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

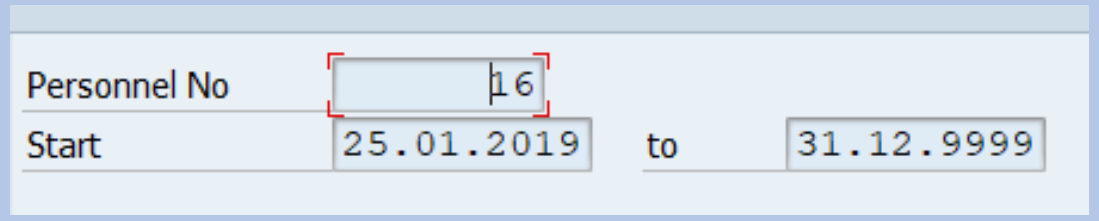
- “An open data protocol to consume RESTful APIs”
- Translation:

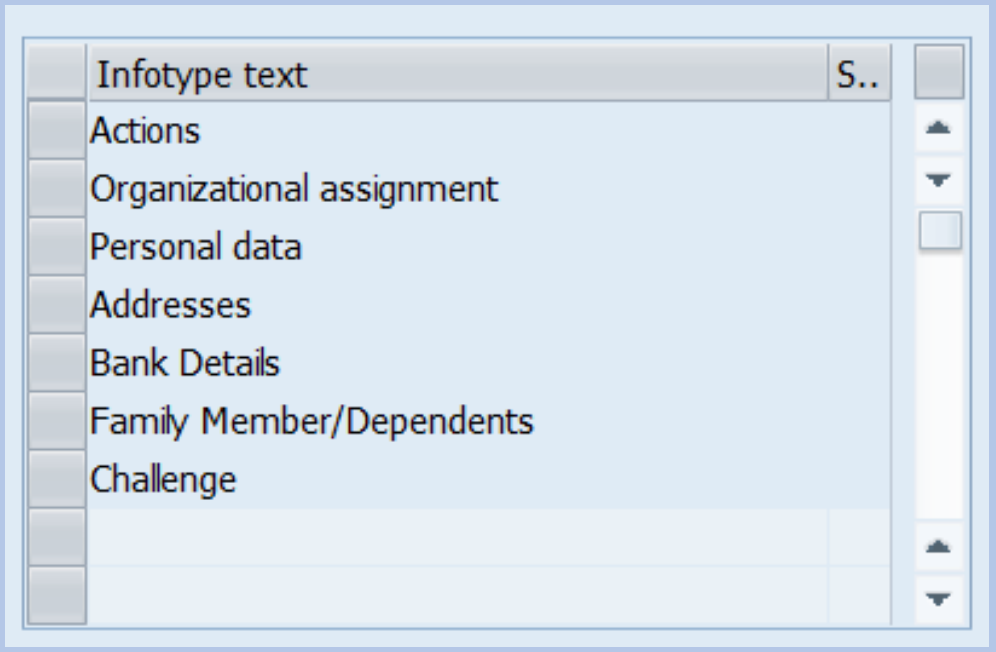
- Open Data Protocol: standardised creation and consumption of REST APIs
- API: a set of protocols and procedures that access the data/features of a system
- REST: a software architectural style: stateless client-server model

