

An OData service to manage employee addresses

What is OData?

- Integration: connecting subsystems to deliver an overarching functionality
- Example: sending master data from Workday to SAP
- “An open data protocol to consume RESTful APIs”
- Translation:
 - Open Data Protocol: standardised creation and consumption of REST APIs
 - API: a set of protocols and procedures that access the data/features of a system
 - REST: a software architectural style: stateless client-server model
- Query using HTTP requests

Human Resources in SAP

- HR data is organised into separate **infotypes** linked by employee number
- Integration requirements are common for HR data, master data stored in a separate system
- Delimit: HR entries should not be deleted. Delimit = expired validity: start date – end date

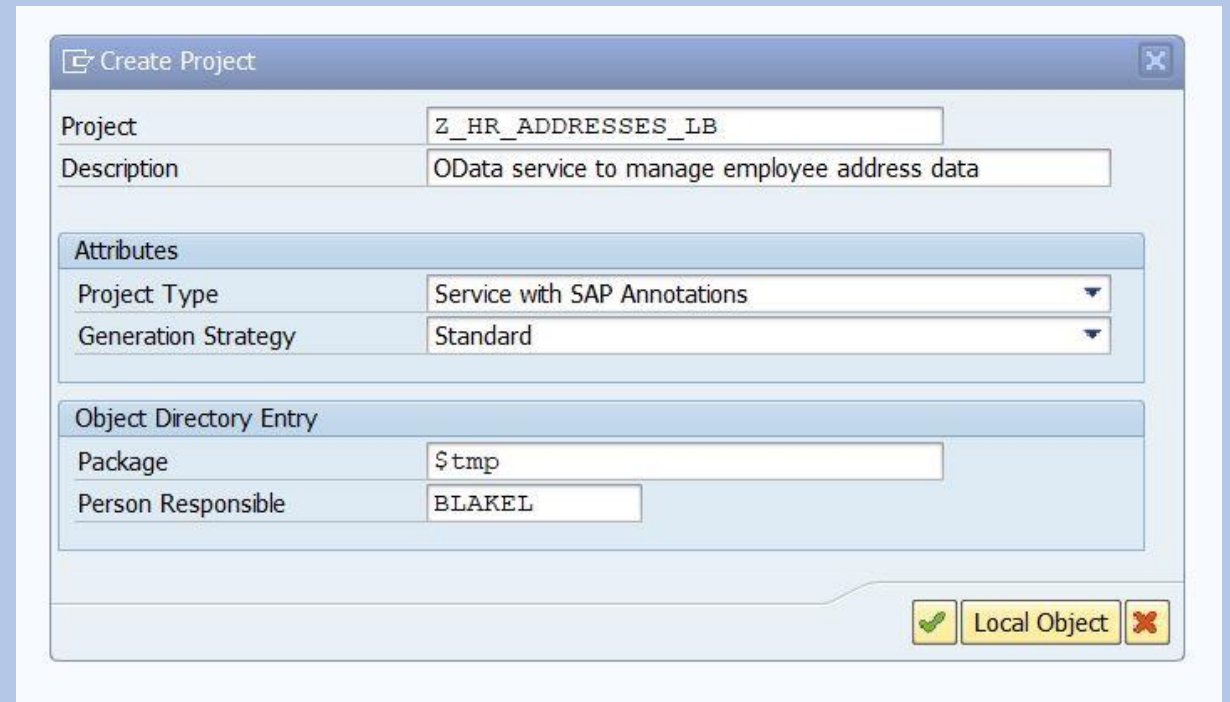
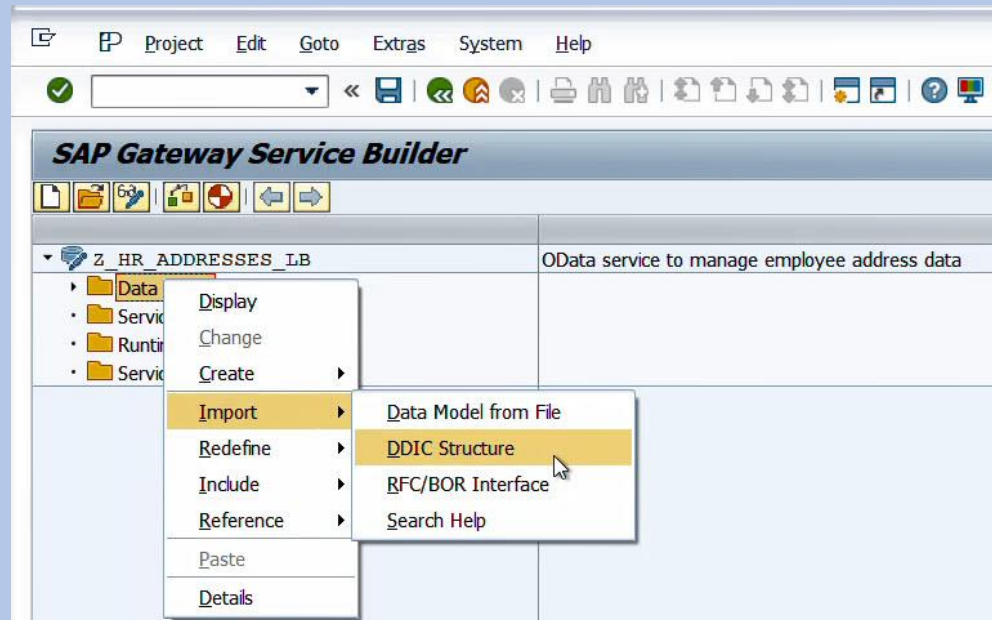
Personnel No	<input type="text" value="16"/>		
Start	<input type="text" value="25.01.2019"/>	to	<input type="text" value="31.12.9999"/>

