

# An OData service to manage employee addresses

Lily Blake

# 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

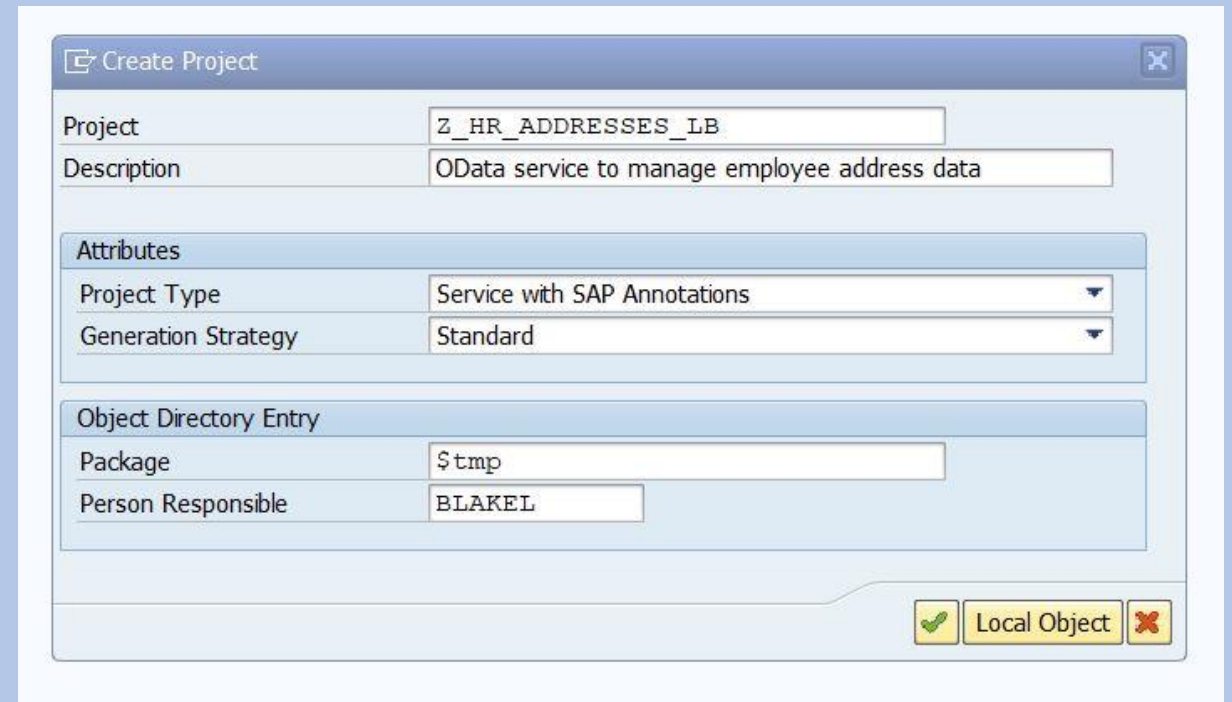
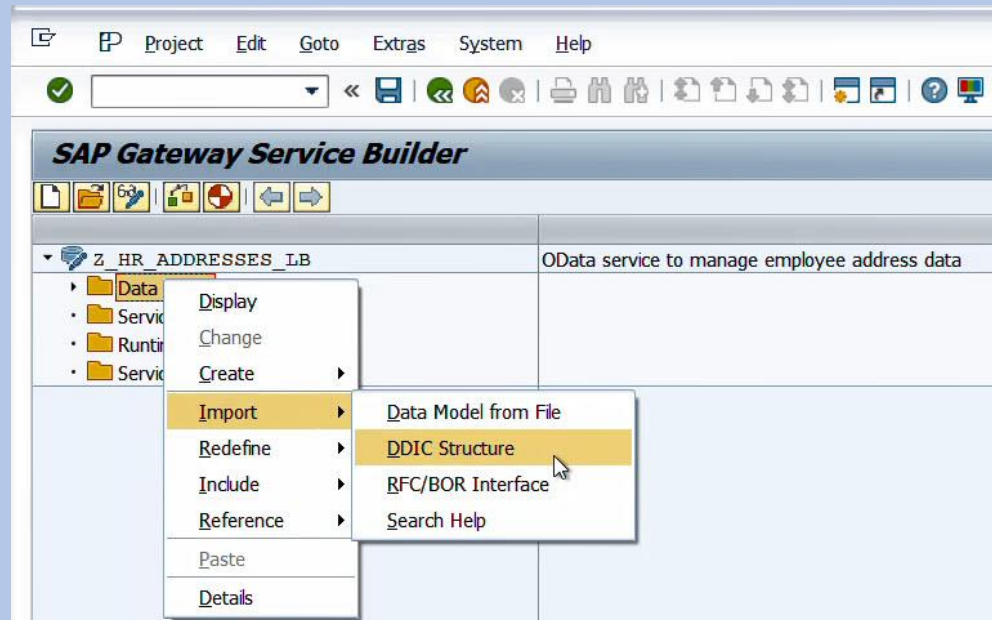
Personnel No	<input type="text" value="16"/>		
Start	<input type="text" value="25.01.2019"/>	to	<input type="text" value="31.12.9999"/>

