines Aufrufs mehrere Attribute hinzufügen --> für sie alle wird die gleiche Datenbanksprache verwendet (aus dem URL-Parameter **language**).

Sie können darüber hinaus eine Zuordnung zwischen einem bestehenden Objekt und einem bestehenden Modell anlegen.

Hinweis: Das Erstellen von Modellen wird nicht unterstützt!

Beispiel:

Erstellen Sie ein neues Objekt des Typs 43 (OT_ORG_UNIT) mit zwei Attributen (AT_NAME und AT_DESC). Datenbanksprache = Englisch; Übergeordnete Gruppe = 4a713de0-5d02-11e3-0fda-fd81e986d7e2 (= Hauptgruppe)

POST[http://](http://<servername:port>/abs/api/objects/United%20Motor%20Group?language=en&parent=4a713de0-5d02-11e3-0fda-fd81e986d7e2)<servername:port>/abs/api/objects/United%20Motor%20Group?language=en&parent=4a713de0-5d02-11e3-0fda-fd81e986d7e2**Hauptteil der Anforderung**

```
{
  "kind": "OBJECT",
  "type": 43,
  "attributes": [
    {
      "kind": "ATTRIBUTE",
      "type": 1,
      "value": "My first OrgUnit "
    },
    {
      "kind": "ATTRIBUTE",
      "type": 9,
      "value": "This is the long description"
    }
  ]
}
```

Anmerkungen :

Die Einträge vom Typ „kind“ im Hauptteil der Anforderung sind redundant, sie werden nicht benötigt.

Beim Erstellen eines Objektes bzw. einer Gruppe mit einem bereits vergebenen Namen gibt es einen Unterschied zu beachten:

- Erstellen Sie eine neue Gruppe mit dem Namen X, und es gibt bereits eine Gruppe mit diesem Namen in der übergeordneten Gruppe, erhält die neue Gruppe automatisch die Bezeichnung X(1).
- Erstellen Sie hingegen ein neues Objekt mit der Bezeichnung X, und es gibt bereits ein Objekt X in der übergeordneten Gruppe, dann trägt das neue Objekt auch die Bezeichnung X. Es gibt also dann zwei Objekte mit dem gleichen Namen innerhalb der Gruppe.

8 Modellinhalt erstellen

Modelle können nicht über die ARIS Mobile API erstellt werden, es ist aber zulässig, Modellobjekte und -kanten anzulegen. Beim Anlegen einer neuen Kante können Objekte verwendet werden, die auf die gleiche Anforderung zugreifen. Da sie noch keine ID haben, müssen Sie eine temporäre Occ-ID vergeben (diese beginnt mit #) und diese beim Definieren von Quelle und Ziel der neuen Kante verwenden.

Beispiel:

PUT

[http:// <servername:port>/abs/api/models/United%20Motor%20Group/42ad5380-3d0e-11e5-6479-22000b630ca4](http://<servername:port>/abs/api/models/United%20Motor%20Group/42ad5380-3d0e-11e5-6479-22000b630ca4)

Hauptteil der Anforderung

```
{
  "modelobjects": [
    {
      "kind": "MODELOBJECT",
      "occid": "#1",
      "type": 22,
      "symbol": 335,
      "attributes": [
        {
          "kind": "ATTRIBUTE",
          "type": 1,
          "value": "new function via model update"
        }
      ]
    },
    {
      "kind": "MODELOBJECT",
      "occid": "#2",
      "type": 18,
      "symbol": 1,
      "attributes": [
        {
          "kind": "ATTRIBUTE",
          "type": 1,
          "value": "new event via model update"
        }
      ]
    }
  ],
  "modelconnections": [
    {
      "kind": "MODELCONNECTION",
      "type": 44,
      "source_occid": "#1",
      "target_occid": "#2"
    }
  ]
}
```

Anmerkungen:

- Beim Anlegen von Objekten wie oben beschrieben, wird eine neue Objektdefinition erstellt. Möchten Sie ein bereits angelegtes Objekt verwenden, müssen Sie dessen GUID angeben:

```
{
  "kind": "MODELOBJECT",
  "occid": "#1",
  "guid" : "4686bd20-3d0e-11e5-6479-22000b630ca4"
  "type" : 22,
  "symbol": 335
}
```

- Da Modellobjekte und -kanten ohne Positionsdaten angelegt werden, werden sie jeweils übereinander in der oberen linken Ecke des Modells angezeigt. Es ist möglich, ein automatisches Layout des Modells zu erzwingen, wenn es erstmalig im ARIS Business Architect Designer geöffnet wird, indem Sie den URL-Parameter `layoutonopen=true` angeben.

