DB Dienste zur Unterstützung von Sensordiensten [DBAP8]

15.12.2014

Henning Budde

Andreas Lockermann

Thomas Partsch

Gefördert durch:

Bundesministerium für Wirtschaft und Technologie

aufgrund eines Beschlusses des Deutschen Bundestages

1. Problemstellung

Der Einsatz von Datenbanksystemen mit relationalen und nicht-relationalen Ansätzen bedingt unterschiedliche Zugriffsmöglichkeiten auf die gespeicherten Nutzdaten. So differieren die Datenbanksprachen und die Datenhaltung (normalisiert oder denormalisiert) je nach dem Datenmodell des eingesetzten Datenbanksystems (z. B. relationales Datenmodell vs. Datenmodell eines Wide Column Stores). Dies führt für die Sensordienste zu einer notwendigen Kenntnis über die eingesetzten Systeme und deren Datenhaltung. Zudem sind Änderungen an den eingesetzten Datenbanksystemen und implementierten Datenbankschemata nicht ohne gleichzeitige Änderung der Sensordienste umsetzbar.

2. Ziele

Der Zugriff auf die Datenbanken muss daher für Sensordienste über eine Middleware abstrahiert werden. Diese stellt geeignete Dienste für den Zugriff auf die Entitäten des konzeptionellen globalen SensorCloud-Datenbankschemas [1] bereit, vereinfacht die Sicht auf die Datenbanken und ermöglicht Änderungen an den eingesetzten Datenbanken und den implementierten Datenbankschemata ohne die Notwendigkeit von Änderungen an den Sensordiensten. Dies kann dadurch erreicht werden, dass sich vorhandene Schnittstellen für Sensordienste nicht verändern und neue Dienste über neue Schnittstellen bereitgestellt werden. Sind Änderungen an vorhandenen Schnittstellen unabdingbar, müssen die alten Schnittstellen weiterhin nutzbar bleiben (z. B. durch Funktions-Überladung).

3. Überlegungen

Als Middleware für die Datenbanken der SensorCloud SensorCloud eignen sich RESTful-Services. Bei RESTful-Services handelt es sich um ein Programmierparadigma, bei dem Services über URIs adressiert werden und welches auf das Hypertext Transfer Protocol (HTTP) aufsetzt. Dadurch erben die RESTful-Services die Eigenschaften von HTTP und die Anforderungen, die an das Protokoll gestellt werden. So müssen RESTful-Services als idempotente Services implementiert werden, um bei einer unbeabsichtigten Hintereinanderausführung eines Services keine ungewollten Nebeneffekte zu erzielen. Zudem muss ein RESTful-Service für einen Benutzer eine eindeutig zu erwartende Funktionalität bieten. [2] Da alle benötigten Informationen für die Benutzung des Service mit dem Aufruf übermittelt werden, ist jeder Service eine abgeschlossene Transaktion. Damit liegt serverseitig ein zustandsloser Automat vor. Besteht eine Verarbeitung aus dem Aufruf einer festgelegten Sequenz von Services, so ist der Zustandsautomat auf Client-Seite zu implementieren. [3] Die Eigenschaft des zustandslosen Automaten unterstützt die horizontale Skalierung von RESTful-Services, wie sie in dem Projekt SensorCloud aufgrund der Big-Data Problematik gefordert ist und impliziert gleichzeitig die Abgeschlossenheit aller notwendigen Datenbanktransaktionen des aufgerufenen Services. Letzteres sichert die Konsistenz der Datenbank nach einem RESTful-Service-Aufruf.

Da RESTful-Services auf das HTTP-Protokoll aufsetzen, sind sie durch Firewalls erreichbar, sofern keine Sperrung von HTTP-Traffic eingerichtet ist. Zudem sind alle bei HTTP einsetzbaren Sicherheitsmechanismen nutzbar. Dazu zählen die Verschlüsselung und Authentifizierung über SSL/TLS.

Das vorliegende Dokument beschreibt die Services der entwickelten Middleware. Die Services lassen sich in den Database Abstraction Layer als generische Zugriffschicht auf die Entitäten des

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Föderierten Datenbanksystems (Kapitel 5), den Webservice zum Zugriff auf Messwertdaten (Kapitel 7) und den weiteren Webservices, die nicht durch die ersten beiden Zugriffsmöglichkeiten abgebildet sind (Kapitel 8), unterteilen.

4. Trusted Datenbankdienste

Ein wesentliches Mittel der Authentifizierung bei der Anmeldung an die DB-Dienste der Cloud sind Zertifikate, die für jeden LocationMaster individuell ausgestellt und in dessen Software hinterlegt sind (LocationMaster-Zertifikat). Ebenso können Services der SensorCloud-Serviceplattform [7] die DAL-Dienste der Cloud über deren individuellen Zertifikate verwenden (Service-Zertifikat). Zur Kontrolle der Authentizität des DALs verwenden dessen Dienste ebenso Zertifikate, die im Folgenden Serverzertifikat genannt werden.

Abbildung 1 zeigt die Authentifizierung an den Trusted Datenbankdiensten der SensorCloud und die gesicherte Kommunikation mit den Diensten des Database Abstraction Layers (DAL).

LocationMaster und Services der SensorCloud-Serviceplattform verbinden sich zum DAL und prüfen über dessen Serverzertifikat die Authentizität des Endpunktes (1). Nach dem Aufbau eines über Transport Layer Security (TLS) gesicherten Kommunikationskanals authentifizieren sich LocationMaster und Services anhand ihrer Zertifikate an dem DAL (2). Der DAL prüft das Zertifikat der Gegenstelle und die Zugriffsrechte (zur Verwaltung der Zugriffsrechte von Nutzern und Gruppen vgl. [8]) auf den aufgerufenen Dienst des DALs. Im Erfolgsfall erhält die Gegenstelle Zugriff auf den Dienst (3). Nur authentifizierte LocationMaster und Services können über den mittels TLS gesicherten Kommunikationskanal die für sie zugelassenen Dienste des DALs verwenden (4). Die Dienste des DALs prüfen über das verwendete Zertifikat (die Identität der Gegenstelle) die Zugriffsrechte auf die angeforderten Daten. Dadurch wird ein Zugriff auf Daten fremder LocationMaster und Services unterbunden. Die Gegenstelle erhält eine entsprechende Mitteilung (5). Jeder Dienst, der mit dem föderierten Datenbanksystem der SensorCloud kommuniziert (6), authentifiziert sich an den Datenbanken des föderierten Systems mit den Benutzerkennungen für die über die Benutzerverwaltung der Datenbankmanagementsysteme der Zugriff auf die Entitäten der Datenbanken pro Dienst geregelt wird (7). Hiermit wird zum Beispiel sichergestellt, dass ein Dienst zum Ändern von Benutzerstammdaten nur die Benutzer-Entitäten verändern, aber nicht neue Sensoren anlegen kann. Die Zugriffsrechte eines Datenbankbenutzers können hierbei je nach DBMS auf erlaubte Entitäten, auf erlaubte Attribute einer Entität und die Art des Zugriffes (SELECT, INSERT, UPDATE, DELETE) differenziert werden.

Gefördert durch:

Bundesministerium
für Wirtschaft
und Technologie

aufgrund eines Beschlusses
des Deutschen Bundestages

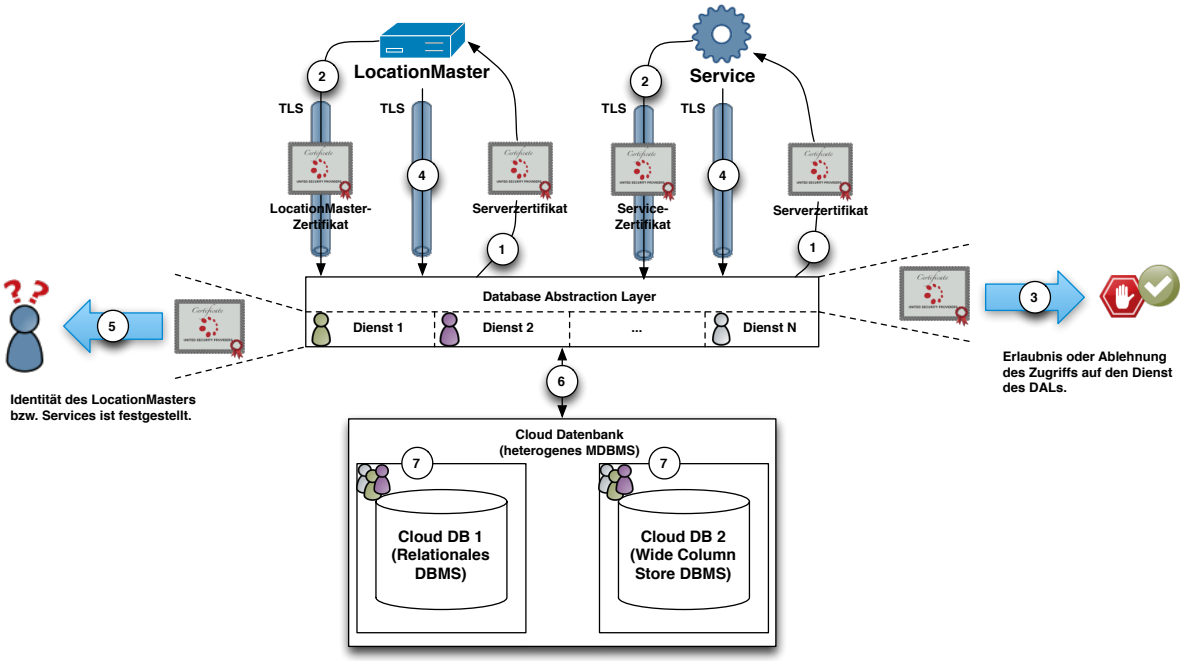

Abbildung 1: Trusted Datenbankdienste

5. Database Abstraction Layer (DAL)

Der Database Abstraction Layer ist im Projekt SensorCloud über RESTful-Services realisiert. Dies unterstützt neben der oben dargestellten horizontalen Skalierung, die Konsistenz der gespeicherten Nutzdaten. Dieser Konsistenzmechanismus ist erforderlich, da im Unterschied zu einem RDBMS verfügt ein Wide Column Store DBMS über keinen inhärenten Transaktionsmechanismus verfügt. Dieser muss über geeignete Datenbankdienste, hier sind es RESTful-Services, bereitgestellt werden. Transaktionsmechanismen sind für Datenbankknoten einer TrustedCloud unerlässlich [5].

Dieses Kapitel beschreibt den generischen DAL, aus einer nicht statischen API besteht. Dies vereinfacht die Pflege des DALs und ermöglicht eine einheitliche API für alle Entitäten, die leicht aus der in diesem Dokument beschriebenen Bildungsvorschrift hergeleitet werden kann.

Technische Beschreibung

Die RESTful-API des DALs wird zur Laufzeit über das JSON globale Schema generiert. Die API dient als Ersatz für lesende, schreibende, ändernde und löschende Zugriffe auf die SensorCloud-Datenbanken mit SQL-Statements aus der Data Manipulation Language (DML). Hierdurch kann für das föderierte Datenbanksystem der SensorCloud sichergestellt werden, dass - im Sinne eines synchronen Datenbestandes zwischen den entfernten LocationMaster-Datenbanken und der zentralen SensorCloud-Datenbank - bei Änderungen an dem Cloud-Datenbestand die betroffenen LocationMaster über eine Tiefensuche identifiziert und an diese Änderungsnachrichten für deren lokalen Datenbestand verschickt werden können.

Ressourcen werden bei dem als RESTful-Service implementierten generischen DAL über die URI-Schreibweise identifiziert (siehe Listing 1). Eine Ressource beschreibt ein oder mehrere Tupel einer Entität (ein oder mehrere Datensätze einer Tabelle). Die HTTP-Request-Methoden unterscheiden analog der verschiedenen DML-Schlüsselwörter die Art des Zugriffes auf eine Ressource. Der DAL nutzt für lesende Zugriffe *GET*- (analog dem SELECT-Statement in DML), für löschende Zugriffe *DELETE*- (analog dem DELETE-Statement in DML), für die Anlage oder Änderung neuer Ressourcen

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

PUT- (analog dem INSERT- und dem UPDATE-Statement in DML) und für die Verknüpfung von Ressourcen über deren Fremdschlüssel *POST-Requests*.

```
<Ressource>     ::=     "http://babeauf.nt.fh-koeln.de/DAL/"     <ENTITÄT>
     <IDENTIFIER>

<IDENTIFIER>  ::=  "/"  <PRIK>  |  "/"  <FKEYColumn>  "/"  <FKEY>  |  "/"
     <UNIQUEColumn> "/" <UKEY> |

<ENTITÄT> ::= <Entität aus dem konzeptionellen Schema>

<PRIK> ::= <Primary Key>

<FKEYColumn> ::= <Foreign Key Attribut aus dem konzeptionellen Schema >

<FKEY> ::= <Fremdschlüssel>

<UNIQUEColumn> ::= <Attribut aus dem konzeptionellen Schema mit UNIQUE
     Index>

<UKEY> ::= <Attributwert der UNIQUEColumn>
```

Listing 1: Ressource in Backus-Naur-Form

Returncodes als HTTP-Status-Code

Der DAL - als RESTful-Service - verwendet HTTP-Status-Codes als Rückgabewert der einzelnen Services. Die HTTP-Status-Codes sind nach jedem RESTful-Service-Aufruf auszuwerten, da diese Aufschluss über den Erfolg des vorangegangenen Aufrufes geben.