- 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

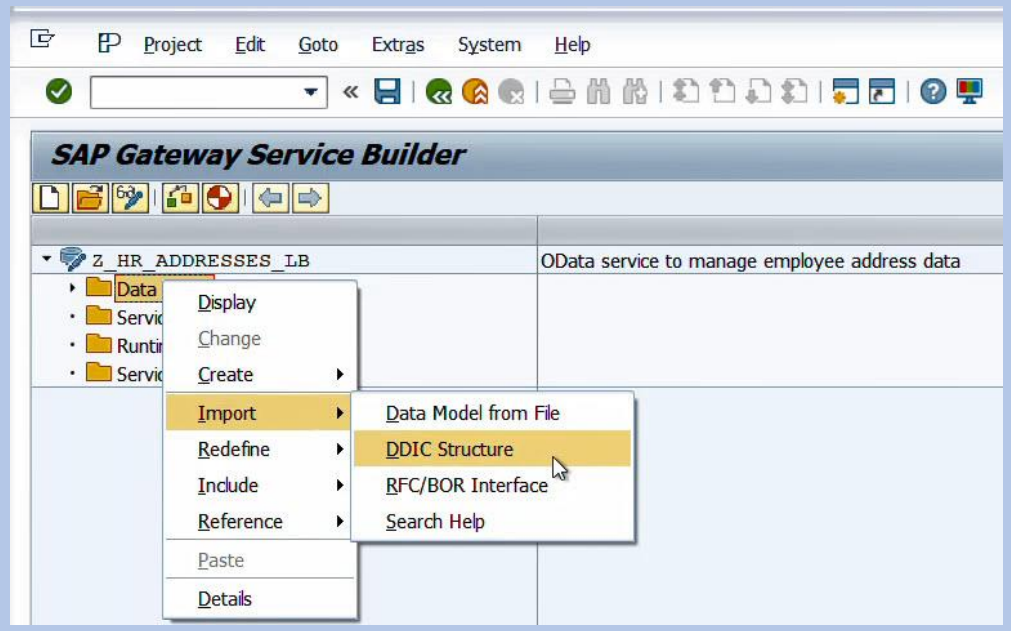


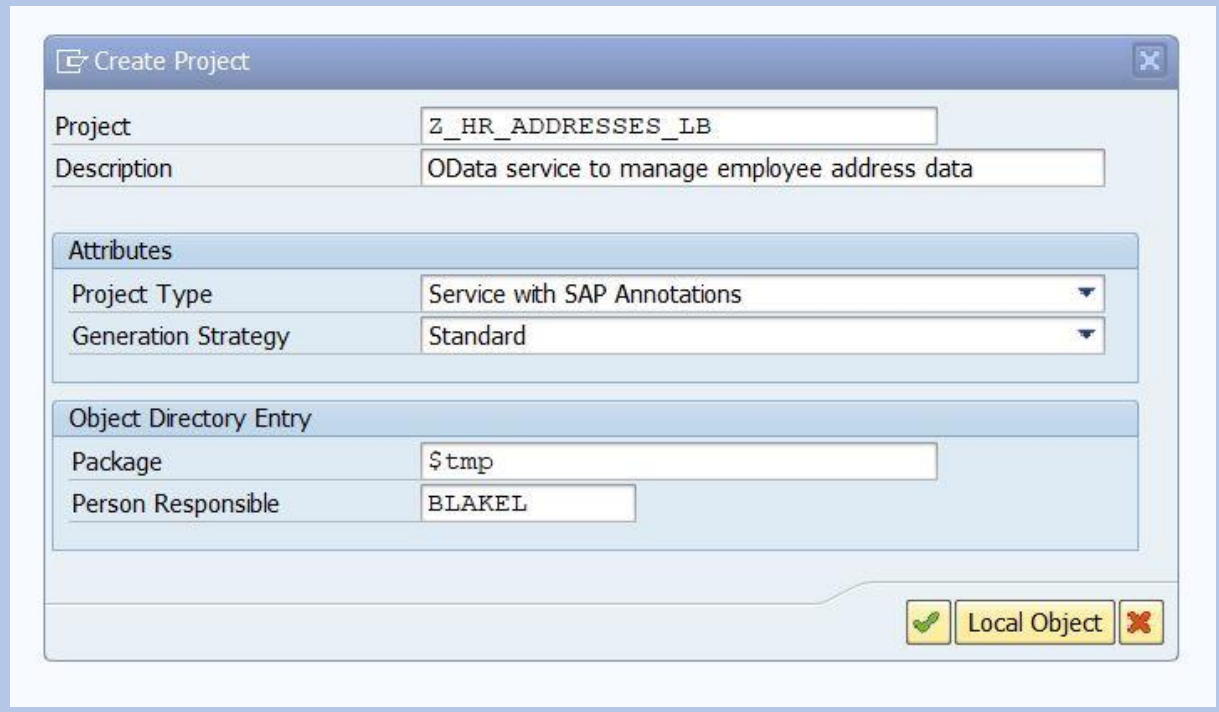


Addresses = PA0006

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

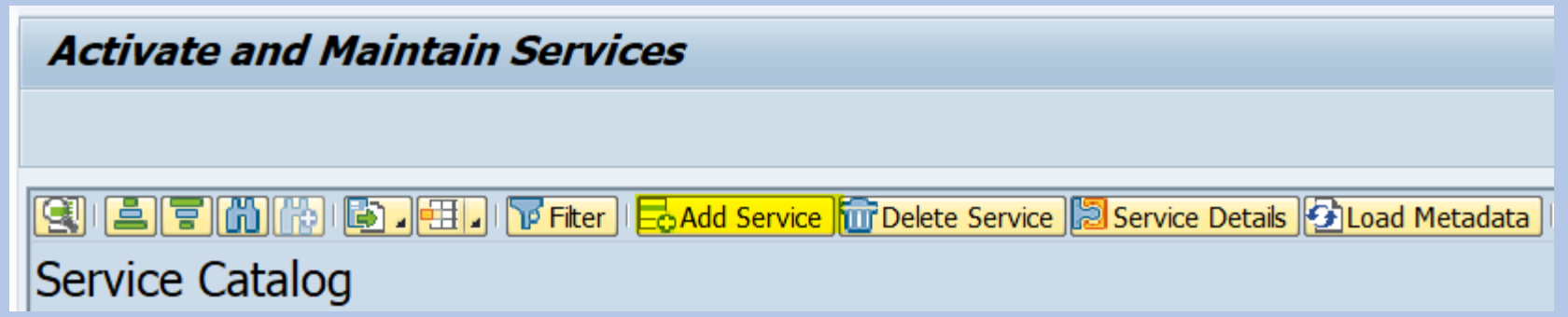
An example: the GET_ENTITY method

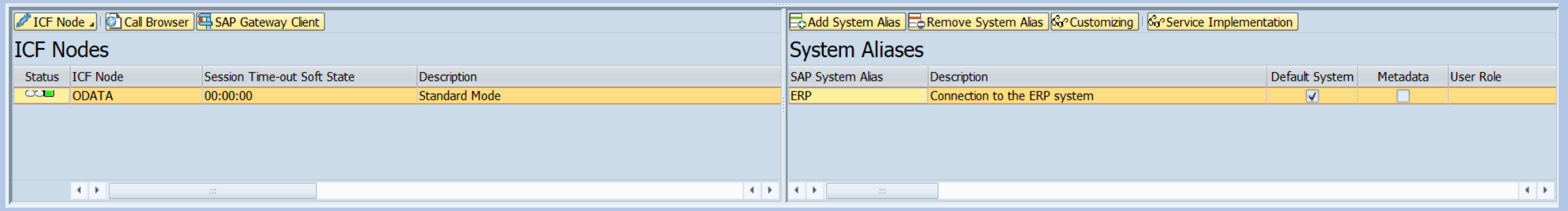
Method PA0006SET_GET_ENTITY Active