Hinweis: Ein Benutzer mit einer Lizenz für ARIS Mobile Access kann innerhalb einer Anforderung nur ein Modellobjekt oder eine Modellkante anlegen.

9 Modellinhalt löschen

Modellobjekte oder -kanten können mithilfe ihrer Occ-IDs aus einem Modell gelöscht werden. Löschen Sie ein Modellobjekt, werden alle davon betroffenen Modellkanten automatisch auch gelöscht. Sie müssen also nicht eigens in die Anforderung aufgenommen werden.

DELETE

[http://<servername:port>/abs/api/models/United%20Motor%20Group/1672a301-3d14-11e5-6479-22000b630ca4/objects?occid=\(-703yxAzzRgN%3Au%3AL%2B-7RaSdNnA6hr%3Ax%3AL%2B33%2Bc\)](http://<servername:port>/abs/api/models/United%20Motor%20Group/1672a301-3d14-11e5-6479-22000b630ca4/objects?occid=(-703yxAzzRgN%3Au%3AL%2B-7RaSdNnA6hr%3Ax%3AL%2B33%2Bc))

Anmerkungen:

- Occ-IDs enthalten oft Sonderzeichen, es ist daher wichtig, dass die korrekten URL-kodierten Werte angegeben werden.
- Achten Sie beim Anforderungspfad darauf, dass er mit „/objects“ oder „/connections“ endet, da ansonsten das Modell selbst gelöscht wird!

Hinweis: Ein Benutzer mit einer Lizenz für ARIS Mobile Access kann innerhalb einer Anforderung nur ein Modellobjekt oder eine Modellkante löschen.

10 Attribute

Attribute sind keine Elemente der oberen Ebene, sie sind eng mit dem jeweils übergeordneten Element, also Objekt, Gruppe oder Modell, verbunden. Sie können Attribute nur zusammen mit dem jeweils zugehörigen, übergeordneten Element abrufen.

Attribute können über eine Ganzzahl (z. B. 1), den API-Namen (z. B. AT_NAME) oder die GUID identifiziert werden. Innerhalb einer Anforderung können verschiedene Typen-Identifizierer verwendet werden, wie z. B. der URL-Parameter **attributes**.

Attribute werden in der Datenbanksprache verwaltet, die vom URL-Parameter **language** vorgegeben wird. Ist dieser nicht verfügbar, wird die Standardsprache der aktuellen Datenbank verwendet.

Anmerkungen:

- Aktuell wird nur eine kleine Untergruppe von Attributtypen unterstützt: Ganzzahl, Fließkommazahl, Text, Zeitstempel, Zeit, Wert, Attribute und Binär.
- Binärdaten müssen als BASE64-codierte Zeichenfolge geschickt werden.

10.1 Erstellung

Sie können ein Attribut zusammen mit dem übergeordneten Element im Rahmen ein und derselben Anforderung erstellen.

Beispiel: Erstellen Sie ein Objekt mit den zwei Attributen „name“ und „description“.

POST

[http:// <servername:port>/abs/api/objects/United%20Motor%20Group?language=en
&parent=4a713de0-5d02-11e3-0fda-fd81e986d7e2](http://<servername:port>/abs/api/objects/United%20Motor%20Group?language=en&parent=4a713de0-5d02-11e3-0fda-fd81e986d7e2)

Hauptteil der Anforderung

```
{
"kind": "OBJECT",
"type": 43,
"attributes": [
  {
    "kind": "ATTRIBUTE",
    "type": "AT_NAME",
    "value": "I am an OrgUnit!"
  },
  {
    "kind": "ATTRIBUTE",
    "type": 9,
    "value": "This is the long description"
  }
]
}
```

10.2 Bestehendes Element aktualisieren oder Element neu anlegen

Möchten Sie einem bestehenden Element ein neues Attribut hinzufügen oder dessen Wert aktualisieren, müssen Sie eine Aktualisierung (PUT) für das Element durchführen.