Die Gesamtheit der DAL-Services nutzt die im Folgenden beschriebenen Returncodes.

200 OK

Die Anfrage wurde ausgeführt. Die Rückgabe im HTTP-Response-Body unterscheidet sich je nach HTTP-Request-Methode:

GET: Der Response-Body enthält die angeforderte Ressource.

POST: Der Response-Body enthält eine Beschreibung des Ergebnisses der durchgeführten Aktion.

201 Created

Die Anfrage wurde ausgeführt und eine neue Ressource wurde angelegt. Die angelegte Ressource wird über den Location-Eintrag im Response-Header referenziert (siehe Listing 2). Der Response-Body enthält zudem die neue Ressource in dem unter Content-Type angegebenen Format.

```
HTTP/1.1 201 Created
Date: Wed, 20 Aug 2014 08:24:53 GMT
Server: Apache/2.2.22 (Debian)
X-Powered-By: PHP/5.4.30-1~dotdeb.1
Location: http://babeauf.nt.fh-koeln.de/DAL/NutzerTelefon/c06187d0-bcb1-
4cdc-affe-f03fe949e3f4
Vary: Accept-Encoding
Content-Length: 4711
Content-Type: application/json
```

Listing 2: Response Header für 201 Created

202 Accepted

Die Anfrage wurde angenommen, aber noch nicht abgearbeitet. Dieser Statuscode wird für asynchrone Ausführung genutzt und gibt keine Rückmeldung über den Erfolg der

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

durchgeführten Operation.

204 No Content

Die Anfrage wurde ausgeführt. Der Response-Body ist leer.

400 Bad Request

Die Anfrage wurde vom Server aufgrund einer fehlerhaften Syntax nicht verstanden. Der Client muss seine Anfrage vor Wiederholung korrigieren.

404 Not Found

Die Ressource konnte nicht gefunden werden.

500 Internal Server Error

Es ist serverseitig ein Fehler aufgetreten, der nicht durch den Client behoben werden kann.

Rückgaben im HTTP-Response-Body

Treten bei der Verarbeitung Fehler auf (HTTP-Status-Code >= 400), wird im HTTP-Response-Body ein JSON-Objekt für eine weitere Fehleranalyse zurückgeben (Listing 3). Das JSON-Objekt liefert detaillierte Informationen über den Fehler.

```
<Rückgabe> ::= '{   "status" : "error" , "msg" : <FEHLERTEXT>, "data" :
    <ZUSATZINFORMATIONEN> }'

<FEHLERTEXT> ::= <Fehlertext im JSON-Datentyp Zeichenkette>

<ZUSATZINFORMATIONEN>   ::=   <TEXTZEILE>   ","   <ZUSATZINFORMATIONEN>   |
    <TEXTZEILE>

<TEXTZEILE> ::= <Zeichenkette mit zusätzlichen Informationen>
```

Listing 3: JSON-Objekt mit Fehlerinformationen

HTTP-Request-Methods

Die HTTP-Request-Methode bestimmt die Art des Zugriffes auf eine Ressource. Der DAL unterscheidet zwischen lesenden (GET) und verändernden Zugriffen (PUT, POST, DELETE).

GET

Der Zugriff auf eine Ressource über einen GET-Request kann wahlweise über die als Primary Key, die als Foreign Key und die als UNIQUE definierten Attribute einer Entität erfolgen (Listing 4). Der Zugriff über den Primary Key erfolgt ohne Angabe des Primary Key Attributes. Dieses wird automatisch durch den DAL ermittelt. Bei Zugriffen über Foreign Keys oder über die als UNIQUE definierten Attribute muss der URI der Attributname hinzugefügt werden.

```
<Ressource>     ::=     "http://babeauf.nt.fh-koeln.de/DAL/"     <ENTITÄT>
    <IDENTIFIER>

<IDENTIFIER>  ::=  "/"  <PRIK>  |  "/"  <FKEYColumn>  "/"  <FKEY>  |  "/"
    <UNIQUEColumn> "/" <UKEY>

<ENTITÄT> ::= <Entität aus dem konzeptionellen Schema>

<PRIK> ::= <Primary Key>

<FKEYColumn> ::= <Foreign Key Attribut aus dem konzeptionellen Schema >

<FKEY> ::= <Fremdschlüssel>

<UNIQUEColumn> ::= <Attribut aus dem konzeptionellen Schema mit UNIQUE
```

Gefördert durch:

Bundesministerium für Wirtschaft und Technologie

aufgrund eines Beschlusses des Deutschen Bundestages

```
    Index>
<UKEY> ::= <Attributwert der UNIQUEColumn>
```

Listing 4: Ressource in Backus-Naur-Form für GET-Request

Die Rückgabe eines GET-Requests erfolgt als JSON-Objekt im Body der HTTP-Response. Das JSON-Objekt beschreibt bei Zugriffen über den Primary Key oder über UNIQUE-Attribute genau eine Ressource und hat dabei den in Listing 5 beschriebenen Aufbau.

```
<Rückgabe> ::= "{" <ENTITÄT> ":" "{" <KEYVALUE> "}" "}"

<ENTITÄT> ::= <Entität aus dem konzeptionellen Schema>

<KEYVALUE> ::= <ATTRIBUT> ":" <ATTRIBUTWERT> "," <KEYVALUE> | <ATTRIBUT>
    ":" <ATTRIBUTWERT>

<ATTRIBUT> ::= <Attributname aus dem konzeptionellen Schema als JSON-
    Datentyp Zeichenkette>

<ATTRIBUTWERT> ::= <Attributwert in dem zu dem Datenbank-Datentyp
    korrespondierenden JSON-Datentyp>
```

Listing 5: GET-Rückgabe mit genau einer Ressource

Bei einem Zugriff über ein Foreign Key Attribut einer Entität können mehrere Ressourcen zurückgeliefert werden. Listing 6 beschreibt das JSON-Objekt bei einem Zugriff über ein Foreign Key Attribut einer Entität.

```
<Rückgabe> ::= "{" <ENTITÄT> ":" "[" <RESSOURCE> "]" "}"

<ENTITÄT> ::= <Entität aus dem konzeptionellen Schema>

<RESSOURCE> ::= "{" <KEYVALUE>  "}" "," <RESSOURCE> | "{" <KEYVALUE>  "}"

<KEYVALUE> ::= <ATTRIBUT> ":" <ATTRIBUTWERT> "," <KEYVALUE> | <ATTRIBUT>
    ":" <ATTRIBUTWERT>

<ATTRIBUT> ::= <Attributname aus dem konzeptionellen Schema als JSON-
    Datentyp Zeichenkette>

<ATTRIBUTWERT> ::= <Attributwert in dem zu dem Datenbank-Datentyp
    korrespondierenden JSON-Datentyp>
```

Listing 6: GET-Rückgabe mit mehreren Ressourcen

Konnte eine Ressource selektiert werden, so ist der HTTP-Status-Code 200. Der HTTP-Status-Code ist 404, wenn die Ressource nicht existiert.

DELETE

Ein DELETE-Request auf eine Ressource erfolgt ausschließlich über deren Primary Key (Listing 7).

```
<Ressource> ::= "http://babeauf.nt.fh-koeln.de/DAL/" <ENTITÄT> "/" <PRIK>

<ENTITÄT> ::= <Entität aus dem konzeptionellen Schema>

<PRIK> ::= <Primary Key>
```

Listing 7: Ressource in Backus-Naur-Form für DELETE-Request

Wird bei einem DELETE-Request eine Fremdschlüsselbeziehung verletzt, ist der HTTP-Status-Code 400 und es werden die abhängigen Ressourcen (Child-Ressourcen) in einem JSON-Objekt mit dem Aufbau aus Listing 3 zurückgegeben.

```
{
    "status": "error",
```

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

```
    "msg": "SQL error: foreign key constraint failed",
    "data": [
        "http://babeauf.nt.fh-koeln.de/DAL/Sensor/1671659",
        "http://babeauf.nt.fh-koeln.de/DAL/Sensor/1690463"
    ]
}
```

Listing 8: Beispielrückgabe für ein DELETE-Request mit Foreign Key Verletzung

Um eine Fremdschlüsselverletzung aufzulösen und die Parent-Ressource zu löschen, kann über die angegeben URI einzeln auf die Child-Ressourcen zugegriffen und der Verweis auf die Parent-Ressource mit einem Update der Child-Ressource entfernt werden (siehe PUT).

Bei einem erfolgreichen Löschen einer Ressource ist der HTTP-Response-Body leer und es wird als HTTP-Status-Code 204 zurückgegeben. Das Löschen einer nichtexistenten Ressource gilt als erfolgreicher Löschvorgang und gibt ebenfalls den HTTP-Status-Code 204 mit leerem HTTP-Response-Body zurück.

PUT

Über einen PUT-Request werden Ressourcen hinzugefügt (INSERT) oder vorhandene Ressourcen aktualisiert (UPDATE).

```
<Ressource> ::= "http://babeauf.nt.fh-koeln.de/DAL/" <ENTITÄT>
<ENTITÄT> ::= <Entität aus dem konzeptionellen Schema>
```

Listing 9: Ressource in Backus-Naur-Form für PUT-Request

Der HTTP-Request enthält dazu im HTTP-Request-Body ein JSON-Objekt mit der einzufügenden oder zu aktualisierenden Ressource (Listing 10).

```
<Übergabe> ::= "{" <KEYVALUE> "}"
<KEYVALUE> ::= <ATTRIBUT> ":" <ATTRIBUTWERT> "," <KEYVALUE> | <ATTRIBUT>
    ":" <ATTRIBUTWERT>
<ATTRIBUT> ::= <Attributname aus dem konzeptionellen Schema als JSON-
    Datentyp Zeichenkette>
<ATTRIBUTWERT> ::= <Attributwert als beliebiger gültiger JSON-Datentyp>
```

Listing 10: JSON-Objekt einer Ressource für einen PUT-Request

Der JSON-Datentyp des Attributwerts muss nicht mit dem Datenbank-Datentyp übereinstimmen. Der DAL castet den Attributwert anhand des Datentyps des Attributs, der im JSON globalen Schema hinterlegt ist. So können Zahlenwerte wahlweise als numerischer JSON-Datentyp oder als Zeichenkette übergeben werden.

Wird eine neue Ressource angelegt, so wird der HTTP-Status-Code 201 mit der angelegten Ressource als JSON-Objekt im HTTP-Response-Body mit dem Aufbau aus Listing 10 zurückgegeben. Im HTTP-Response-Header beschreibt das Location-Feld die URI der Ressource (siehe 201 Created). Bei der Anlage von neuen Ressourcen muss das Primary Key Attribut mitgegeben werden, darf aber unbelegt bleiben (NULL-Wert oder leerer String). Wird ein unbelegter Attributwert für das Primary Key Attribut mitgegeben, wird automatisch eine UUID für die Ressource generiert. Andernfalls wird der beim Aufruf mitgegeben Attributwert als Primary Key übernommen.

Wird eine vorhandene Ressource durch den HTTP-GET-Request aktualisiert, wird der HTTP-Status-Code 204 mit leerem HTTP-Response-Body zurückgegeben.

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

POST

Ein POST-Request verknüpft zwei Entitäten über eine Fremdschlüsselbeziehung. Die in der URI identifizierte Ressource ist hierbei die Child-Ressource und die in dem HTTP-Request-Body als JSON –Objekt übergebende Ressource die Parent-Ressource. Existiert die Parent-Ressource noch nicht, wird diese automatisch vom DAL angelegt.

```
<Ressource> ::= "http://babeauf.nt.fh-koeln.de/DAL/" <ENTITÄT> "/" <PRIK>

<ENTITÄT> ::= <Entität aus dem konzeptionellen Schema>

<PRIK> ::= <Primary Key>
```

Listing 11: Ressource in Backus-Naur-Form für einen POST-Request

Die Parent-Ressource ist die Ressource deren Primary Key als Foreign Key der Child-Ressource zugordnet wird. Der DAL erkennt automatisch das passende Foreign Key Attribut der Child-Ressource und füllt dieses mit dem Primary Key der Parent-Ressource. Es wird nur von einer einmaligen Verknüpfung zwischen zwei Entitäten ausgegangen. Die Verknüpfung zwischen zwei Entitäten über mehrere verschiedene Foreign Key Attribute ist nicht vorgesehen. Der DAL kann bei Bedarf dahingehend noch erweitert werden.

```
<Parent> ::= "{" <ENTITÄT> ":" "{" <KEYVALUE> "}" "}"

<ENTITÄT> ::= <Entität aus dem konzeptionellen Schema>

<KEYVALUE> ::= <ATTRIBUT> ":" <ATTRIBUTWERT> "," <KEYVALUE> | <ATTRIBUT>
    ":" <ATTRIBUTWERT>

<ATTRIBUT> ::= <Attributname aus dem konzeptionellen Schema als JSON-
    Datentyp Zeichenkette>

<ATTRIBUTWERT> ::= <Attributwert als beliebiger gültiger JSON-Datentyp>
```

Listing 12: JSON-Objekt einer Parent-Ressource bei einem POST-Request

Existiert die Parent-Entität schon, können alle Attribute bis auf das Primary Key Attribut bei dem JSON-Objekt weggelassen werden.