Infotype text	S..
Actions	
Organizational assignment	
Personal data	
Addresses	
Bank Details	
Family Member/Dependents	
Challenge	

Addresses = PA0006

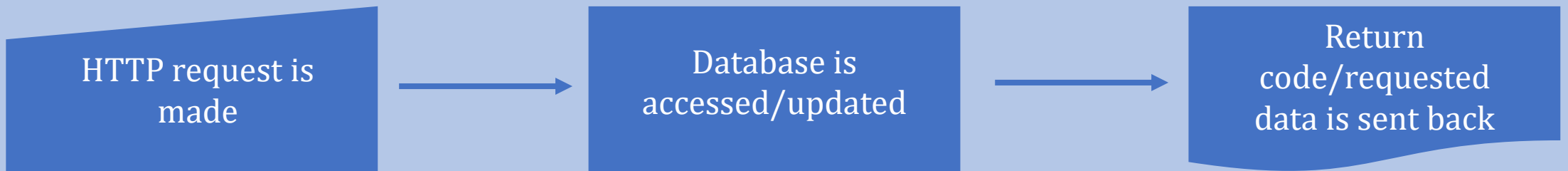
Part 1: Creating a new OData Service

- tcode: SEGW (SAP Gateway Service Builder)
- Create a new project →
- Import data model from DDIC structure ↓



Part 2: Implementations

Create – Read – Update - Delete



Part 2: Implementations: GET_ENTITY

```
Method PA0006SET_GET_ENTITY Active
1 METHOD pa0006set_get_entity.
2 **TRY.
19
20 "data declarations
21 DATA: ls_name_value TYPE LINE OF /iwbsp/t_mgw_name_value_pair,
22        lv_pernr TYPE pa0006-pernr,
23        lt_er_entity TYPE TABLE OF pa0006,
24        lwa_er_entity LIKE LINE OF lt_er_entity,
25        lv_counter TYPE string.
26
27 "get employee number from key value pair
28 READ TABLE it_key_tab INTO ls_name_value
29 WITH KEY name = 'pernr'.
30 IF sy-subrc IS INITIAL.
31     lv_pernr = ls_name_value-value.
32 ENDIF.
33
34 "select all entries for a given employee number into lt
35 SELECT * FROM pa0006
36        INTO CORRESPONDING FIELDS OF TABLE @lt_er_entity
37        WHERE pernr = @lv_pernr.
38
39 "sort lt by end date so the valid entry is the first entry
40 SORT lt_er_entity BY endda DESCENDING.
41
42 "move the first entry from lt to exporting variable
43 LOOP AT lt_er_entity INTO lwa_er_entity.
44     IF lv_counter IS INITIAL.
45         MOVE-CORRESPONDING lwa_er_entity TO er_entity.
46         EXIT.
47     ENDIF.
48     lv_counter = 'X'.
49 ENDLOOP.
50
51 ENDMETHOD.
52
```