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

Part 3: Activating the Service

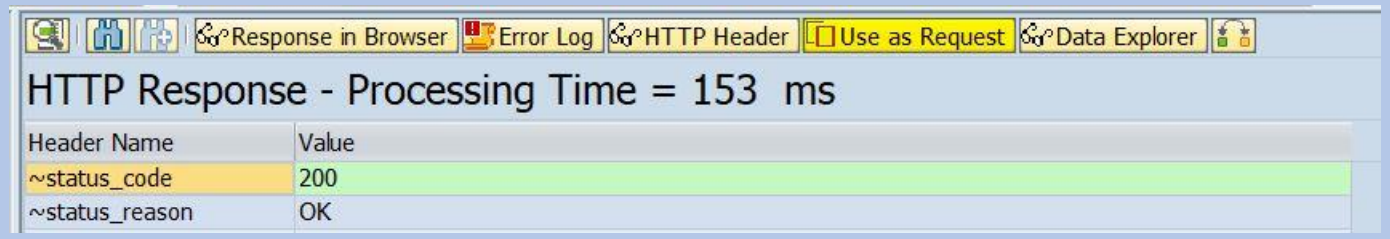
- /n/iwfnd/maint_service

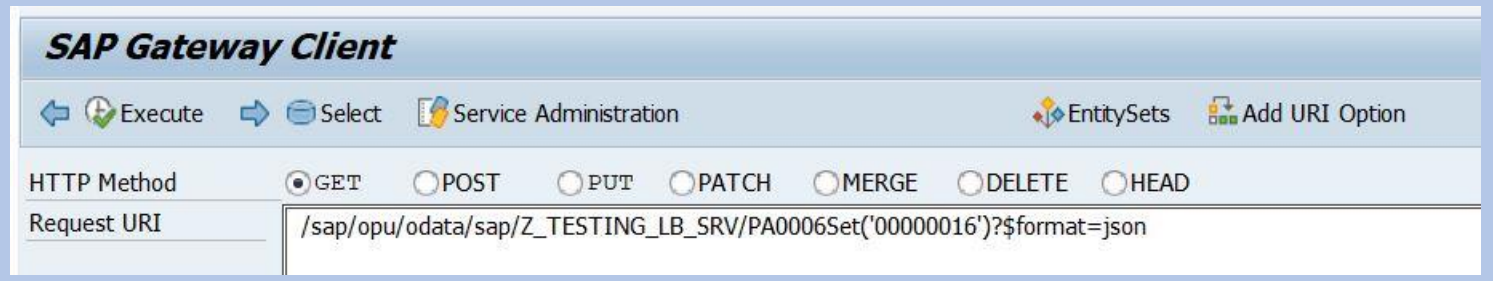




Part 4: Querying the OData service

- A live demonstration
- Testing each method





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