Existiert die Parent-Entität noch nicht und soll diese von dem DAL angelegt werden, darf der Primary Key Attributwert unbelegt bleiben (null oder leerer String). Bei einem unbelegten Primary Key Attributwert, wird automatisch eine UUID für die Ressource generiert. Andernfalls wird der beim Aufruf mitgegeben Attributwert als Primary Key übernommen.

Analog der Anlage und Aktualisierung von neuen Ressourcen muss der JSON-Datentyp der Attributwerte wegen seines, wie bereits beim PUT-Zugriff oben beschriebenen inhärenten Casting-Mechanismus nicht mit dem Datenbank-Datentyp übereinstimmen.

Mögliche Versionierung über den DAL

Änderungen an dem konzeptionellen Schema können aufgrund der generischen API des DALs zu einer Fehlfunktion vorhandener Programme führen. Um für Programme eine gleichbleibende API zu gewährleisten, wird im Folgenden auf eine Versionierung der DAL-API eingegangen.

Eine Versionsnummer wird bei der Nutzung des DALs vor den Entitätsnamen hinzugefügt. Das JSON globale Schema wird um die Versionsnummer erweitert. Diese wird bei Änderungen des konzeptionellen Schemas und einer Neugenerierung des JSON globalen Schemas fortlaufend erhöht. Der DAL ermittelt anhand der in der URI enthaltenen Versionsnummer das zu nutzenden JSON globale Schema und liefert so den Programmen eine gleichbleibende API.

```
<Ressource>  ::=  "http://babeauf.nt.fh-koeln.de/DAL/"  <VERSION>  "/"
```

Gefördert durch:

Bundesministerium für Wirtschaft und Technologie

aufgrund eines Beschlusses des Deutschen Bundestages

```
    <ENTITÄT> <IDENTIFIER>

<VERSION> ::= <Versionsnummer des konzeptionellen Schemas>

<IDENTIFIER> ::= "/" <PRIK> | "/" <FKEYColumn> "/" <FKEY> | "/"
    <UNIQUEColumn> "/" <UKEY> |

<ENTITÄT> ::= <Entität aus dem konzeptionellen Schema>

<PRIK> ::= <Primary Key>

<FKEYColumn> ::= <Foreign Key Attribut aus dem konzeptionellen Schema >

<FKEY> ::= <Fremdschlüssel>

<UNIQUEColumn> ::= <Attribut aus dem konzeptionellen Schema mit UNIQUE
    Index>

<UKEY> ::= <Attributwert der UNIQUEColumn>
```

Listing 13: Ressource des DALs in Backus-Naur-Form mit Versionsnummer

Der DAL nutzt das der Versionsnummer entsprechende JSON globale Schema, um bei einem GET-Request (SELECT) die JSON-Rückgabeobjekte entsprechend zu formatieren und bei einem PUT-Request (UPDATE/INSERT) die von den Programmen übergebenen JSON-Objekte gemäß des Tabellenschemas zu einem INSERT-Statement umzubauen.

Folgende Fälle können bei einer Änderung des konzeptionellen Schemas auftreten und müssen für die Realisierung einer Versionierung gesondert beachtet werden:

1. Löschen einer Spalte
   Bei dem Löschen einer Spalte einer Entität wird die Spalte nicht physikalisch in der zugrundeliegenden Datenbank gelöscht. Der DAL kennt anhand der übergebenen Versionsnummer, ob eine logisch gelöschte Spalte bei einem GET (SELECT) in dem erzeugten JSON-Objekt mit zurückgegeben werden muss, oder nicht. Bei einem PUT/POST (UPDATE/INSERT) mit älterer Versionsnummer wird die Spalte wie gewohnt aktualisiert oder neue Werte eingefügt. Bei einem PUT/POST (UPDATE/DELETE) mit aktueller Versionsnummer werden logisch gelöschte Spalte bei einem PUT/POST (UPDATE/INSERT) ignoriert.
2. Hinzufügen einer Spalte
   Bei dem Hinzufügen einer Spalte einer Entität wird die Spalte in der zugrundeliegenden Datenbank angelegt. Bei einem GET (SELECT) auf einen älteren Datensatz mit neuester Schema-Versionsnummer wird die Spalte mit leerem Wert oder den bei der Anlage bestimmten Default-Wert zurückgegeben. Bei einem GET (SELECT) auf einen Datensatz mit älterer Schema-Versionsnummer wird die Spalte bei dem erzeugten JSON-Objekt nicht mit zurückgegeben. Bei einem PUT/POST (UPDATE/INSERT) mit aktueller Versionsnummer wird die Spalte wie gewohnt aktualisiert oder neue Werte eingefügt. Bei einem PUT/POST (UPDATE/DELETE) mit älterer Versionsnummer werden logisch gelöschte Spalte bei einem PUT/POST (UPDATE/INSERT) ignoriert.
3. Ändern des Datentyps einer Spalte
   Änderungen an Datentypen von Attributen einer Entität werden in der Datenbank physikalisch ausgeführt. Der DAL liest über die genutzte Version bei einem GET (SELECT) den zu nutzenden Datentyp des Attributs bei dem zurückzugebenden JSON-Objekts. Bei einem PUT/POST (UPDATE/INSERT) wird der Wert des Attributs des übergebenden JSON-Objekts auf den aktuellen Datentyp der Spalte gecastet.
4. Umbenennen einer Spalte

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Beim Umbenennen einer Spalte wird eine neue Spalte mit dem neuen Namen angelegt und die alte Spalte bleibt mit ihrem bisherigen Namen bestehen. Beim Anlegen einer neuen Spalte muss der Inhalt der alten Spalte in die neue Spalte kopiert werden.

6. Momentan generierte RESTful-Services des DALs

Das JSON globale Schema verfügt momentan über 69 Entitätentypen. Für jeden Entitätstyp gibt es vier Services (GET, PUT, POST, DELETE). Somit verfügt der DA über Z = 4 x 69 RESTful-Services.

7. Messwert-Service

Dieses Kapitel beschreibt den Zugriff auf in der Cloud gespeicherte Messwerte über eine als Webservice realisierte API. Die API kennt die Datenhaltung der Messwerte in der Cloud-Datenbank und bietet eine einheitliche Zugriffsart auf die ursprünglichen und auch aggregierten Messwerte eines Zeitraums.

Technische Beschreibung

Die API für den Zugriff auf die Messwerte eines zeitlichen Intervalls wird über einen parametrisierbaren Webservice bereitgestellt und dient als Ersatz für lesende Zugriffe auf die in den SensorCloud-Datenbanken gespeicherten Messwerten. Hierdurch wird für alle nutzenden Programme die gleiche Performance bei lesenden Zugriffen gesichert. Gleichzeitig stellt die API integrierte Aggregationsfunktionen auf Messwerte bereit. Listing 1 beschreibt die URI des Webservice in Backus-Naur-Form.

```
<Ressource> ::= "http://babeauf.nt.fh-
     koeln.de/Messwerte/index.php?senid=" <SENID> "&parvor_nr=" <PARVOR>
     <OPTIONEN>

<SENID> ::= <SenID des Sensors>

<PARVOR> ::= <Index der Messwertbeschreibung innerhalb der
     Parsingvorschrift>

<OPTIONEN> ::= [ <OPTION1> ] [ <OPTION2> ] [ <OPTION3> ] [ <OPTION4> ]

<OPTION1> ::= "&ts=" <TIMESTAMP>

<TIMESTAMP> ::= <UNIX-Zeitstempel>

<OPTION2> ::= "&period=" <INTERVALL>

<INTERVALL> ::= <ZAHL> <QUANTIFIER>

<ZAHL> ::=  <ZIFFER> <ZAHL> | <ZIFFER>

<ZIFFER> ::= "0" | "1" | "2" | "3" | … | "9"

<QUANTIFIER> ::= "m" | "w" | "d"

<OPTION3> ::= "aggregiert=" <BOOLEAN>

<BOOLEAN> ::= "true" | "false"

<OPTION4> ::= "dx=" <ZAHL>
```

Listing 14: Ressource in Backus-Naur-Form

Der Webservice kann wahlweise über GET-Requests oder POST-Requests verwendet werden.

Gefördert durch:

Bundesministerium für Wirtschaft und Technologie

aufgrund eines Beschlusses des Deutschen Bundestages

Parameter

senid

Eindeutige ID für einen Sensor aus der Entität Sensor. Dieser Parameter muss angegeben werden.

parvor_nr

Index der Messwertbeschreibung innerhalb der Parsingvorschrift aus der Entität SensorProdukt. Dieser Parameter muss angegeben werden.

ts

Dieser Parameter ist optional. Wird er angegebenen, so bestimmt er den UNIX-Zeitstempel des Startzeitpunktes des Selektionszeitraums. Das über *period* spezifizierte Intervall wird dann auf den Startzeitpunkt gerechnet und ergibt das Ende des Selektionszeitraums.

Wird kein UNIX-Zeitstempel mitgegeben, so wird aus der Entität MesswertAktuell der letzte Zeitpunkt eines Messwertes bestimmt und dieser als Endezeitpunkt für das über *period* spezifizierte Intervall verwendet. Das über *period* spezifizierte Intervall wird dann von dem Endezeitpunkt subtrahiert und ergibt den Beginn des Selektionszeitraums.

period

Dieser Parameter ist optional und beschreibt die Dauer des Selektionszeitraums. Der Parameter besteht aus einer Zahl gefolgt von d (Tag(e)), w (Woche(n)) oder m (Monat(e)). Wird keine Dauer mitgegeben, wird als Standard 1d (1 Tag) verwendet.

aggregiert

Dieser Parameter ist optional und bestimmt, ob die Messwerte mit den in der Parsingvorschrift angegebenen Aggregationsfunktionen zusammengefasst oder die exakten Messwerte zurückgegeben werden sollen. Wenn der Parameter nicht angegeben wird, werden die exakten Messwerte zurückgegeben.

dx

Dieser Parameter ist optional und beschreibt den zeitlichen Mindestabstand zwischen den zurückgegebenen Messwerten in Sekunden. Wird der Parameter nicht angegeben, unterscheidet sich der Standardwert anhand des verwendeten Intervalls (period).

period	Default
d	120
w	900
m	1800

Tabelle 1: dx-Standardwerte in Abhängigkeit des Selektionszeitraums

Rückgabe

Die Rückgabe erfolgt als JSON-Objekt im Body der HTTP-Response. Listing 6 beschreibt das JSON-Objekt.

```
<Rückgabe> ::= "{" '"sid":' <SENID>  ',"n":' <NAME> ',"bt":' <TIMESTAMP>
     ',"t":' "[" <TIMESTAMPDELTAS> "]" ',"v":' "[" <MESSWERTE> "]"
     ',"period":' "{" '"number":' <ZAHL> ',"unit":'" <QUANTIFIER> "}"
```

Gefördert durch:

Bundesministerium für Wirtschaft und Technologie

aufgrund eines Beschlusses des Deutschen Bundestages

```
     ","countRows":' <ZAHL>  ","executionTime":' <FLOAT> ","dx":' <ZAHL>

<SENID> ::= <SenID des Sensors>

<NAME> ::= <Name der physikalischen Größe>

<TIMESTAMP> ::= <UNIX-Timestamp als Bezugstimestamp>

<TIMESTAMPDELTAS> ::= <TIMESTAMPDELTA> "," <TIMESTAMPDELTAS> |
     <TIMESTAMPDELTA>

<TIMESTAMPDELTA> ::= <Delta zum in bt angegeben Timestamp>

<MESSWERTE> ::= <MESSWERT> "," < MESSWERTE > | <MESSWERT>

<MESSWERT> ::= <gemessener Messwert>

<ZAHL> ::=  <ZIFFER> <ZAHL> | <ZIFFER>

<ZIFFER> ::= "0" | "1" | "2" | "3" | … | "9"

<FLOAT> ::= <ZAHL> "." <ZAHL>

<QUANTIFIER> ::= "m" | "w" | "d"
```

Listing 15: GET-Rückgabe mit mehreren Ressourcen

Rückgabeparameter

sid

Eindeutige ID für einen Sensor aus der Entität Sensor als JSON-Datentyp Zeichenkette. Dieser Parameter entspricht dem Parameter aus dem Webserviceaufruf.

n

Der Name des gemessenen physikalischen Phänomens als JSON-Datentyp Zeichenkette.

bt

UNIX-Zeitstempel auf den sich alle in *t* angegeben Zeitpunkte beziehen (*basetime*) als JSON-Datentyp Zahl.

t

Feld mit Timestampdeltas zur basetime als JSON-Datentyp Zahl. Ein Timestampdelta korrespondiert zum Messwert im *v*-Feld an der gleichen Feldposition.

v

Feld mit den Messwerten. Ein Messwert korrespondiert zum Timestampdelta im *t*-Feld an der gleichen Feldposition. Der JSON-Datentyp der Rückgabe variiert je nach dem in der Sensorproduktsemantik angegeben Datentyp.

period

Beschreibt die Dauer des Selektionszeitraums und teilt sich auf in *number* und *unit*. Dieser Parameter entspricht dem Parameter aus dem Webserviceaufruf

number

Numerischer Anteil des Selektionszeitraums als JSON-Datentyp Zahl.

