OData Adapter

Bridge 7 OData, short for Open Data Protocol, defines a protocol for querying and updating databases using REST. The protocol allows Web clients to get, publish and edit database resources using simple HTTP messages.

The E2E Bridge features the import of OData CSDL files describing an OData REST interface. The importer will generate the data (entities and types) and service elements to your xUML model. Refer to Importing OData Entities and Services for more information on the E2E OData importer.
After the import, you can call the OData REST service using the E2E REST Adapter.

Example File (Builder project Add-ons/OData):

<your example path>\Add-ons\OData\uml\odataSampleService.xml

On this Page:

- Preventing Concurrent Updates Using ETag
- Server Side Pagination via OData NextLink

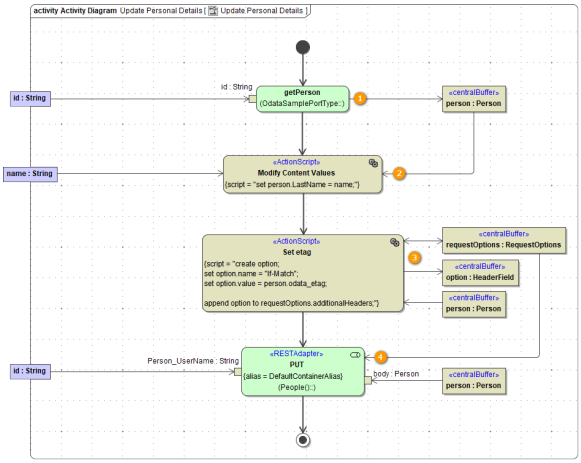
Related Pages:

- Importing OData Entities and Services
- OData Import Rules
- REST Adapter

Preventing Concurrent Updates Using ETag

OData features the usage of ETag to prevent concurrent updates. If wanting to update a record, you need to read it first. This read will provide you with an ETag. You need to pass this ETag with the request headers of your update/delete call. The request will only be performed, if the ETag has not changed meanwhile.

Calling OData Service Using ETag



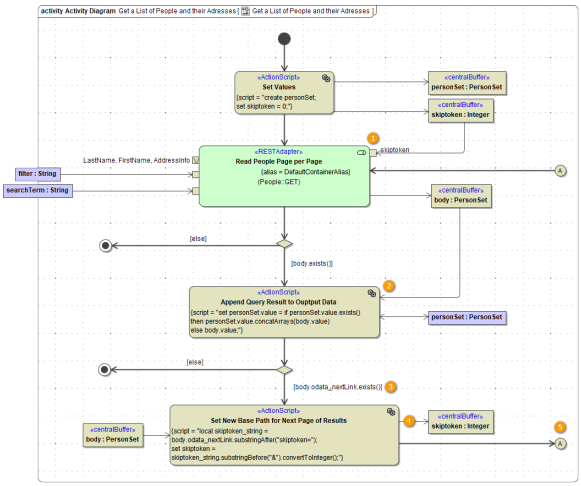
1. Read the record first.
2. Make your modifications.
3. Set the ETag provided in **odata_etag** to the request headers.
   The ETag needs to be set in header "If-Match".
4. Update the record.

If everything was fine, the update request will return with HTTP status code 2xx. If the record has been changed meanwhile, the request will return an error status.

Server Side Pagination via OData NextLink

OData features server side pagination. When requesting a collection of database items, the client will only get a first page of records and a so called "NextLink". This NextLink is stored in field odata_nextLink. Use this link for subsequent requests to get the next pages.

Calling OData Service Using odata_nextLink



1. Read a page of records.
2. Append the query result to the output structure.
3. Check if there is a NextLink available.
4. Set **skiptoken** from odata_nextLink.
   Parameter **skiptoken** will instruct the OData service to skip the specified count of records (first to actual page) and return the next page of records.
5. Repeat this until all search results are delivered (no NextLink available, `odata_nextLink = NULL`).