Retrieve employee number from http request



SELECT all entries for the number into an internal table



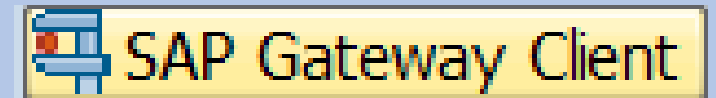
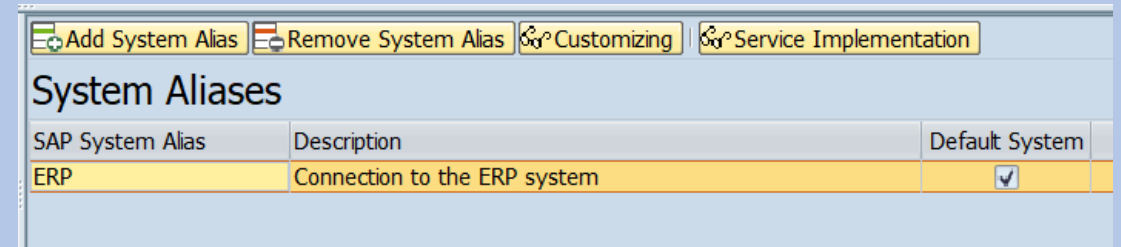
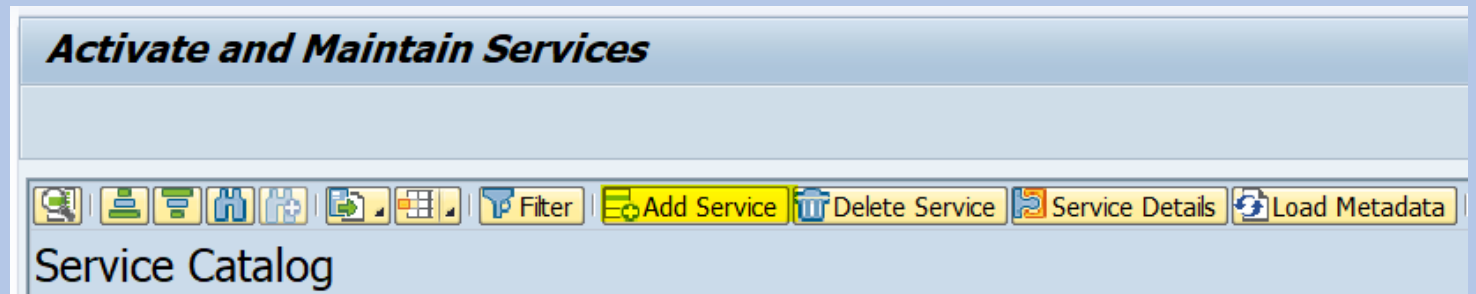
Sort internal table by end date



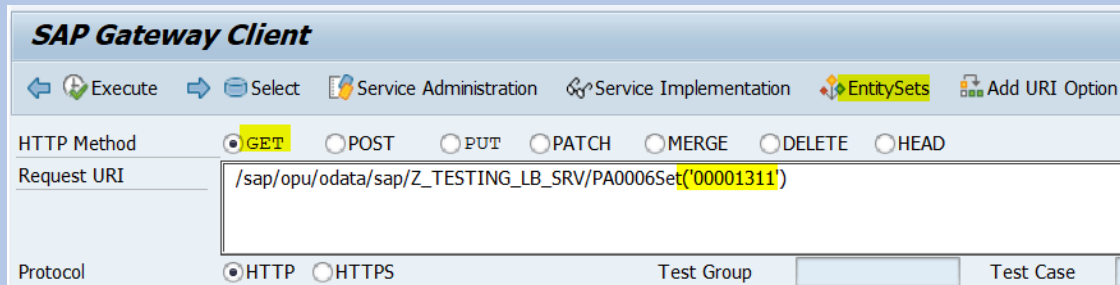
Add the first entry in the internal table to the exporting variable