Gefördert durch:

Bundesministerium für Wirtschaft und Technologie

aufgrund eines Beschlusses des Deutschen Bundestages

unit

Die Einheit des Selektionszeitraums (d=Tag(e), w=Woche(n), M=Monat(e)) als JSON-Datentyp Zeichenkette.

countRows

Anzahl der Datensätze zu dem Selektionszeitraums in der Datenbank als JSON-Datentyp Zahl. (Anm.: Die Anzahl der Datensätze in der Datenbank ist >= der Anzahl der zurückgegebenen Messwerte.)

executionTime

Gesamtdauer der Ausführung des Webservice als JSON-Datentyp Zahl.

dx

Beschreibt den zeitlichen Mindestabstand zwischen den zurückgegebenen Messwerten in Sekunden als JSON-Datentyp Zahl. Dieser Parameter entspricht dem Parameter aus dem Webserviceaufruf.

Fehlerausgabe

Der Messwert-Service nutzt keine HTTP-Status-Codes als Fehercodes. Bei Fehlern beendet sich der Webservice mit sprechender Fehlermeldung in dem zurückgegebenen JSON-Objekt.

8. Weitere Webservices

An der Fachhochschule Köln wurden und werden als Abschlussarbeiten u.a. eine Visualisierung von Messwerten, Apps und Web-Services zur Verwaltung von Bestandsdaten und Sensoren eines Kunden, sowie die Steuerung eines Roboters über die SensorCloud realisiert.

Im Rahmen dieser Abschlussarbeiten werden RESTful-Services bereitgestellt, die auch von zukünftigen Projekten genutzt werden sollen. Dabei werden Entitäten des konzeptionellen Datenbankschemas [1] in Ressourcen der RESTful-Services gekapselt. Dabei kann eine Ressource einer oder mehrere Entitäten des konzeptionellen Datenbankschemas behandeln.

Aktuell stehen RESTful-Services für folgende Entitätstypen zur Verfügung:

- AktorAnforderung
- Nutzer
- AktorVerbund
- Sensor
- SensorSemantik
- Messwert

Folgende REST-Services sind über http://babeauf.nt.fh-koeln.de:8080/SensorCloudRest/crud/ erreichbar und liefern JSON-Datensätze als Rückgabe. Im Anschluss an die folgenden Tabellen sind Beispiele der Rückgaben angegeben.

REST Service Aktor getAktorListByAktSerID

URL	http://babeauf.nt.fh-koeln.de:8080

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

	/SensorCloudRest/crud/Aktor/AktSerID/734d959d-9aa1-47e1-92b4-febdd89583f0
Java Klasse des Services	HttpAktor
Methodenname	getAktorListByAktSerID
Aufruf mit HTTP-Methode	GET
Methodenkopf	**public** String getAktorListByAktSerID(@PathParam("aktSerID") String aktSerID)
Parameter	PathParameter mit AktSerID der CF AktorService
Rückgabetyp	Objekt von der Java Klasse AktorList im JSON Format
Beschreibung	Der Web Service liefert für einen AktorService die Liste aller Aktoren, die diesem Service angehören. Aktoren von verschiedenen Herstellern oder mit unterschiedlichen Konfigurationen können trotzdem den selben Service anbieten.
Verwendete Coulumn Families	AktorServiceMitglieder, Aktor

REST Service AktorService getAktorServicesByAktID

URL	http://babeauf.nt.fh-koeln.de:8080 /SensorCloudRest/crud/AktorService/AktID/38f46f3e-9f13-45b7-8294-e1505e1e358a
Java Klasse des Services	HttpAktorService
Methodenname	getAktorServiceByAktID
Aufruf mit HTTP-Methode	GET
Methodenkopf	**public** String getAktorServicesByAktID(@PathParam("aktID") String aktID)
Parameter	PathParameter mit AktID der CF Aktor
Rückgabetyp	Objekt von der Klasse AktorServiceList im JSON Format
Beschreibung	Der Service liefert eine Liste aller AktorServices, die dem angegebenen Aktor angehören. Der AktorService beschreibt, was der Aktor für eine Anwendung anbietet oder in welchem Zusammenhang dieser mit Umwelt

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

	kooperiert.
Verwendete DBKlassen	DBAktorServiceMitglieder, DBAktorService
Verwendete Coulumn Families	AktorServiceMitglieder, AktorService

REST Service AktorServiceFunktion getFunktionListByAktSerID

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/AktorServiceFunktion/AktSerID/0df517b3-1fdb-4942-9bab-b2e257b32f94
Java Klasse des Services	HttpAktorServiceFunktion
Methodenname	getFunktionListByAktSerID
Aufruf mit HTTP-Methode	GET
Methodenkopf	**public** String getFunktionListByAktSerID(@PathParam("aktSerID") String aktSerID)
Parameter	PathParameter mit AktSerID der CF AktorService
Rückgabetyp	Objekt von der Java Klasse AktorServiceFunktionList im JSON Format
Beschreibung	Der Service liefert eine Liste aller AktorServiceFunktionen, die dem angegebenen AktorService angehören.
Verwendete DBKlassen	DBAktorserviceMitglieder, DBAktorServiceFunktion
Verwendete Coulumn Families	AktorServiceFunktionMitglieder, AktorServiceFunktion

REST Service AktorVerbund getAktorVerbundByNutStaID

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/AktorVerbund/NutStaID/c24d959d-9aa1-47e1-92b4-4ebdd89583f0
Java Klasse des Services	HttpAktorVerbund
Methodenname	getAktorVerbundByNutStaID
Aufruf mit HTTP-Methode	GET
Methodenkopf	**public** String getAktorVerbundByNutStaID(

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

	@PathParam("nutStaID") String nutStaID)
Parameter	PathParameter mit NutStaID der CF NutzerStammdaten
Rückgabetyp	Objekt von der Java Klasse AktorVerbundList im JSON Format
Beschreibung	Der Service liefert eine Liste aller Verbünde mit Aktoren die einer Person gehören. So sucht der Service erst einmal, welche Aktoren einem Nutzer gehören. Und anschließend über AktorVerbundMitglieder die entsprechenden Verbünde.
Verwendete DBKlassen	DBAktor, DBAktorVerbundMitglieder, DBAktorVerbund
Verwendete Coulumn Families	Aktor, AktorVerbundMitglieder, AktorVerbund

REST Service AktorVerbund getAktorByAktorVerID

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/AktorVerbund/AktVerID/d6b05143-63ad-4ad9-804d-00e89cf1baa9
Java Klasse des Services	HttpAktorVerbund
Methodenname	getAktorByAktVerID
Aufruf mit HTTP-Methode	GET
Methodenkopf	**public** String getAktorByAktVerID(@PathParam("aktVerID") String aktVerID)
Parameter	PathParameter mit AktVerID der CF AktorVerbund
Rückgabetyp	Objekt von der Java Klasse AktorList im JSON Format
Beschreibung	Der Service liefert eine Liste aller Aktoren, die in einem Verbund zusammengefasst sind. So sucht der Service über die AktorVerbundMitglieder die entsprechenden Aktoren des Verbundes.
Verwendete DBKlassen	DBAktor, DBAktorVerbundMitglieder, DBAktorVerbund
Verwendete Coulumn Families	Aktor, AktorVerbundMitglieder, AktorVerbund

REST Service AktorVerbund createAktorVerbund

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/AktorVerbund
Java Klasse des Services	HttpAktorVerbund
Methodenname	createAktorVerbund
Aufruf mit HTTP-Methode	PUT
Methodenkopf	**public** String createAktorVerbund(String data)
Parameter	Objekt von der Java Klasse AktorVerbundMitAktor im JSON Format
Rückgabetyp	Die UUID des eingefügten Verbundes als String
Beschreibung	Der Service fügt einen **neuen** Verbund, mit einem Aktor als Mitglied in die DB ein. So werden alle erforderlichen Daten in die CF AktorVerbund hinterlegt. Und in AktorVerbundMitglieder ein Verweis des Aktoren und des Verbundes hinterlegt. Somit ist ein Aktor mit einem neu erzeugten „Aktor“-Verbund verbunden.
Verwendete DBKlassen	DBAktorVerbundMitglieder, DBAktorVerbund
Verwendete Coulumn Families	AktorVerbundMitglieder, AktorVerbund

REST Service AktorVerbund addAktorToAktorVerbund

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/AktorVerbund
Java Klasse des Services	HttpAktorVerbund
Methodenname	addAktorToAktorVerbund
Aufruf mit HTTP-Methode	POST
Methodenkopf	**public** String addAktorToAktorVerbund(String data)
Parameter	Objekt von der Java Klasse AktorVerbundMitAktor im JSON Format
Rückgabetyp	Die UUID des geänderten Verbundes als String im PlainText
Beschreibung	Der Service fügt zu einem **bestehenden** „Aktor“-Verbund, einen Aktor hinzu. So fügt der Service lediglich ein „Verweis“ in die CF AktorVerbundMitglieder ein. Somit

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

	wird ein Aktor zu einem „Aktor“-Verbund hinzugefügt.
Verwendete DBKlassen	DBAktorVerbundMitglieder, DBAktorVerbund
Verwendete Coulumn Families	AktorVerbundMitglieder, AktorVerbund

REST Service Event getEventObjectListByNutStaID

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/Event/NutStaID/c24d959d-9aa1-47e1-92b4-4ebdd89583f0
Java Klasse des Services	HttpEvent
Methodenname	getEventObjectListByNutStaID
Aufruf mit HTTP-Methode	GET
Methodenkopf	**public** String getEventObjectListByNutStaID(@PathParam("nutStaID") String nutStaID)
Parameter	PathParameter mit NutStaID der CF NutzerStammdaten
Rückgabetyp	Objekt von der Java Klasse EventList im JSON Format, oder wenn es keine Einträge gibt eine String mit dem Wert „leer“.
Beschreibung	Der Service liefert eine Liste von Objekten, welche aus der CF Event stammen. Ein Event ist aber keiner Person direkt zugeordnet. Die Sensoren und Aktoren sind jedoch einer Person zugeordnet. Ein Event enthält über die CF SensorEvent und CF EventAktion verweise zu den Sensoren und Aktoren. So kann eine Zuordnung des Events über die Sensoren/Aktoren einer Person erfolgen. Die Reihenfolge der Aufrufe der CF: Sensor->SensorEvent-> EventMitglieder->Event.
Verwendete DBKlassen	DBSensor, DBSensorEvent, DBEventMitglieder, DBEvent
Verwendete Coulumn Families	Sensor, SensorEvent, EventMitglieder, Event

REST Service Event getEventRegelByEveID

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/Event

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Java Klasse des Services	HttpEvent
Methodenname	getEventRegelByEveID
Aufruf mit HTTP-Methode	GET
Methodenkopf	**public** String getEventRegelByEveID(@PathParam("eveID") String eveID)
Parameter	PathParameter mit EveID der CF Event
Rückgabetyp	Objekt von der Java Klasse EventRegel im JSON Format
Beschreibung	Der Service liefert eine komplette Regel eines Events für die entsprechende Event ID. Die Event ID kann z.B. über den Service 3.3.11 erfragt werden. Die Regel entält den Kompleten Datensatz von: SensorEvent, EventAktion, EventBenachrichtigung, und Event.
Verwendete DBKlassen	DBSensorEvent, DBEventMitglieder, DBEvent, DBEventAktion, DBEventBenachrichtigung
Verwendete Coulumn Families	SensorEvent, EventMitglieder, Event, EventAktion, EventBenachrichtigung

REST Service Event updateEventRegel

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/Event
Java Klasse des Services	HttpEvent
Methodenname	updateEventRegel
Aufruf mit HTTP-Methode	POST
Methodenkopf	**public** String updateEventRegel(String data)
Parameter	Objekt von der Java Klasse EventRegel im JSON Format
Rückgabetyp	String mit Wert „ausgefuehrt“ Wenn Service denVorgang abgeschlossen hat.
Beschreibung	Der Service kann die SensorEvent und EventAktion entsprechend dem übersendeten Datensatz ändern.