Beispiel: Aktualisieren Sie den Namen eines bestehenden Objekts und fügen Sie das Attribut „author“ hinzu.

PUT

[http:// <servername:port>/abs/api/objects/United%20Motor%20Group/b7e90c56-00cf-11e2-21d1-5c260a628455?language=en](http://<servername:port>/abs/api/objects/United%20Motor%20Group/b7e90c56-00cf-11e2-21d1-5c260a628455?language=en)

Hauptteil der Anforderung

```
{
  "attributes": [
    {
      "kind": "ATTRIBUTE",
      "type": 1,
      "value": "Just renamed it"
    },
    {
      "kind": "ATTRIBUTE",
      "type": "AT_AUTH",
      "value": "I am the author"
    }
  ]
}
```

10.3 Ein Attribut löschen

Das Löschen eines Attributs kann nicht über die Aktualisierung des jeweils übergeordneten Elements geschehen. Sie müssen dafür die Operation DELETE einsetzen. Die Typennummern der zu löschenden Attribute müssen über den URL-Parameter **typennumbers** angegeben werden.

DELETE

[http:// <servername:port>/abs/api/objects/United%20Motor%20Group/b7e90c56-00cf-11e2-21d1-5c260a628455/attributes?language=en&typennumbers=AT_DESC, AT_AUTH](http://<servername:port>/abs/api/objects/United%20Motor%20Group/b7e90c56-00cf-11e2-21d1-5c260a628455/attributes?language=en&typennumbers=AT_DESC, AT_AUTH)

Anmerkungen:

- Der URL-Pfad muss auf „/attributes“ enden, da Sie sonst das übergeordnete Objekt UNWIEDERBRINGLICH löschen!
- Das Ergebnis einer erfolgreichen Löschoperation ist leer und hat den STATUS = OK.

```
{
  "kind": "RESULT",
  "request": "abs#deleteObjectAttributes",
  "status": "OK",
  "item_count": 0,
  "items": []
}
```

Hinweis: Benutzer mit Zugang über ARIS Mobile Access können innerhalb einer Anforderung nur jeweils ein Attribut löschen.

11 Zuordnungen

Eine Zuordnung stellt lediglich einen Link zwischen einem bestehenden Objekt und einem bestehenden Modell dar. Mit der ARIS Repository API können Sie Zuordnungen erstellen und löschen. Es ist nicht möglich, eine bestimmte Zuordnung abzurufen, sie werden automatisch beim Abrufen eines Objekts mit abgerufen.

Beispiel: Erstellen Sie eine Zuordnung.

POST

[http:// <servername:port>/abs/api/objects/United Motor Group/6aa1b167-fac0-11de-55c7-001a6b3c820f/assignment/a39aefb0-falc-11db-2729-000bcd0cce4e](http://<servername:port>/abs/api/objects/United%20Motor%20Group/6aa1b167-fac0-11de-55c7-001a6b3c820f/assignment/a39aefb0-falc-11db-2729-000bcd0cce4e/objectGUID)
objectGUID (Ursprung der Zuordnung)
modelGUID (Ziel der Zuordnung)

Beispiel: Löschen Sie eine Zuordnung.