Part 3: Activating the Service

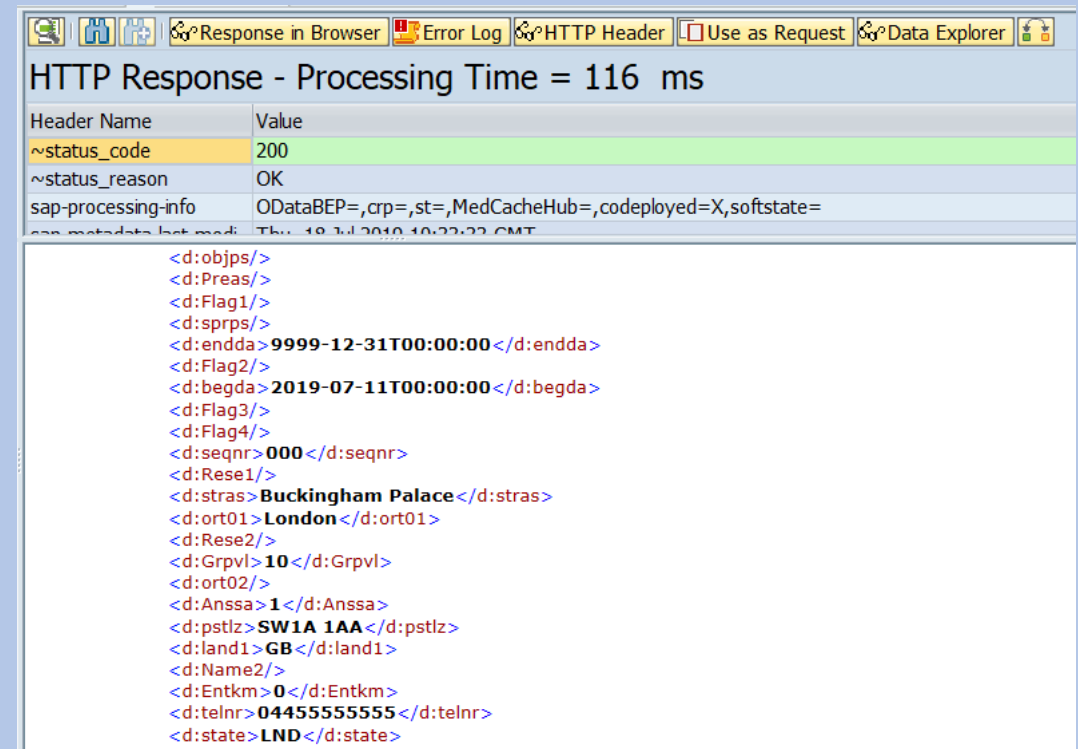
- /n/iwfnd/maint_service
- Catalog of available Services
- Add Service button
- Need to ensure a system alias exists
- To test the service, execute the SAP Gateway Client



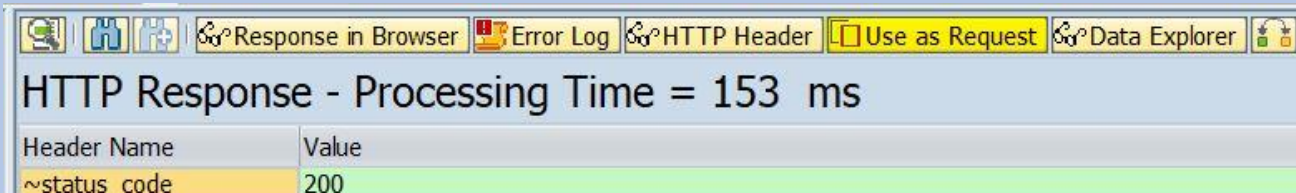
Part 4: Querying the OData service



- SAP Gateway Client
- EntitySets button
- Add Employee Number as key variable
- Results of the query appear in the HTTP response box



