**Challenge SmartClassroom**

Stand 10.3.2021

**Teil Oracle DB inkl. REST-Schnittstelle aufbauen**

Ziel: Es wird aufgezeigt, wie eine Tabelle für Sensordaten mit der REST Schnittstelle für Schreiben und Lesen von Daten erstellt werden kann.

**Die Cloud DB und das Schemas:**

Es wurde eine Oracle Cloud DB erstellt auf der ein User für diese Challenge erstellt wurde.

Um auf die DB zugreifen zu können wird ein Wallet benötigt. Dies wird zur Verfügung gestellt und muss auf dem eigenen Rechner sein.

Es wurde das Schema SENSOR\_DATALAKE (smarTclassrooM2Da) angelegt. In diesem Schema sollen die Daten der Sensoren gespeichert werden. Dieser User ist für die Studierenden vorgesehen.

**Datenstruktur:**

Dazu wurde die Tabelle any\_sensor\_data\_entry mit folgender Struktur angelegt:

Create table any\_sensor\_data\_entry

(entry\_id number,

insert\_time timestamp,

Sensor\_id number,

Sensor\_name varchar2(100),

Sensor\_type varchar2(20),

Value1 number,

Scale1 number,

Unit1 varchar2(10),

Value2 number,

Scale2 number,

Unit2 varchar2(10),

Value3 number,

Scale3 number,

Unit3 varchar2(10),

entry\_document varchar2(4000));

Die Sequenz wird benötigt, um die enty\_id zu vergeben:

CREATE SEQUENCE SEQ\_ENTRY\_ID

**REST Schnittstelle einrichten**

REST Service auf Tabelle einschalten

Auf Tabelle mit rechter Maustaste «Enable Rest Service» wählen.

Enable objet ankreuzen.

Das erzeugt folgenden Code:

DECLARE

PRAGMA AUTONOMOUS\_TRANSACTION;

BEGIN

ORDS.ENABLE\_OBJECT(p\_enabled => TRUE,

p\_schema => 'SENSOR\_DATALAKE',

p\_object => 'ANY\_SENSOR\_DATA\_ENTRY',

p\_object\_type => 'TABLE',

p\_object\_alias => 'any\_sensor\_data\_entry',

p\_auto\_rest\_auth => TRUE);

commit;

END;

Weiter muss ein REST Data Service Module erstellt werden, dazu unter REST Data Services auf Modules mit rechter Maustaste new Module wählen. Diesen den Namen und Prefix sens geben und Publish wählen. Als URI Pattern any\_sensor\_data\_entry angeben.

was folgenden Code ergibt:

BEGIN

ORDS.DEFINE\_MODULE(

p\_module\_name => 'sens',

p\_base\_path => 'sens',

p\_items\_per\_page => 25,

p\_status => 'PUBLISHED',

p\_comments => NULL);

ORDS.DEFINE\_TEMPLATE(

p\_module\_name => 'sens',

p\_pattern => 'any\_sensor\_data\_entry',

p\_priority => 2,

p\_etag\_type => 'HASH',

p\_etag\_query => NULL,

p\_comments => NULL);

COMMIT;

END;

Dann den Handler GET mit SELECT \* FROM any\_sensor\_data\_entry hinzufügen.

Nun können die Daten via REST gelesen werden:

https://glusfqycvwrucp9-db202012181437.adb.eu-zurich-1.oraclecloudapps.com/ords/sensor\_datalake/sens/any\_sensor\_data\_entry

Damit die Daten nur mit Passwort gelesen werden können muss ein Privileg «sens» angelegt werden mit der Rolle «SODA Develper». Bei den Protected Modules muss dann sens gewählt werden.

Dies kann im REST Data Sevice konfiguriert werden oder via Code:

l\_roles(1) := 'SODA Developer';

l\_modules(1) := 'sens';

ORDS.DEFINE\_PRIVILEGE(

p\_privilege\_name => 'sens',

p\_roles => l\_roles,

p\_patterns => l\_patterns,

p\_modules => l\_modules,

p\_label => '',

p\_description => '',

p\_comments => NULL);

**Umsetzung von POST (Insert)**

CREATE OR REPLACE PROCEDURE create\_one\_sensor\_data (

p\_sensor\_id IN any\_sensor\_data\_entry.sensor\_id%TYPE,

p\_sensor\_name IN any\_sensor\_data\_entry.sensor\_name%TYPE,

p\_sensor\_type IN any\_sensor\_data\_entry.sensor\_type%TYPE,

p\_value1 IN any\_sensor\_data\_entry.value1%TYPE,

p\_scale1 IN any\_sensor\_data\_entry.scale1%TYPE,

p\_unit1 IN any\_sensor\_data\_entry.unit1%TYPE,

p\_entry\_document IN any\_sensor\_data\_entry.entry\_document%TYPE

)

AS

BEGIN

INSERT INTO any\_sensor\_data\_entry (entry\_id, insert\_time, sensor\_id, sensor\_name, sensor\_type, value1, scale1, unit1, entry\_document)

VALUES (seq\_entry\_id.nextval, systimestamp, p\_sensor\_id, p\_sensor\_name, p\_sensor\_type, p\_value1, p\_scale1, p\_unit1, p\_entry\_document);

EXCEPTION

WHEN OTHERS THEN

HTP.print(SQLERRM);

END;

BEGIN

ORDS.define\_template(

p\_module\_name => 'sens',

p\_pattern => 'any\_sensor\_data\_entry/');

ORDS.define\_handler(

p\_module\_name => 'sens',

p\_pattern => 'any\_sensor\_data\_entry/',

p\_method => 'POST',

p\_source\_type => ORDS.source\_type\_plsql,

p\_source => 'BEGIN

create\_one\_sensor\_data(

p\_sensor\_id => :sensor\_id,

p\_sensor\_name => :sensor\_name,

p\_sensor\_type => :sensor\_type,

p\_value1 => :value1,

p\_scale1 => :scale1,

p\_unit1 => :unit1

p\_entry\_document => :entry\_document

);

END;',

p\_items\_per\_page => 0);

COMMIT;

END;

/

Beispiel JSON, das zum Schreiben genutzt werden kann:

json='{"sensor\_id":"2","sensor\_name":"test ab web","sensor\_type":"CO2","value1":"817","scale1":"0","unit1":"ppm","entry\_document":"Wenns klappt, ist super"}';