DELETE

[http:// <servername:port>/abs/api/objects/United Motor Group/6aa1b167-fac0-11de-55c7-001a6b3c820f/assignment/a39aefb0-falc-11db-2729-000bcd0cce4e](http://<servername:port>/abs/api/objects/United%20Motor%20Group/6aa1b167-fac0-11de-55c7-001a6b3c820f/assignment/a39aefb0-falc-11db-2729-000bcd0cce4e)

12 Modellgrafik

Die ARIS API enthält keine Positionsdaten beim Abrufen eines Modells mit seinem Inhalt. Es ist jedoch möglich, eine Modellgrafik abzurufen. Sie hat das Format PNG (Portable Network Graphics) und wird als BASE64-codierte Zeichenfolge ausgegeben. Mithilfe der beiden optionalen URL-Parameter `maxwidth` und `maxheight` können Sie die Größe der erstellten Grafik (in Pixeln) festlegen.

Beispiel:

GET

<http://<servername:port>/abs/api/models/United%20Motor%20Group/0a26eb70-fd52-11db-2729-000bcd0cce4e/graphic>

13 Elemente in der Datenbank finden

Sie können basierend auf mehreren Kriterien, die durch AND verbunden sind, nach Elementen suchen. Die Liste der Suchergebnisse lässt sich so einschränken. Sie müssen den URL-Parameter KIND (Typ) angeben, um festzulegen, ob nach einem Modell oder nach einem Objekt gesucht werden soll. Sie können darüber hinaus den gewünschten Typ (eine kommaseparierte Liste von Objekttyp- oder Modelltypnummern, API-Namen oder Typ-GUIDs werden auch unterstützt) und ein Filterkriterium zu einem Attribut definieren.

Beispiel: Suchen Sie alle Modelle vom Typ 13 (MT_EEPC), deren Namen mit Sale (mit großem und kleinem „S“) beginnt.

GET

http://<servername:port>/abs/api/databases/United%20Motor%20Group/find?kind=MODEL&typefilter=13&language=en&attrfilter=AT_NAME=Sale

Anmerkungen:

- Verfügbare Elementtypen: MODEL, OBJECT, GROUP
- Für alle Attribute: + (isMaintained), - (isNotMaintained)
- Für Zahlen sind die herkömmlichen Operatoren verfügbar: =, !=, >, <, >=, <=
- Für Boolesche Werte: = und !=
- Für Textattribute: =
Hinweis: nicht gleich (!=) wird nicht unterstützt, da dadurch möglicherweise sehr umfangreiche Scans im zugrundeliegenden Datenbankindex ausgelöst werden.
- Der Parameter **attrfilter** kann vollständig von Klammern umgebene Ausdrücke enthalten (Klammer = { }) sowie die Operatoren AND und OR.

Suchen nach Text: Beispiel für den Wert attrfilter:

```
AT_NAME = Jones
9       = Human Resources      // (9 = AT_DESC)
AT_AUTH -                               // (AT_AUTH not maintained)
{AT_NAME = Jones} AND {AT_DESC = Human Resources}
{AT_NAME = Jones} OR {AT_DESC = Human Resources} AND {AT_AUTH -}
```

- **Wichtig:** In Klammern stehende Ausdrücke dürfen nicht die Werte { oder } enthalten. Muss der Suchwert geschwungene Klammern enthalten, geben Sie bei der Suche nach Textattributen „\}“ ein.
- Das normale Suchverhalten läuft wie bei herkömmlichen Suchmaschinen ab: Die Inhalte von Textattributen werden anhand von Textmarken gescannt, Trennzeichen wie Leerzeichen oder Bindestriche (-) werden ignoriert. Sind Textmarken aus dem Suchwert im Textattribut enthalten (unabhängig von Reihenfolge/Position), wird das als Übereinstimmung gewertet.

Beispiel: Suchausdruck: AT_NAME = holder certificate

findet „Holder Certificate“ und „certificate holder“ sowie „the certificate holder“.

- URL-Parameter **matchcase**: wenn „true“, unterscheidet die Suche zwischen Groß-/Kleinschreibung und nach Reihenfolge der Textmarken.
- URL-Parameter **exactsearch**: wenn „true“, werden nur exakte Übereinstimmungen ausgegeben. Sonderzeichen und der Platzhalter (*) werden als einfaches Zeichen bewertet, darüber hinaus spielt die Reihenfolge der Textmarken eine Rolle.