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Verwendete DBKlassen	DBSensorEvent DBEventAktion
Verwendete Coulumn Families	SensorEvent, EventAktion

REST Service Gruppen getGruppenByID

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/Gruppen/NutStaID/c24d959d-9aa1-47e1-92b4-4ebdd89583f0
Java Klasse des Services	HttpGruppen
Methodenname	getGruppeByID
Aufruf mit HTTP-Methode	GET
Methodenkopf	**public** String getGruppeByID(@PathParam("nutStaID") String nutStaID)
Parameter	PathParameter mit NutStaID der CF NutzerStammdaten
Rückgabetyp	Objekt von der Java Klasse EventRegel im JSON Format
Beschreibung	Der Service liefert eine Liste mit Gruppen, in denen die jeweilige Person ein Mitglied ist.
Verwendete Java DBKlassen	DBGruppenMitglieder, DBGruppen
Verwendete Coulumn Families	GruppenMitglieder, Gruppen

REST Service Gruppen getMitgliederByGruID

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/Gruppen/GruID/f4141ff6-d556-40ad-8561-b1797a30676a
Java Klasse des Services	HttpGruppen
Methodenname	getMitgliederByGruID
Aufruf mit HTTP-Methode	GET
Methodenkopf	**public** String getMitgliederByGruID(@PathParam("gruID") String gruID)

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Parameter	PathParameter mit GruID der CF Gruppen
Rückgabetyp	Objekt von der Java Klasse MitgliederList im JSON Format
Beschreibung	Der Service liefert eine Liste von Mitgliedern, für die angegebene Gruppe(GruID). Die Daten des Mitglieds umfassen dabei die Email und das Profil der Personen.
Verwendete Java DBKlassen	DBGruppenMitglieder, DBNutzerStammdaten, DBNutzerEmail
Verwendete Coulumn Families	GruppenMitglieder, NutzerStammdaten, NutzerEmail

REST Service Gruppen insertMitglied

URL	http://babeauf.nt.fh-koeln.de:8080/SensorCloudRest/crud/Gruppen/inviteMitglied
Java Klasse des Services	HttpGruppen
Methodenname	insertMitglied
Aufruf mit HTTP-Methode	PUT
Methodenkopf	**public** String insertMitglied(String data)
Parameter	Objekt von der Java Klasse GruppenMitglied im JSON Format
Rückgabetyp	Wenn es keine Person in der DB gibt die Mitglied werden soll einer Gruppe, dann wird eine Meldung ausgegeben. Ansonsten die UUID des neuen Mitglieds.
Beschreibung	Der Service fügt einer Gruppe ein Mitglied hinzu, entsprechend dem übergebenen Datensatz. Es wird in der CF GruppenMitglieder ein Verweis hinterlegt, über Person und der dazu zugehörenden Gruppe.
Verwendete Java DBKlassen	DBGruppenMitglied, DBNutzerEmail
Verwendete Coulumn Families	GruppenMitglied, NutzerEmail

REST Service Gruppen insertGruppe

URL	http://babeauf.nt.fh-koeln.de:8080/SensorCloudRest/crud/Gruppen/createGruppe

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Java Klasse des Services	HttpGruppen
Methodenname	insertGruppe
Aufruf mit HTTP-Methode	PUT
Methodenkopf	**public** String insertGruppe(String data)
Parameter	Objekt von der Java Klasse NeueGruppeMitNutzer im JSON Format
Rückgabetyp	UUID von GruppenMitglieder, in der ein Verweis steht, welche Person zu welcher Gruppe hinzugefügt wurde.
Beschreibung	Es wird eine neue Gruppe erstellt und der Gruppe ein Mitglied hinzugefügt, gemäß dem übergebenen Datensatz.
Verwendete Java DBKlassen	DBGruppenMitglied, DBNutzerEmail
Verwendete Coulumn Families	GruppenMitglied, NutzerEmail

REST Service Gruppen deleteGruppe

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/Gruppen/delete
Java Klasse des Services	HttpGruppen
Methodenname	deleteGruppe
Aufruf mit HTTP-Methode	POST
Methodenkopf	**public** String deleteGruppe(String data)
Parameter	Objekt von der Java Klasse NutStaIDAndGruID im JSON Format
Rückgabetyp	Bestätigung über das verlassen der Gruppe: „Gruppe verlassen“
Beschreibung	Die Person wird aus der Gruppe gestrichen, in der sie sich vorher befand.
Verwendete Java DBKlassen	DBGruppenMitglied
Verwendete Coulumn Families	GruppenMitglied

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

REST Service Login authentifizieren

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/Login/authentifizieren
Java Klasse des Services	HttpLogin
Methodenname	authentifizieren
Aufruf mit HTTP-Methode	POST
Methodenkopf	**public** String authetifizieren(String data)
Parameter	Objekt von der Java Klasse Login im JSON Format
Rückgabetyp	Objekt von der Java Klasse NutzerStammdaten im JSON Format, wenn das Einloggen erfolgreich war. Ansonsten wird der String mit Wert „Denied“ zurückgegeben.
Beschreibung	Bevor der User auf die Funktionen der Android App erhält, muss er sich erst einmal einloggen. Für das Einloggen werden die Email und das Passwort gebraucht.
Verwendete Java DBKlassen	DBNutzerStammdaten, DBNutzerEmail, DBNutzerSicherheit
Verwendete Coulumn Families	NutzerStammdaten, NutzerEmail, NutzerSicherheit

REST Service Messert getMesswerteBySenIDForDay

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/Messwert/SenID/{sensorID}/ MesWerNam/{meswerNam}/ MesWerTimYea/{jahr}/ MesWerTimMon/{monat}/ MesWerTimDay/{tag}
Java Klasse des Services	HttpMesswert
Methodenname	getMesswerteBySenIDForDay
Aufruf mit HTTP-Methode	GET
Methodenkopf	**public** String getMesswerteBySenIDForDay(@PathParam("senID") String senID,

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

	@PathParam("mesWerNam") String mesWerNam, @PathParam("mesWerTimYea") String mesWerTimYea, @PathParam("mesWerTimMon") String mesWerTimMon, @PathParam("MesWerTimDay") String MesWerTimDay)
Parameter	PathParameter mit SenID der CF Sensor, mit meswerNam der CF Messwerte. Des weiteren noch der Tag, Monat und Jahr für welchen die Messwerte ausgegeben werden sollen
Rückgabetyp	Objekt von der Java Klasse DatasetMitSemantik im JSON Format
Beschreibung	Der Service kann für einen angegebenen Sensor, dessen Messung und den angegeben Tag die werte mit dem dazugehörenden Zeitpunkt liefern. Zusätzlich wir noch die Sensorsemantik für die Messung des Sensoren mitgeliefert. So kann der Nutzer mit Hilfe der Sensorsemantik, Angaben über die Umrechnung der Werte, Einheit der Werte etc. erhalten. Die Messwerte werden aber nicht direkt über die Messwert CF bezogen, sondern über die Messlinie. Die Messlinie fasst die Messwerte, jeweils einer Stunde, zusammen. Die Messlinie wird aus Performance Gründen genutzt.
Verwendete Java DBKlassen	DBMesslinie, DBMesswert, DBSensorProdukt
Verwendete Coulumn Families	Messwert, Messlinie, SensorProdukt

REST Service NutzerEmail getNutzerEmailByID

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/NutzerEmail/NutStaID/c24d959d-9aa1-47e1-92b4-4ebdd89583f0
Java Klasse des Services	HttpNutzerEmail
Methodenname	getNutzerEmailByID
Aufruf mit HTTP-Methode	GET
Methodenkopf	**public** String getNutzerEmailByID(

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

	@PathParam("nutStaID") String nutStaID)
Parameter	PathParameter mit NutStaID der CF NutzerStammdaten
Rückgabetyp	Objekt von der Java Klasse NutzerEmail im JSON Format
Beschreibung	Der Service liefert die Liste mit E-Maildaten, für die gesuchte Person
Verwendete Java DBKlassen	DBNutzerEmail
Verwendete Coulumn Families	NutzerEmail

REST Service NutzerEmail updateNutzerEmail

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/NutzerEmail
Java Klasse des Services	HttpNutzerEmail
Methodenname	updateNutzerEmail
Aufruf mit HTTP-Methode	POST
Methodenkopf	**public** String updateNutzerEmail(String data)
Parameter	Objekt von der Java Klasse NutzerEmail im JSON Format
Rückgabetyp	Bestätigung über Ausführung des Services als String mit Wert „ausgeführt“
Beschreibung	Der Service ändert die Daten in der CF NutzerEmail, entsprechend der erhaltenen Daten.
Verwendete Java DBKlassen	DBNutzerEmail
Verwendete Coulumn Families	NutzerEmail

REST Service NutzerEmail insertNutzerEmail

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/NutzerEmail
Java Klasse des Services	HttpNutzerEmail
Methodenname	insertNutzerEmail

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Aufruf mit HTTP-Methode	PUT
Methodenkopf	**public** String insertNutzerEmail(String data)
Parameter	Objekt von der Java Klasse NutzerEmail im JSON Format
Rückgabetyp	Rückgabe der UUID der neuen Ressource
Beschreibung	Der Service fügt die Daten in der CF NutzerEmail, entsprechend der erhaltenen Daten.
Verwendete Java DBKlassen	DBNutzerEmail
Verwendete Coulumn Families	NutzerEmail

REST Service NutzerEmail deleteNutzerEmail

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/NutzerEmail
Java Klasse des Services	HttpNutzerEmail
Methodenname	deleteNutzerEmail
Aufruf mit HTTP-Methode	DELETE
Methodenkopf	**public** String deleteNutzerEmail(String data)
Parameter	UUID von Typ String im JSON Format
Rückgabetyp	Bestätigung über Ausführung als String mit Wert „ausgeführt“
Beschreibung	Der Service löscht den Eintrag in der CF NutzerEmail, der der erhaltenen UUID entsprichrt.
Verwendete Java DBKlassen	DBNutzerEmail
Verwendete Coulumn Families	NutzerEmail

REST Service NutzerSicherheit getNutzerSicherheitByID

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/NutzerSicherheit/NutStaID/c24d959d-9aa1-47e1-92b4-4ebdd89583f0

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Java Klasse des Services	HttpNutzerSicherheit
Methodenname	getNutzerSicherheitByID
Aufruf mit HTTP-Methode	GET
Methodenkopf	**public** String getNutzerSicherheitByID(@PathParam("nutStaID") String nutStaID)
Parameter	PathParameter mit NutStaID der CF NutzerStammdaten
Rückgabetyp	Objekt von der Java Klasse NutzerSicherheit im JSON Format
Beschreibung	Der Service liefert die Liste mit Passwörtern und Keys, für die gesuchte Person
Verwendete Java DBKlassen	DBNutzerSicherheit
Verwendete Coulumn Families	NutzerSicherheit

REST Service NutzerSicherheit updateNutzerSicherheit

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/NutzerSicherheit
Java Klasse des Services	HttpNutzerSicherheit
Methodenname	updateNutzerSicherheit
Aufruf mit HTTP-Methode	POST
Methodenkopf	**public** String updateNutzerSicherheit(String data)
Parameter	Objekt von der Java Klasse NutzerSicherheit im JSON Format
Rückgabetyp	Bestätigung über Ausführung des Services als String mit Wert „ausgeführt“
Beschreibung	Der Service ändert die Daten in der CF NutzerSicherheit, entsprechend der erhaltenen Daten.
Verwendete Java DBKlassen	DBNutzerSicherheit
Verwendete Coulumn Families	NutzerSicherheit

REST Service NutzerSicherheit insertNutzerSicherheit

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/NutzerSicherheit
Java Klasse des Services	HttpNutzerSicherheit
Methodenname	insertNutzerSicherheit
Aufruf mit HTTP-Methode	PUT
Methodenkopf	**public** String insertNutzerSicherheit(String data)
Parameter	Objekt von der Java Klasse NutzerSicherheit im JSON Format
Rückgabetyp	Rückgabe der UUID der neuen Ressource
Beschreibung	Der Service fügt die Daten in der CF NutzerSicherheit, entsprechend der erhaltenen Daten.
Verwendete Java DBKlassen	DBNutzerSicherheit
Verwendete Coulumn Families	NutzerSicherheit

REST Service NutzerSicherheit deleteNutzerSicherheit

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/NutzerSicherheit
Java Klasse des Services	HttpNutzerSicherheit
Methodenname	deleteNutzerSicherheit
Aufruf mit HTTP-Methode	DELETE
Methodenkopf	**public** String deleteNutzerSicherheit(String data)
Parameter	UUID von Typ String im JSON Format
Rückgabetyp	Bestätigung über Ausführung als String mit Wert „ausgeführt“
Beschreibung	Der Service löscht den Eintrag in der CF NutzerSicherheit, der der erhaltenen UUID entsprichrt.
Verwendete Java DBKlassen	DBNutzerSicherheit
Verwendete Coulumn Families	NutzerSicherheit

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

REST Service NutzerStammdaten getNutzerStammdatenByINutStaD

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/NutzerStammdaten/NutStaID/c24d959d-9aa1-47e1-92b4-4ebdd89583f0
Java Klasse des Services	Http NutzerStammdaten
Methodenname	get NutzerStammdaten ByNutStaID
Aufruf mit HTTP-Methode	GET
Methodenkopf	**public** String getNutzerStammdatenByNutStaID(@PathParam("nutStaID") String nutStaID)
Parameter	PathParameter mit NutStaID der CF NutzerStammdaten
Rückgabetyp	Objekt von der Java Klasse NutzerStammdaten im JSON Format
Beschreibung	Der Service liefert das Profil einer Person, wie etwa Name Vorname, Anrede usw.
Verwendete Java DBKlassen	DBNutzerStammdaten
Verwendete Coulumn Families	NutzerStammdaten

REST Service NutzerStammdaten updateNutzerStammdaten

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/NutzerStammdaten
Java Klasse des Services	HttpNutzerStammdaten
Methodenname	updateNutzerStammdaten
Aufruf mit HTTP-Methode	POST
Methodenkopf	**public** String updateNutzerStammdaten(String data)
Parameter	Objekt von der Java Klasse NutzerStammdaten im JSON Format
Rückgabetyp	Bestätigung über Ausführung des Services als String mit Wert „ausgeführt“

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Beschreibung	Der Service ändert die Daten in der CF NutzerStammdaten, entsprechend der erhaltenen Daten.
Verwendete Java DBKlassen	DBNutzerStammdaten
Verwendete Coulumn Families	NutzerStammdaten

REST Service NutzerStammdaten registriereNutzer

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/NutzerStammdaten
Java Klasse des Services	HttpNutzerStammdaten
Methodenname	registriereNutzer
Aufruf mit HTTP-Methode	PUT
Methodenkopf	**public** String registriereNutzer(String data)
Parameter	Objekt von der Java Klasse Registrieren im JSON Format
Rückgabetyp	Bestätigung über Ausführung des Services als String mit Wert „Nutzer erfolgreich eingefügt“
Beschreibung	Der Service fügt neue Profildaten, E-Maildaten, Passwörter, Telefondaten und die Adresse ein. Diese Eingaben gehören einer einzigen Person an. Somit kann ein kompletter Datensatz für eine Person angelegt werden.
Verwendete Java DBKlassen	DBNutzerStammdaten, DBAdresse, DBNutzerTelefon, DBNutzerSicherheit, DBNutzerEmail
Verwendete Coulumn Families	NutzerStammdaten, Adresse, NutzerTelefon, NutzerSicherheit, NutzerEmail