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



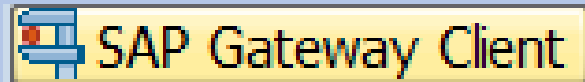
# Part 2: Implementations

An example: the GET\_ENTITY method

Method	PA0006SET_GET_ENTITY	Active
1	<code>method PA0006SET_GET_ENTITY.</code>	
2	<code>**TRY.</code>	
19		
20	<code>DATA: ls_name_value TYPE LINE OF /iwbsp/t_mgw_name_value_pair,</code>	
21	<code>lv_pernr TYPE pa0006-pernr.</code>	
22		
23	<code>READ TABLE it_key_tab INTO ls_name_value</code>	
24	<code>WITH KEY name = 'pernr'.</code>	
25	<code>IF sy-subrc IS INITIAL.</code>	
26	<code>lv_pernr = ls_name_value-value.</code>	
27	<code>ENDIF.</code>	
28		
29	<code>SELECT SINGLE * FROM pa0006</code>	
30	<code>INTO CORRESPONDING FIELDS OF @er_entity</code>	
31	<code>WHERE pernr = @lv_pernr.</code>	
32		
33	<code>endmethod.</code>	

# Part 3: Activating the Service

- /n/iwfnd/maint\_service



**Activate and Maintain Services**

Service Catalog

Filter Add Service Delete Service Service Details Load Metadata

ICF Node Call Browser SAP Gateway Client

ICF Nodes

Status	ICF Node	Session Time-out	Soft State	Description
	ODATA	00:00:00		Standard Mode

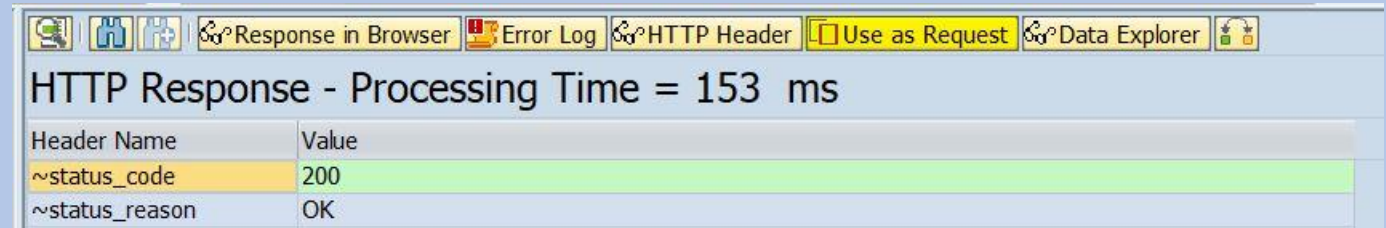
Add System Alias Remove System Alias Customizing Service Implementation

System Aliases

SAP System Alias	Description	Default System	Metadata	User Role
ERP	Connection to the ERP system	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

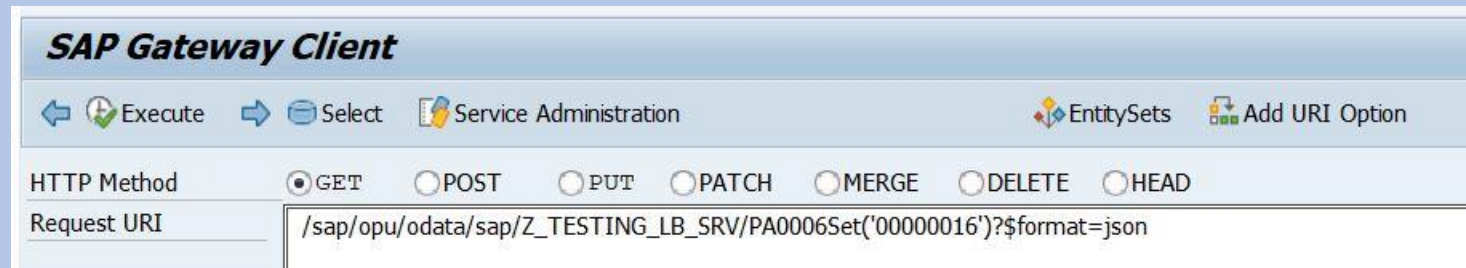
# Part 4: Querying the OData service

- A live demonstration
- Testing each method



The screenshot shows the 'HTTP Response' tab in the SAP Gateway Client. The title bar indicates 'HTTP Response - Processing Time = 153 ms'. Below the title bar is a table with two columns: 'Header Name' and 'Value'. The first row is highlighted in green and shows '~status\_code' with a value of '200'. The second row shows '~status\_reason' with a value of 'OK'.

Header Name	Value
~status_code	200
~status_reason	OK



The screenshot shows the SAP Gateway Client interface. The title bar reads 'SAP Gateway Client'. Below the title bar are buttons for 'Execute', 'Select', and 'Service Administration'. To the right are 'EntitySets' and 'Add URI Option'. The 'HTTP Method' section has radio buttons for GET (selected), POST, PUT, PATCH, MERGE, DELETE, and HEAD. The 'Request URI' field contains the text '/sap/opu/odata/sap/Z\_TESTING\_LB\_SRV/PA0006Set('00000016')?\$format=json'.

**SAP Gateway Client**

Execute Select Service Administration EntitySets Add URI Option

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

Request URI /sap/opu/odata/sap/Z\_TESTING\_LB\_SRV/PA0006Set('00000016')?\$format=json



# 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