Beispiel: Suchausdruck: AT_NAME = holder certificate

findet „Holder Certificate“, aber nicht „certificate holder“ oder „The Holder Certificate“

-
- Sondersuchmuster für Text: (Parameter `exactsearch` muss "false" lauten).
A* gibt alle Zeichenfolgen wieder, die mit „A“ beginnen (A muss eine Zeichenkette der Länge ≥ 2 sein, die ersten beiden Zeichen müssen Buchstaben oder Ziffern sein.).
A*B gibt alle Zeichenfolgen aus, die mit „A“ beginnen (A muss eine Zeichenkette der Länge ≥ 2 sein, die ersten beiden Zeichen müssen Buchstaben oder Ziffern sein) und auf „B“ enden.

14 Seitenformat

Beim Einsatz der Suchfunktion in der Datenbank werden Elemente in der Regel an den aufrufenden Benutzer zurückgemeldet. Standardmäßig werden pro Seite 100 Einträge angezeigt. Eine davon abweichende Seitenabmessung kann über den URL-Parameter **pagesize** angegeben werden.

Ergibt die Suche mehr Ergebnisse, als auf der Seite angezeigt werden können, enthält die Antwort den Eintrag **next_pagetoken**. Der Kunde weiß so, dass noch weitere Daten geladen werden.

Beispiel:

Anforderung:

http://<servername:port>/abs/api/databases/United%20Motor%20Group/find?kind=MODEL&language=en&attrfilter=AT_NAME&wildcards=true&attrcriterion==S*

Antwort:

```
{
  "kind": "RESULT",
  "request": "abs#find",
  "status": "OK",
  "item_count": "100",
  "next_pagetoken": "1:100:guid",
  "items": [
    ... 100 items here
  ]
}
```

Um zur nächsten Ergebnisseite weiterzublättern, muss der Kunde die gleiche Anforderung erneut schicken und die Marke „page“ angeben:

http://<servername:port>/abs/api/databases/United%20Motor%20Group/find?kind=MODEL&language=en&attrfilter=AT_NAME&wildcards=true&attrcriterion==S*&pagetoken=1:100:guid

Antwort:

```
{
  "kind": "RESULT",
  "request": "abs#find",
  "status": "OK",
  "item_count": "35",
  "items": [
    ... remaining 35 items here
  ]
}
```

Die zweite Antwort enthält nicht den Eintrag **next_pagetoken** --> der Kunde weiß, dass keine weiteren Elemente geladen werden müssen.

Hinweis: Die maximal zulässige Seitengröße beträgt 100 Einträge für ARIS Mobile Access-Benutzer und 500 Einträge für ARIS Access Full/ARIS Access Read-Only. Wird eine zu große Seitengröße angegeben, korrigiert das System sie auf das zulässige Maximum.

15 Die Suchergebnisse anordnen

Ergibt eine Anforderung mehr als ein Element, beispielsweise nach einer Datenbanksuche, werden diese nach ihrer GUID in aufsteigender Reihenfolge angeordnet. Das Anordnen nach GUID ist in der Regel hilfreich und garantiert stabile Ergebnisse, dies besonders hinsichtlich der Seitengröße.

Sie können mithilfe des URL-Parameters **orderby** eine andere Sortierreihenfolge angeben:

```
...&orderby = name    (anordnen nach Attribut AT_NAME, aufsteigend)
...&orderby = name    (anordnen nach Attribut AT_NAME, absteigend)
```

Andere mögliche Werte sind **modified** (aufsteigend), **-modified** (absteigend), **path** (Gruppenpfad, beginnend bei Hauptgruppe).

16 Generische Abfragen

Generische Abfragen stellen einen Versuch dar, Elemente in Diagrammform abzurufen. Syntax und Funktionalität basieren auf der CONNECT Portal Designer Abfragesprache, die für Datensteckbriefe verwendet wird.

Eine Abfrage kann bei einem definierten Element (also einem, über die GUID oder die Typennummer festgelegten Objekt oder Modell) beginnen. Sie können auch weitere Elemente hinzufügen, die über das Startelement oder eine Folge von Filterelementen erreicht werden können.