REST Service NutzerTelefon getNutzerTelefonByNutStaID

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/NutzerTelefon/NutStaID/c24d959d-9aa1-47e1-92b4-4ebdd89583f0
Java Klasse des Services	HttpNutzerTelefon
Methodenname	getNutzerTelefonByNutStaID

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Aufruf mit HTTP-Methode	GET
Methodenkopf	**public** String getNutzerTelefonByNutStaID(@PathParam("nutStaID") String nutStaID)
Parameter	PathParameter mit NutStaID der CF NutzerStammdaten
Rückgabetyp	Objekt von der Java Klasse NutzerTelefon im JSON Format
Beschreibung	Der Service liefert die Liste mit Telefondaten, für die gesuchte Person
Verwendete Java DBKlassen	DB NutzerTelefon
Verwendete Coulumn Families	NutzerTelefon

REST Service NutzerTelefon updateNutzerTelefon

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/NutzerTelefon
Java Klasse des Services	HttpNutzerTelefon
Methodenname	updateNutzerTelefon
Aufruf mit HTTP-Methode	POST
Methodenkopf	**public** String updateNutzerTelefon(String data)
Parameter	Objekt von der Java Klasse NutzerTelefon im JSON Format
Rückgabetyp	Bestätigung über Ausführung des Services als String mit Wert „ausgeführt“
Beschreibung	Der Service ändert die Daten in der CF NutzerTelefon, entsprechend der erhaltenen Daten.
Verwendete Java DBKlassen	DBNutzerTelefon
Verwendete Coulumn Families	NutzerTelefon

REST Service NutzerTelefon insertNutzerTelefon

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/NutzerTelefon

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Java Klasse des Services	HttpNutzerTelefon
Methodenname	insertNutzerTelefon
Aufruf mit HTTP-Methode	PUT
Methodenkopf	**public** String insertNutzerTelefon(String data)
Parameter	Objekt von der Java Klasse NutzerTelefon im JSON Format
Rückgabetyp	Rückgabe der UUID der neuen Ressource
Beschreibung	Der Service fügt neue Daten in die CF NutzerTelefon, entsprechend der erhaltenen Daten.
Verwendete Java DBKlassen	DBNutzerTelefon
Verwendete Coulumn Families	NutzerTelefon

REST Service NutzerTelefon deleteNutzerTelefon

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/NutzerEmail
Java Klasse des Services	HttpNutzerTelefon
Methodenname	deleteNutzerTelefon
Aufruf mit HTTP-Methode	DELETE
Methodenkopf	**public** String deleteNutzerTelefon(String data)
Parameter	UUID von Typ String im JSON Format
Rückgabetyp	Bestätigung über Ausführung als String mit Wert „ausgeführt“
Beschreibung	Der Service löscht den Eintrag in der CF NutzerTelefon, der der erhaltenen UUID entsprichrt.
Verwendete Java DBKlassen	DBNutzerTelefon
Verwendete Coulumn Families	NutzerNutzerTelefon

REST Service Sensor getSensorListByNutStaID

URL	http://babeauf.nt.fh-koeln.de:8080

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

	/SensorCloudRest/crud/Sensor/NutStaID/c24d959d-9aa1-47e1-92b4-4ebdd89583f0
Java Klasse des Services	HttpSensor
Methodenname	getSensorListByNutStaID
Aufruf mit HTTP-Methode	GET
Methodenkopf	**public** String getSensorListByNutStaID(@PathParam("nutStaID") String nutStaID)
Parameter	PathParameter mit NutStaID der CF[1] NutzerStammdaten
Rückgabetyp	Objekt von der Java Klasse SensorList im JSON Format
Beschreibung	Der Service liefert eine Liste von Sensoren welche einem Nutzer angehören. Ein Sensor gehört immer nur einem Nutzer. Über die NutStaID ist die Person zu identifizieren.
Verwendete DBKlassen	DBSensor
Verwendete Column Families	Sensor

REST Service Sensor getSensorListBySenSerID

URL	http://babeauf.nt.fh-koeln.de:8080 /SensorCloudRest/crud/Sensor/SenSerID/734d959d-9aa1-47e1-92b4-febdd89583f0
Java Klasse des Services	HttpSensor
Methodenname	getSensorListBySenSerID
Aufruf mit HTTP-Methode	GET
Methodenkopf	**public** String getSensorListBySenSerID(@PathParam("senSerID") String senSerID)
Parameter	PathParameter mit SenSerID der CF SensorService
Rückgabetyp	Objekt von der Java Klasse SensorList im JSON Format
Beschreibung	Der Web Service liefert für einen SensorService die Liste aller Sensoren, die diesem Service angehören. Sensoren von verschiedenen Herstellern oder mit unterschiedlichen Konfigurationen können trotzdem den selben Service

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

	anbieten.
Verwendete DBKlassen	DBSensor, DBSensorServiceMitglieder
Verwendete Coulumn Families	SensorServiceMitglieder, Sensor

REST Service SensorProdukt getSensorProduktBySenProID

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/SensorProdukt/ SenProID/79bf3a3d-bdc6-4d26-b575-854178557d36
Java Klasse des Services	HttpSensorProdukt
Methodenname	getSensorProduktBySenProID
Aufruf mit HTTP-Methode	GET
Methodenkopf	**public** String getSensorProduktBySenProID(@PathParam("senProID") String senProID)
Parameter	PathParameter mit SenProID der CF SensorProdukt
Rückgabetyp	Objekt von der Java Klasse SensorProduktSemantik im JSON Format
Beschreibung	Der Web Service liefert für einen Sensor, dessen Semantik. So dort Angaben enthalten über: Messwertbereich, Einheit des Wertes, Umrechnung des Wertes in eine Dezimalzahl usw.
Verwendete Java DBKlassen	DBSensorProdukt
Verwendete Coulumn Families	SensorProdukt

REST Service SensorService getSensorServicesBySenID

URL	http://babeauf.nt.fh-koeln.de:8080 /SensorCloudRest/crud/SensorService/SenID/1690463
Java Klasse des Services	HttpSensorService
Methodenname	getSensorServicesBySenID
Aufruf mit HTTP-Methode	GET

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Methodenkopf	**public** String getSensorServicesBySenID(@PathParam("senID") String senID)
Parameter	PathParameter mit SenID der CF Sensor
Rückgabetyp	Objekt von der Klasse SensorServiceList im JSON Format
Beschreibung	Der Service liefert eine Liste aller „Sensor“-Services, die dem angegebenen Sensor angehören. Der „Sensor“-Service beschreibt, was der Sensor für eine Anwendung anbietet oder in welchem Zusammenhang dieser mit Umwelt kooperiert.
Verwendete DBKlassen	DBSensorServiceMitglieder, DBSensorService
Verwendete Coulumn Families	SensorServiceMitglieder, SensorService

REST Service SensorService getSensorServicesBySerLinID

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/SensorService/ SerLinID/2db3c3c0-88f0-4fe1-a7d0-0cf26defabaa
Java Klasse des Services	HttpSensorService
Methodenname	getSensorServicesBySerLinID
Aufruf mit HTTP-Methode	GET
Methodenkopf	**public** String getServicesBySerLinID(@PathParam("serLinID") String serLinID)
Parameter	PathParameter mit SerLinID der CF ServiceLinie
Rückgabetyp	Objekt von der Klasse ServiceMitTypList im JSON Format
Beschreibung	Der Web Service liefert eine Liste aller „Sensor/Aktor“-Services, die sich in einer Servicelinie befinden. Unter der Servicelinie können nämlich Services von Aktoren und Sensoren zusammengefasst werden.
Verwendete DBKlassen	DBServiceLinienMitglieder, DBSensorService, DBAktorService
Verwendete Coulumn Families	ServiceLinienMitglieder, SensorService, AktorService

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

REST Service SensorServiceFunktion getFunktionListBySerSerID

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/SensorServiceFunktion/ SerSerID/25405030-cdb8-487b-84f4-84692c0ed9eb
Java Klasse des Services	HttpSensorServiceFunktion
Methodenname	getFunktionListBySerSerID
Aufruf mit HTTP-Methode	GET
Methodenkopf	**public** String getFunktionListBySenSerID(@PathParam("senSerID") String senSerID)
Parameter	PathParameter mit SerSerID der CF SensorService
Rückgabetyp	Objekt von der Java Klasse SensorServiceMitFunktionList im JSON Format
Beschreibung	Der Web Service liefert eine Liste der Funktionen eines „Sensor“-Services.
Verwendete DBKlassen	DBSensorServiceFunktionMitglieder, DBSensorServiceFunktion
Verwendete Coulumn Families	SensorServiceFunktionMitglieder, SensorServiceFunktion

REST Service SensorVerbund getSensorVerbundByNutStaID

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/SensorVerbund/ NutStaID/c24d959d-9aa1-47e1-92b4-4ebdd89583f0
Java Klasse des Services	HttpSensorVerbund
Methodenname	getSensorVerbundByNutStaID
Aufruf mit HTTP-Methode	GET
Methodenkopf	**public** String getSensorVerbundByNutStaID(@PathParam("nutStaID") String nutStaID)
Parameter	PathParameter mit NutStaID der CF NutzerStammdaten

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Rückgabetyp	Objekt von der Java Klasse SensorVerbundList im JSON Format
Beschreibung	Der Web Service liefert eine Liste der „Sensor“-Verbünde, einer Person.
Verwendete DBKlassen	DBSensor, DBSensorVerbundMitglieder, DBSensorVerbund
Verwendete Coulumn Families	Sensor, SensorVerbundMitglieder, SensorVerbund

REST Service SensorVerbund getSensorBySenVerID

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/SensorVerbund/ SenVerID/8e0f49ea-c5c5-4e20-a5af-0d22093bafb0
Java Klasse des Services	HttpSensorVerbund
Methodenname	getSensorBySenVerD
Aufruf mit HTTP-Methode	GET
Methodenkopf	**public** String getSensorBySenVerID(@PathParam("senVerID") String senVerID)
Parameter	PathParameter mit SenVerID der CF SensorVerbund
Rückgabetyp	Objekt von der Java Klasse SensorList im JSON Format
Beschreibung	Der Web Service liefert eine Liste der Sensoren, die in einem gesuchten Verbund zusammenfasst sind.
Verwendete DBKlassen	DBSensor, DBSensorVerbundMitglieder
Verwendete Coulumn Families	Sensor, SensorVerbundMitglieder

REST Service SensorVerbund createSensorVerbund

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/SensorVerbund
Java Klasse des Services	HttpSensorVerbund
Methodenname	createSensorVerbund

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Aufruf mit HTTP-Methode	PUT
Methodenkopf	**public** String createSensorVerbund(String data)
Parameter	Objekt von der Java Klasse SensorVerbundMitSensor im JSON Format
Rückgabetyp	UUID als String des neuen Verbundes
Beschreibung	Der Service erstellt einen neue Eintrag in SensorVerbund. Des weiteren verbindet der den Sensor, aus den empfangenen Daten, mit dem neuen Verbund. Dies geschieht durch den Eintag in SensorVerbundMitglieder.
Verwendete DBKlassen	DBSensorVerbund, DBSensorVerbundMitglieder
Verwendete Coulumn Families	SensorVerbund, SensorVerbundMitglieder

REST Service SensorVerbund addSensorToSensorVerbund

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/SensorVerbund
Java Klasse des Services	HttpSensorVerbund
Methodenname	addSensorToSensorVerbund
Aufruf mit HTTP-Methode	POST
Methodenkopf	**public** String addSensorToSensorVerbund(String data)
Parameter	Objekt von der Java Klasse SensorVerbundMitSensor im JSON Format
Rückgabetyp	String mit Wert „ausgeführt“, wenn der Eintrag gemacht worden ist.
Beschreibung	Der Service fügt einem „Sensor“-Verbund einen Sensor hinzu. Dies geschieht durch den Eintag in SensorVerbundMitglieder.
Verwendete DBKlassen	DBSensorVerbundMitglieder
Verwendete Coulumn Families	SensorVerbundMitglieder

REST Service ServiceLinien getServiceLinienByNutStaID

URL	http://babeauf.nt.fh-koeln.de:8080/

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

	SensorCloudRest/crud/ServiceLinien/ NutStaID/c24d959d-9aa1-47e1-92b4-4ebdd89583f0
Java Klasse des Services	HttpServiceLinien
Methodenname	getServiceLinienByNutStaID
Aufruf mit HTTP-Methode	GET
Methodenkopf	**public** String getServiceLinienByNutStaID(@PathParam("nutStaID") String nutStaID)
Parameter	PathParameter mit NutStaID der CF NutzerStammdaten
Rückgabetyp	Objekt von der Java Klasse ServiceLinienList im JSON Format
Beschreibung	Der Web Service liefert eine Liste der Servicelinien, die der Person gehören.
Verwendete DBKlassen	DBServiceLinien
Verwendete Coulumn Families	ServiceLinien

REST Service ServiceLinien createServiceLinien

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/ServiceLinien
Java Klasse des Services	HttpServiceLinien
Methodenname	createServiceLinien
Aufruf mit HTTP-Methode	PUT
Methodenkopf	**public** String createServiceLinien(String data)
Parameter	Objekt von der Java Klasse SensorServiceMitServiceLinie im JSON Format
Rückgabetyp	UUID als String der neuen Servicelinie
Beschreibung	Der Service erstellt einen neue Eintrag in ServiceLinien. Des weiteren werden „Sensor“-Service, mit der neue erstellten Serviclinie verbunden. Dies geschieht durch den Eintag in ServiceLinienMitglieder.