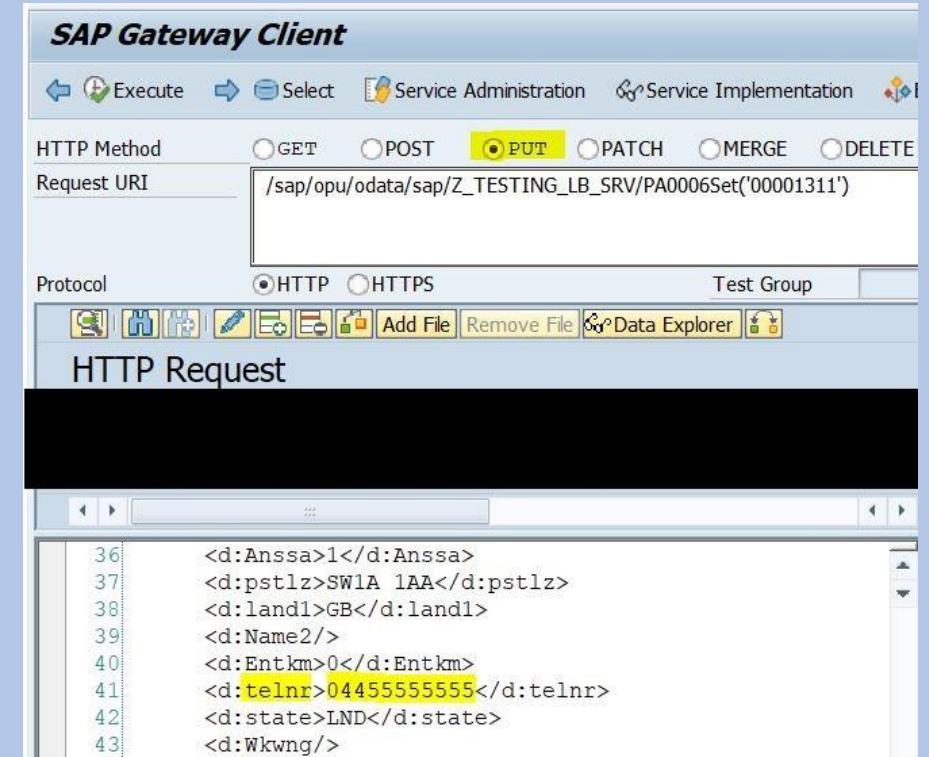
Part 4: Querying the OData service



HTTP Response - Processing Time = 153 ms

Header Name	Value
~status_code	200

- Use as Request button copies Response to the request box
- UPDATE_ENTITY
- Make any changes to the XML in the request box
- PUT method



SAP Gateway Client

Execute Select Service Administration Service Implementation

HTTP Method: ☐ GET ☐ POST ☒ PUT ☐ PATCH ☐ MERGE ☐ DELETE

Request URI: /sap/opu/odata/sap/Z_TESTING_LB_SRV/PA0006Set('00001311')

Protocol: ☒ HTTP ☐ HTTPS Test Group

Add File Remove File Data Explorer

HTTP Request

```
36 <d:Anssa>1</d:Anssa>
37 <d:pstlz>SW1A 1AA</d:pstlz>
38 <d:landl>GB</d:landl>
39 <d:Name2/>
40 <d:Entkm>0</d:Entkm>
41 <d:telnr>0445555555</d:telnr>
42 <d:state>LND</d:state>
43 <d:Wkwng/>
```

Project Conclusions and Evaluation

- A development should be informed by the functional requirements: each technical deliverable is designed to solve a functional problem
- Ideally would communicate with the API using an external interface
- A more sophisticated service could be created to handle:
 - Subtypes: permanent address, work address, emergency address, etc
 - More complex requests: \$stop, \$select, etc