Die allgemeinen Eigenschaften einer Elementdefinition lauten folgendermaßen:

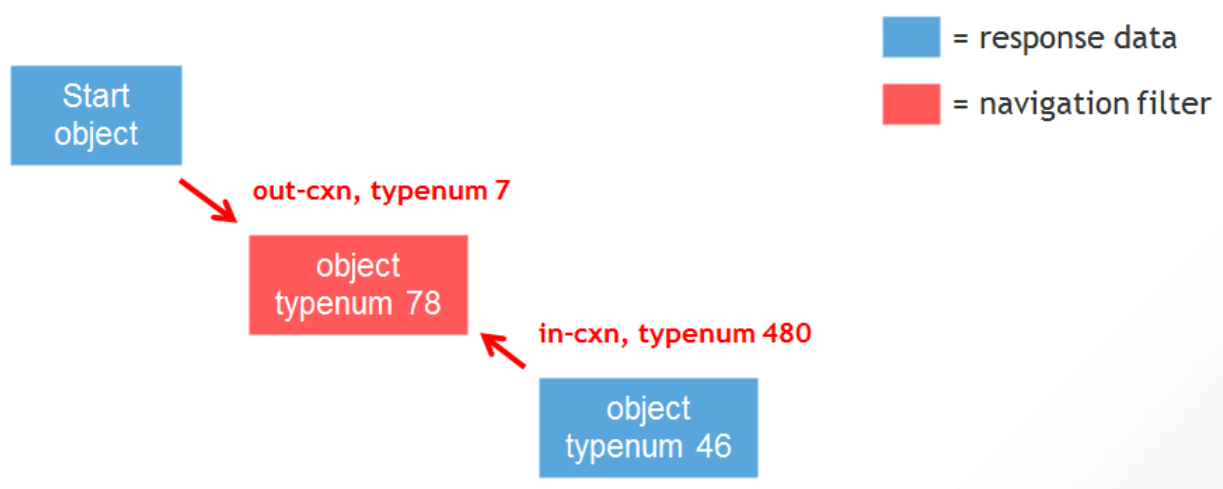
type : MODEL, OBJECT, CONNECTION, ASSIGNMENT, OCCURRENCE, CONNECTIONOCCURRENCE

function : FILTER, TARGET (TARGET bedeutet, dass Objekte, die mit diesem Element übereinstimmen, in das Ergebnis aufgenommen werden.)

(FILTER bedeutet, dass dieses Element nur als Filter dient. Dies ist der Standardwert und daher optional.)

direction: IN, OUT, BOTH (nur zulässig für den Typ CONNECTION)

typenum: Kommaseparierte Liste von Typennummern, API-Namen, Typ-GUIDs)



Anmerkungen:

- Nicht alle möglichen Kombinationen ergeben immer sinnvolle Ergebnisse! Die Syntax verspricht mehr als das, was implementiert wird.
- Elementdefinitionen enthalten in der Regel andere Elementdefinitionen und bilden daher einen verschachtelten Pfad von Elementen. Innerhalb eines Pfads ist nur ein Element der Funktion TARGET zulässig.
- Obwohl eine generische Abfrage schreibgeschützt ist, muss sie als POST-Anforderung geschickt werden, da die Abfragedefinition im Hauptteil der Anforderung enthalten sein muss.
- Ist das Startelement ein Objekt, muss die Anforderungs-URL folgendermaßen lauten:
.../objects/<db>/query; Bei Modellen ist es .../models/<db>/query

Der Hauptteil der Anforderung muss die Abfrage selbst als verschachtelte Liste von Elementdefinitionen der Abfrage enthalten.