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Verwendete DBKlassen	DBServiceLinien, DBServiceLinienMitglieder
Verwendete Coulumn Families	ServiceLinien, ServiceLinienMitglieder

REST Service ServiceLinien createServiceLinienAktor

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/ServiceLinien/Aktor
Java Klasse des Services	HttpServiceLinien
Methodenname	createServiceLinienAktor
Aufruf mit HTTP-Methode	PUT
Methodenkopf	**public** String createServiceLinienAktor(String data)
Parameter	Objekt von der Java Klasse AktorServiceMitServiceLinie im JSON Format
Rückgabetyp	UUID als String der neuen Servicelinie
Beschreibung	Der Service erstellt einen neue Eintrag in ServiceLinien. Des weiteren werden „Aktor“-Service, mit der neue erstellten Serviclinie verbunden. Dies geschieht durch den Eintag in ServiceLinienMitglieder.
Verwendete DBKlassen	DBServiceLinien, DBServiceLinienMitglieder
Verwendete Coulumn Families	ServiceLinien, ServiceLinienMitglieder

REST Service ServiceLinien AddServiceToServiceLinien

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/ServiceLinien
Java Klasse des Services	HttpServiceLinien
Methodenname	addServiceToServiceLinien
Aufruf mit HTTP-Methode	POST
Methodenkopf	**public** String addServiceToServiceLinien(String data)
Parameter	Objekt von der Java Klasse SensorServiceMitServiceLinie im JSON Format

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Rückgabetyp	UUID als String mit Wert „ausgefuehrt“ wenn Service durchlaufen ist.
Beschreibung	Der Service verbindet einen „Sensor“-Service mit einer ServiceLinien. Dies geschieht durch den Eintag in ServiceLinienMitglieder.
Verwendete DBKlassen	DBServiceLinienMitglieder
Verwendete Coulumn Families	ServiceLinienMitglieder

REST Service ServiceLinien addServiceToServiceLinienAktor

URL	http://babeauf.nt.fh-koeln.de:8080/ SensorCloudRest/crud/ServiceLinien/Aktor
Java Klasse des Services	HttpServiceLinien
Methodenname	addServiceToServiceLinienAktor
Aufruf mit HTTP-Methode	POST
Methodenkopf	**public** String addServiceToServiceLinienAktor(String data)
Parameter	Objekt von der Java KlasseAktorServiceMitServiceLinie im JSON Format
Rückgabetyp	UUID als String mit Wert „ausgefuehrt“ wenn Service durchlaufen ist.
Beschreibung	Der Service verbindet einen „Aktor“-Service mit einer ServiceLinien. Dies geschieht durch den Eintag in ServiceLinienMitglieder.
Verwendete DBKlassen	DBServiceLinienMitglieder
Verwendete Coulumn Families	ServiceLinienMitglieder

1ColumnFamily ; ähnlich der Tabelle in SQL Datenbanken

Folgende REST-Services sind über http://babeauf.nt.fh-koeln.de:8080/REST erreichbar und liefern TEXT-Datensätze als Rückgabe.

REST Service HttpGet getSensorListe

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ get/sensor
Java Klasse des Services	HttpGet
Methodenname	getSensorListe
Aufruf mit HTTP-Methode	GET
Methoden-Kopf	getSensorListe()
Parameter	keiner
Rückgabetyp	Eine ArrayList vom Typ String im JSON Format.
Beschreibung	Liefert eine Liste mit allen Wetter-API SenID´s.
Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	Adresse, NutzerStammdaten, Sensor

REST Service HttpGet getPlzBySenID

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ get/plz/ eb219be2-cf92-4eec-80b8-b35f3931b4d6
Java Klasse des Services	HttpGet
Methodenname	getPlzBySenID
Aufruf mit HTTP-Methode	GET
Methoden-Kopf	getPlzBySenID(@PathParam("SenID") String SenID)
Parameter	SenID vom Typ String.
Rückgabetyp	Ein leerer String wenn kein Eintrag vorhanden ist, ansosten die Postleitzahl als String.
Beschreibung	Liefert die Postleitzahl eines Sensors.
Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	Adresse, NutzerStammdaten, Sensor

Gefördert durch:

Bundesministerium für Wirtschaft und Technologie

aufgrund eines Beschlusses des Deutschen Bundestages

REST Service HttpGet getEventListeBySenID

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ get/eventListe/ eb219be2-cf92-4eec-80b8-b35f3931b4d6
Java Klasse des Services	HttpGet
Methodenname	getEventListeBySenID
Aufruf mit HTTP-Methode	GET
Methoden-Kopf	getEventListeBySenID(@PathParam("SenID") String SenID)
Parameter	SenID vom Typ String.
Rückgabetyp	Eine ArrayList vom Typ String im JSON Format.
Beschreibung	Liefert alle EveID´s eines Sensors.
Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	Event, SensorEvent, EventMitglieder

REST Service HttpGet getAusgelösteEventsListeByEveIDAndWetterdaten

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ get/ausgelösteEvents/ {EveIDListe}/{deltaWetterdaten}
Java Klasse des Services	HttpGet
Methodenname	getAusgeloesteEventsListeBy EveIDAndWetterdaten
Aufruf mit HTTP-Methode	GET
Methoden-Kopf	getAusgelösteEventsListeBy EveIDAndWetterdaten(@PathParam("EveIDListe") String EveIDListe, @PathParam("deltaWetterdaten") String deltaWetterdaten)

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Parameter	EveIDListe als ArrayList vom Typ String als JSON String und ein Objekt vom Typ Wetterdaten als JSON String.
Rückgabetyp	Ein leerer String wenn kein Eintrag vorhanden ist, ansosten ein Objekt vom Typ Unwetter als JSON String.
Beschreibung	Liefert ein Unwetterobjekt, mit Informationen, ob und wann ein Unwetter eingetroffen ist.
Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	Event, EventMitglieder, SensorEvent

REST Service HttpGet getOrkanEveBySenID

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ get/orkanEvent/ eb219be2-cf92-4eec-80b8-b35f3931b4d6
Java Klasse des Services	HttpGet
Methodenname	getOrkanEveBySenID
Aufruf mit HTTP-Methode	GET
Methoden-Kopf	getOrkanEveBySenID(@PathParam("SenID") String SenID)
Parameter	SenID vom Typ String.
Rückgabetyp	Ein Objekt vom Typ Event als JSON String.
Beschreibung	Liefert ein Event Objekt mit der Eventbezeichung "Orkan" zu einem Wetter-API Sensor.
Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	Event, SensorEvent, EventMitglieder

REST Service HttpGet getSturmEveBySenID

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ get/sturmEvent/ eb219be2-cf92-4eec-80b8-b35f3931b4d6
Java Klasse des Services	HttpGet
Methodenname	getSturmEveBySenID
Aufruf mit HTTP-Methode	GET
Methoden-Kopf	getSturmEveBySenID(@PathParam("SenID") String SenID)
Parameter	SenID vom Typ String.
Rückgabetyp	Ein Objekt vom Typ Event als JSON String.
Beschreibung	Liefert ein Event Objekt mit der Eventbezeichung "Sturm" zu einem Wetter-API Sensor.
Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	Event, SensorEvent, EventMitglieder

REST Service HttpGet getRegenEveBySenID

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ get/regenEvent/ eb219be2-cf92-4eec-80b8-b35f3931b4d6
Java Klasse des Services	HttpGet
Methodenname	getRegenEveBySenID
Aufruf mit HTTP-Methode	GET
Methoden-Kopf	getRegenEveBySenID(@PathParam("SenID") String SenID)
Parameter	SenID vom Typ String.
Rückgabetyp	Ein Objekt vom Typ Event als JSON String.

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Beschreibung	Liefert ein Event Objekt mit der Eventbezeichung "Regen" zu einem Wetter-API Sensor.
Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	Event, SensorEvent, EventMitglieder

REST Service HttpGet getSchneeEveBySenID

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ get/schneeEvent/ eb219be2-cf92-4eec-80b8-b35f3931b4d6
Java Klasse des Services	HttpGet
Methodenname	getSchneeEveBySenID
Aufruf mit HTTP-Methode	GET
Methoden-Kopf	getSchneeEveBySenID(@PathParam("SenID") String SenID)
Parameter	SenID vom Typ String.
Rückgabetyp	Ein Objekt vom Typ Event als JSON String.
Beschreibung	Liefert ein Event Objekt mit der Eventbezeichung "Schnee" zu einem Wetter-API Sensor.
Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	Event, SensorEvent, EventMitglieder

REST Service HttpGet getEventBenachrichtigungByEveID

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ get/eventBenachrichtigung/ b7946633-1450-4416-9e5f-702414f94d5a
Java Klasse des Services	HttpGet

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Methodenname	getEventBenachrichtigungBy EveID
Aufruf mit HTTP-Methode	GET
Methoden-Kopf	getEventBenachrichtigungBy EveID(@PathParam("EveID") String EveID)
Parameter	EveID vom Typ String.
Rückgabetyp	Ein leerer String wenn kein Eintrag vorhanden ist, ansosten der EveBenWeg als String.
Beschreibung	Liefert den EveBenWeg zu einer EveID.
Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	EventBenachrichtigung

REST Service HttpGet getEventAktionListeByEveID

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ get/eventAktionListe/ b7946633-1450-4416-9e5f-702414f94d5a
Java Klasse des Services	HttpGet
Methodenname	getEventAktionListeByEveID
Aufruf mit HTTP-Methode	GET
Methoden-Kopf	getEventAktionListeByEveID(@PathParam("EveID") String EveID)
Parameter	EveID vom Typ String.
Rückgabetyp	Eine ArrayList vom Typ String im JSON Format. Der String ist ein geparstes Objekt vom Typ EventAktion im JSON Format.
Beschreibung	Liefert alle EventAktionen zu einer EveID.
Verwendete DBKlassen	Cassandra, CRUD

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Verwendete Column Families	EventAktion

REST Service HttpGet getAktorAktIDByEveAktiZieID

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ get/aktorAktID/ 053404ac-5780-4e11-8dbc-3a234d1da6d0
Java Klasse des Services	HttpGet
Methodenname	getAktorAktIDByEveAktiZieID
Aufruf mit HTTP-Methode	GET
Methoden-Kopf	getAktorAktIDByEveAktiZieID(@PathParam("EveAktiZieID") String EveAktiZieID)
Parameter	EveAktiZieID vom Typ String.
Rückgabetyp	AktAktProID von Typ String.
Beschreibung	Liefert AktAktProID zu einer EveAktiZieID.
Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	Aktor

REST Service HttpGet getNutzerNameBySenID

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ get/nutzerName/ eb219be2-cf92-4eec-80b8-b35f3931b4d6
Java Klasse des Services	HttpGet
Methodenname	getNutzerNameBySenID
Aufruf mit HTTP-Methode	GET
Methoden-Kopf	getNutzerNameBySenID(@PathParam("SenID") String SenID)
Parameter	SenID vom Typ String.