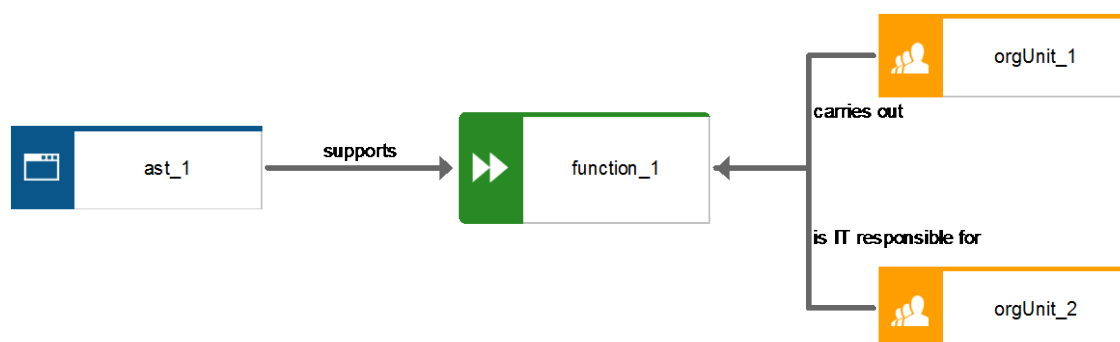
Beispiel 1 (Beginn bei Objekt):

POST:

[http:// <servername:port>/abs/api/objects/United%20Motor%20Group/query](http://<servername:port>/abs/api/objects/United%20Motor%20Group/query)

Hauptteil der Anforderung:

```
{
  "start_guids" : "a387cc60-3c34-11e5-216f-782bcb6367e0",
  "items" : [
    { "type" : "CONNECTION",
      "direction" : "IN",
      "items" : [
        {
          "type" : "OBJECT",
          "typenum" : "6,43",
          "function" : "TARGET"
        }
      ]
    }
  ]
}
```



Ergebnis: Startfunktion (function_1), ast_1 (type 6) , orgUnit_1 (type 43) , orgUnit_2 (type 43)

Beispiel 2 (Beginn bei Modell):

POST:

<http://<servername:port>/abs/api/models/United%20Motor%20Group/query>

Hauptteil der Anforderung:

```
{
  "start_guids" : "0a26eb70-fd52-11db-2729-000bcd0cce4e",
  "items" : [
    {
      "type" : "OCCURRENCE",
      "items" : [
        {
          "type" : "OBJECT",
          "typenum" : "22",
          "function" : "FILTER"
        },
        {
          "type" : "CONNECTIONOCCURRENCE",
          "items" : [
            {
              "type" : "CONNECTION",
              "function" : "FILTER",
              "direction" : "IN",
              "items" : [
                {
                  "type" : "OBJECT",
                  "typenum" : "43,6",
                  "function" : "TARGET"
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```

Ergebnis: Startmodell, Funktion "Configure vehicle" (type 22), Anwendungssystemtyp COT (type 6)**ÜBER SOFTWARE AG**

Die Software AG bietet die erste Digital Business Plattform an. Das Unternehmen hat nach Meinung führender Marktanalysten eine führende Rolle in diesem Bereich und unterstützt Sie dabei, bestehende Systeme vor Ort und in der Cloud zu einer einzigen Plattform zu vereinen, um Ihre Geschäftsabläufe zu optimieren und die Zufriedenheit Ihrer Kunden zu steigern. Mit der Software AG können Sie rasch Digitale Business Anwendungen erstellen und umsetzen, um Marktchancen zeitnah zu nutzen. Holen Sie das Maximum aus Big Data heraus, optimieren Sie Ihre Entscheidungsprozesse, erreichen Sie mehr mit dem „Internet der Dinge“ und stimmen Sie Ihre Prozesse dank intelligenter Governance, Risikomanagement und Compliance rascher auf gesetzliche Änderungen und Herausforderungen ab. Führende Unternehmen weltweit vertrauen der Software AG, wenn es um Innovation, Wettbewerbsfähigkeit und ihre Stellung auf dem Markt in der digitalen Welt geht. Erfahren Sie mehr unter SoftwareAG.com.

© 2016 Software AG. Alle Rechte vorbehalten. Software AG und alle Software AG-Produkte sind Warenzeichen oder registrierte Warenzeichen der Software AG. Namen anderer Produkte oder anderer Unternehmen können Marken ihrer jeweiligen Schutzrechtsinhaber sein.