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Rückgabetyp	Ein leerer String wenn kein Eintrag vorhanden ist, ansosten der Nutzer Vor- und Nachname als String.
Beschreibung	Liefert Nutzer Vor- und Nachname eines Sensors.
Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	Sensor, NutzerStammdaten

REST Service HttpGet getAllAktorAnforderung

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ get/aktorAnforderung
Java Klasse des Services	HttpGet
Methodenname	getAllAktorAnforderung
Aufruf mit HTTP-Methode	GET
Methoden-Kopf	getAllAktorAnforderung()
Parameter	keinen
Rückgabetyp	Eine ArrayList vom Typ String im JSON Format. Der String ist ein geparstes Objekt vom Typ AktorAnforderung im JSON Format.
Beschreibung	Liefert alle Einträge in der Entität AktorAnforderung ohne "null" Einträge.
Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	AktorAnforderung

REST Service HttpGet getAllAktorVerMitAktIDByEveAktiZieID

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ get/aktorVerbundMitglieder/ 0b5a047c-fe05-4f2b-88b0-6d71fa9905c2
Java Klasse des Services	HttpGet
Methodenname	getAllAktorVerMitAktIDByEveAktiZieID

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Aufruf mit HTTP-Methode	GET
Methoden-Kopf	getAllAktorVerMitAktIDByEveAktiZieID(@PathParam("EveAktiZieID") String EveAktiZieID)
Parameter	EveAktiZieID vom Typ String.
Rückgabetyp	Ein leerer String wenn kein Eintrag vorhanden ist, ansosten die AktVerMitAktID´s als ArrayList von Typ String im JSON Format.
Beschreibung	Liefert alle AktVerMitAktID´s anhand EveAktiZieID.
Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	AktorVerbundMitglieder

REST Service HttpGet getAllUserByPLZ

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ get/benutzer/50674
Java Klasse des Services	HttpGet
Methodenname	getAllUserByPLZ
Aufruf mit HTTP-Methode	GET
Methoden-Kopf	getAllUserByPLZ(@PathParam("plz") String plz)
Parameter	plz vom Typ String.
Rückgabetyp	Ein String mit den Informationen Vorname, Nachname, Strasse und Ort aller Nutzer der Wetter-API.
Beschreibung	Liefert alle Benutzer der Wetter-API (NutStaVor, NutStaNam, NutStaID) anhand der Postleitzahl.
Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	Adresse, NutzerStammdaten, Sensor

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

REST Service HttpGet getAllNutzerAdresseSensor

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ get/allNutzer
Java Klasse des Services	HttpGet
Methodenname	getAllNutzerAdresseSensor
Aufruf mit HTTP-Methode	GET
Methoden-Kopf	getAllNutzerAdresseSensor()
Parameter	keiner
Rückgabetyp	Eine ArrayList vom Typ String im JSON Format. Der String ist ein geparstes Objekt vom Typ APINutzer im JSON Format.
Beschreibung	Liefert ein APINutzer Objekt mit allen Nutzern des SensorCloud Systems.
Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	Adresse, NutzerStammdaten, Sensor

REST Service HttpGet getAktorInformationenByNutStaID

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ get/aktorInformationen/ 843f7a3c-f114-437c-a992-1bf846e6da0c
Java Klasse des Services	HttpGet
Methodenname	getAktorInformationenByNut StaID
Aufruf mit HTTP-Methode	GET
Methoden-Kopf	getAktorInformationenByNut StaID(@PathParam("NutStaID") String NutStaID)

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Parameter	NutStaID vom Typ String.
Rückgabetyp	Eine ArrayList vom Typ String im JSON Format. Der String ist ein geparstes Objekt vom Typ AktorInformationen im JSON Format.
Beschreibung	Liefert alle Aktorinformationen anhand der NutStaID.
Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	Aktor, AktorProdukt

REST Service HttpGet getSensorIDByNutStaID

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ get/senID/ 843f7a3c-f114-437c-a992-1bf846e6da0c
Java Klasse des Services	HttpGet
Methodenname	getSensorIDByNutStaID
Aufruf mit HTTP-Methode	GET
Methoden-Kopf	getSensorIDByNutStaID(@PathParam("NutStaID") String NutStaID)
Parameter	NutStaID vom Typ String.
Rückgabetyp	Ein leerer String wenn kein Eintrag vorhanden ist, ansosten die SenID als String.
Beschreibung	Liefert die SenID anhand der NutStaID.
Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	Sensor

REST Service HttpPut putAktorAnforderungIntoAktorAnforderung

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ put/aktorAnforderung/{aktAnf}

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Java Klasse des Services	HttpPut
Methodenname	putAktorAnforderungIntoAktor Anforderung
Aufruf mit HTTP-Methode	PUT
Methoden-Kopf	putAktorAnforderungIntoAktor Anforderung(@PathParam("aktAnf") String aktorAnforderungString)
Parameter	Ein Objekt der Klasse AktorAnforderung als JSON String.
Rückgabetyp	keiner
Beschreibung	Macht einen Eintrag in der Entität AktorAnforderung.
Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	AktorAnforderung

REST Service HttpPut putSensorIntoSensor

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ put/sensor/{NutStaID}
Java Klasse des Services	HttpPut
Methodenname	putSensorIntoSensor
Aufruf mit HTTP-Methode	PUT
Methoden-Kopf	putSensorIntoSensor(@PathParam("NutStaID") String NutStaID)
Parameter	NutStaID vom Typ String.
Rückgabetyp	keiner
Beschreibung	Legt einen Wetter-API Sensor mit der übergebenen NutStaID in der Entität Sensor an.

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	Sensor

REST Service HttpPut putSensorEventOrkanIntoSensorEvent

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ put/sensorEventOrkan/{SenID}/{NutEmaAdr}
Java Klasse des Services	HttpPut
Methodenname	putSensorEventOrkanIntoSensorEvent
Aufruf mit HTTP-Methode	PUT
Methoden-Kopf	putSensorEventOrkanIntoSensorEvent(@PathParam("SenID") String SenID, @PathParam("NutEmaAdr") String NutEmaAdr)
Parameter	SenID vom Typ String und NutEmaAdr vom Typ String.
Rückgabetyp	EveID vom Typ String.
Beschreibung	Mach die notwendigen Einträge in der Entität SensorEvent für das Event Orkan. Außerdem dem Eintrag in der Entität EventMitglieder und EventBenachrichtigung.
Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	SensorEvent

REST Service HttpPut putSensorEventSturmIntoSensorEvent

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ put/sensorEventSturm/{SenID}/{NutEmaAdr}
Java Klasse des Services	HttpPut
Methodenname	putSensorEventSturmIntoSensorEvent
Aufruf mit HTTP-Methode	PUT
Methoden-Kopf	putSensorEventSturmIntoSensorEvent(

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

	@PathParam("SenID") String SenID, @PathParam("NutEmaAdr") String NutEmaAdr)
Parameter	SenID vom Typ String und NutEmaAdr vom Typ String.
Rückgabetyp	EveID vom Typ String.
Beschreibung	Mach die notwendigen Einträge in der Entität SensorEvent für das Event Sturm. Außerdem dem Eintrag in der Entität EventMitglieder und EventBenachrichtigung.
Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	SensorEvent

REST Service HttpPut putSensorEventRegenIntoSensorEvent

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ put/sensorEventRegen/{SenID}/{NutEmaAdr}
Java Klasse des Services	HttpPut
Methodenname	putSensorEventRegenIntoSensorEvent
Aufruf mit HTTP-Methode	PUT
Methoden-Kopf	putSensorEventRegenInto SensorEvent(@PathParam("SenID") String SenID,@PathParam("NutEmaAdr") String NutEmaAdr)
Parameter	SenID vom Typ String und NutEmaAdr vom Typ String.
Rückgabetyp	EveID vom Typ String.
Beschreibung	Mach die notwendigen Einträge in der Entität SensorEvent für das Event Regen. Außerdem dem Eintrag in der Entität EventMitglieder und EventBenachrichtigung.
Verwendete DBKlassen	Cassandra, CRUD

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

Verwendete Column Families	SensorEvent

REST Service HttpPut putSensorEventSchneeIntoSensorEvent

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ put/sensorEventSchnee/{SenID}/{NutEmaAdr}
Java Klasse des Services	HttpPut
Methodenname	putSensorEventSchneeIntoSensorEvent
Aufruf mit HTTP-Methode	PUT
Methoden-Kopf	putSensorEventSchneeInto SensorEvent(@PathParam("SenID") String SenID,@PathParam("NutEmaAdr") String NutEmaAdr)
Parameter	SenID vom Typ String und NutEmaAdr vom Typ String.
Rückgabetyp	EveID vom Typ String.
Beschreibung	Mach die notwendigen Einträge in der Entität SensorEvent für das Event Schnee. Außerdem dem Eintrag in der Entität EventMitglieder und EventBenachrichtigung.
Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	SensorEvent

REST Service HttpPut putEventAktionIntoEventAktion

URL	http://babeauf.nt.fh-koeln.de:8080/REST/ put/EventAktion/{SenID}
Java Klasse des Services	HttpPut
Methodenname	putEventAktionIntoEventAktion
Aufruf mit HTTP-Methode	PUT
Methoden-Kopf	putEventAktionIntoEventAktion(@PathParam("EveAktiEveID")

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

	String EveAktiEveID, @PathParam("eventAktion") String eveAktion)
Parameter	EveAktiEveID vom Typ String und ein Objekt von Typ EventAktion als JSON String.
Rückgabetyp	keiner
Beschreibung	Macht einen Eintrag in der Entität EventAktion mit den übergebenen Informationen.
Verwendete DBKlassen	Cassandra, CRUD
Verwendete Column Families	EventAktion

Folgende REST-Services sind über http://babeauf.nt.fh-koeln.de:8080/SensorCloud/db/ erreichbar und liefern JSON-Datensätze als Rückgabe.

REST Service SensorProduktSemantik getSensorSemantik

URL	http://babeauf.nt.fh-koeln.de:8080 /SensorCloud/db/SensorSemantik/SenID/{SenID}
Java Klasse des Services	HttpSensorSemantik
Methodenname	getSensorSemantik
Aufruf mit HTTP-Methode	GET
Methoden-Kopf	getSensorSemantik(@PathParam("senID") String senID)
Parameter	UUID von Typ String im JSON Format
Rückgabetyp	Objekt von der Java Klasse SensorProduktSemantik im JSON Format
Beschreibung	Der Service liefert ein Objekt der Java Klasse SensorSemantik für die angegebene SenID. Über die SenID wird die SenProID ermittelt und deren Semantik zurück gegeben.
Verwendete DBKlassen	DBSensorSematik
Verwendete Column Families	Sensor, SensorProdukt

Gefördert durch:
Bundesministerium für Wirtschaft und Technologie
aufgrund eines Beschlusses des Deutschen Bundestages

REST Service SensorProduktSemantik getSensorSemantikAll

URL	http://babeauf.nt.fh-koeln.de:8080 /SensorCloud/db/SensorSemantik/All
Java Klasse des Services	HttpSensorSemantik
Methodenname	getSensorSemantikAll
Aufruf mit HTTP-Methode	GET
Methoden-Kopf	getSensorSemantik()
Parameter	/
Rückgabetyp	Liste von der Java Klasse SensorProduktSemantik im JSON Format
Beschreibung	Der Service liefert eine Liste der Java Klasse SensorSemantik für alle vorhandenen SensorProduktSemantiken.
Verwendete DBKlassen	DBSensorSematik
Verwendete Column Families	SensorProdukt

Rückgaben der obigen Webservices

Beispiel JSON-Datensatz


```
{
   "list":
       [
            {
                "nutEmaID":"edfb2bf8-84dd-4114-aeb6-b139907170fa",
                "nutEmaNutStaID":"c24d959d-9aa1-47e1-92b4-4ebdd89583f0",
                "nutEmaAdr":"xyz@smail.fh-koeln.de",
                "nutEmaBez":"office"
            },
            {
                "nutEmaID":"350eae63-a2d9-487e-a694-3f62f03c6f06",
                "nutEmaNutStaID":"c24d959d-9aa1-47e1-92b4-4ebdd89583f0",
                "nutEmaAdr":"xyz@domain.tld",
                "nutEmaBez":"privat"
            }
```


Gefördert durch:

Bundesministerium für Wirtschaft und Technologie

aufgrund eines Beschlusses des Deutschen Bundestages

```
        ]
}
```

Skripting 1: Beispiel JSON-Datensatz

Beispiel Text-Datensatz

```
["47bb6fbc-bb9d-4213-948b-2780d9712c88","6829b7f3-6bd1-4e49-bde6-1b674bb6a148"]
```

Skripting 2: Beispiel Text-Datensatz

9. Diskussion

RESTful-Services eignen sich aufgrund des Aufsetzens auf HTTP und der dadurch geerbten Eigenschaften und dessen guten horizontalen Skalierbarkeit sehr gut für das Projekt SensorCloud. Die abstrahierte Sicht auf die Datenbank vereinfacht den Entwicklern von Sensordiensten sich die benötigten Daten, ohne Wissen der eingesetzten Datenbanken und des Datenbankschemas, zu selektieren und zu verändern. Zudem vereinfacht es die Pflege von implementierten Sensordiensten bei Änderungen an dem föderierten Datenbanksystems und dessen Datenbankschema.

Aktuell greifen die RESTful-Services nativ auf die Datenbank zu. Um die Implementierung der RESTful-Services zusätzlich von den Änderungen an den eingesetzten Datenbanken zu kapseln kann über den Einsatz von JavaBeans nachgedacht werden.

Quellen

[1] H. Budde et al.: *Konzeptionelles Schema*. DBAP4, Fachhochschule Köln, Gruppe FDBS, Köln, 2013.

[2] R. Fielding, et al.: *9 Method Definitions.* In: *RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1.* 2004, http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html, (Aufruf: 10.12.2013)

[3] R. Fielding: *Architectural Styles and the Design of Network-based Software Architectures*. Doctoral dissertation, University of California, Irvine, 2000.

[4] T. Partsch: *SensorCloud Datenmodellierung und Performanceüberlegungen mit einem strukturierten Datenspeicher als Zielsystem*. Master Thesis, Fachhochschule Köln, Köln, 2014.

[5] A. Stec: *Transaktionen in einem föderierten Datenbanksystem der SensorCloud*. Master Thesis, Fachhochschule Köln, Köln, 2013.

[6] T. Dierks, E. Rescorla: *The Transport Layer Security (TLS) Protocol Version 1.2*. RFC 5246, IETF, 2008.

[7] R. Häußling, B. Rumpe, K. Wehrle: *SensorCloud Teilvorhabenbeschreibung RWTH Aachen University*. Förderantrag der RWTH Aachen University zur Einreichung beim Bundesministerium für Wirtschaft und Technologie im Rahmen des Wettbewerbes TrustedCloud, RWTH Aachen University, Aachen, 2011.

[8] H. Budde, A. Lockermann, T. Partsch: *DB Dienste zur Unterstützung des LocationMaster als Trustpoint*. DBAP9, Zwischenbericht, Fachhochschule Köln, Gruppe FDBS, Köln, 2014.

Gefördert durch:

Bundesministerium für Wirtschaft und Technologie

aufgrund eines Beschlusses des Deutschen Bundestages