xDI DEV

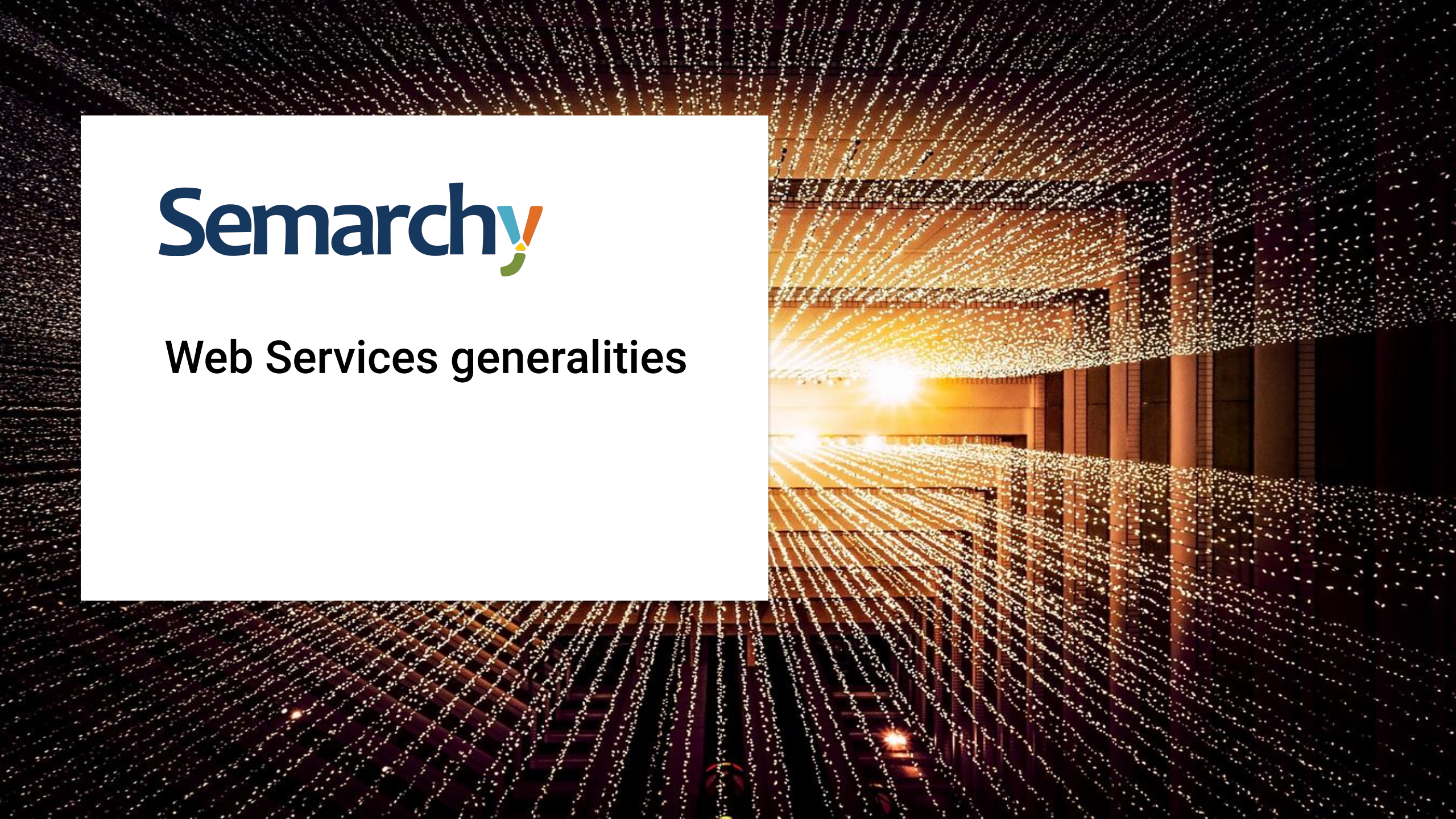
F Web Services

- Web Service

xDI DEV

F Web Services

F1 - Principles & Metadata



Web Services generalities

Web Services

Provide a **service via Internet :**

Allows the communication between two computers (or applications)

Must have two essential properties

Multi-plateforme : not mandatory that client and server have the same configuration to communicate. The Web Service allows to meet up at the same level

Shared : a web service is often available for more than a customer

When a Web Ser is used,

A **client** send a query to a server and activates an action from this server

The **server** returns an answer to the client

SOAP

- Initially, Web Services exclusively built via SOAP
  - SOAP: Single Object Access Protocol
- The communication protocol is defined for SOAP norm in the WSDL
  - WSDL: Web Services Description Language
  - Using an XML grammar allowing to describe the Web Service
- Input and output parameters can be
  - Single parameters
  - XML

REST

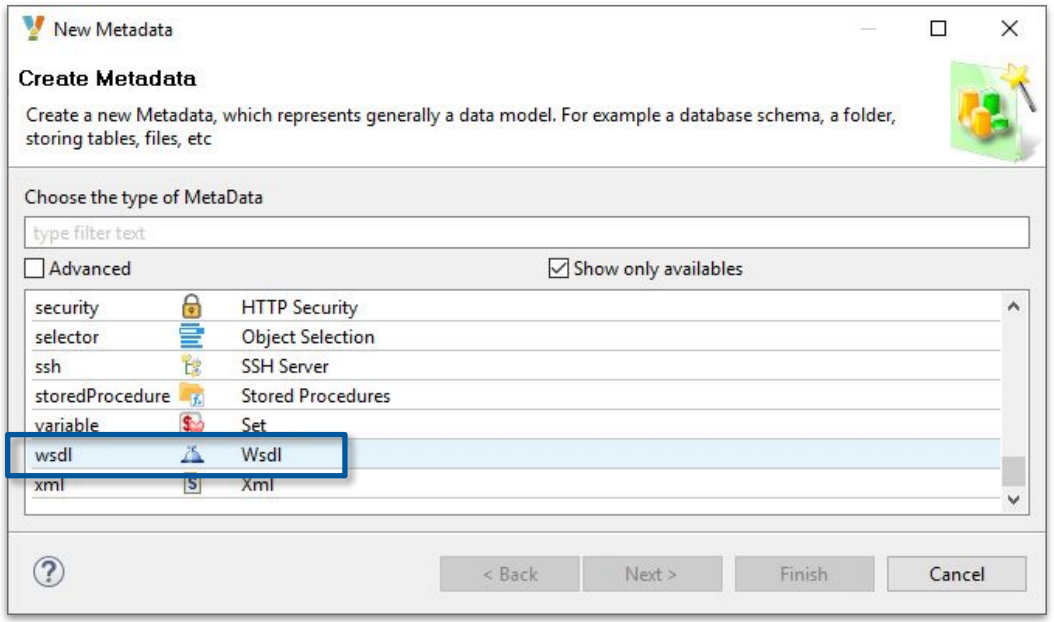
Web Services are now mostly developed using REST mode

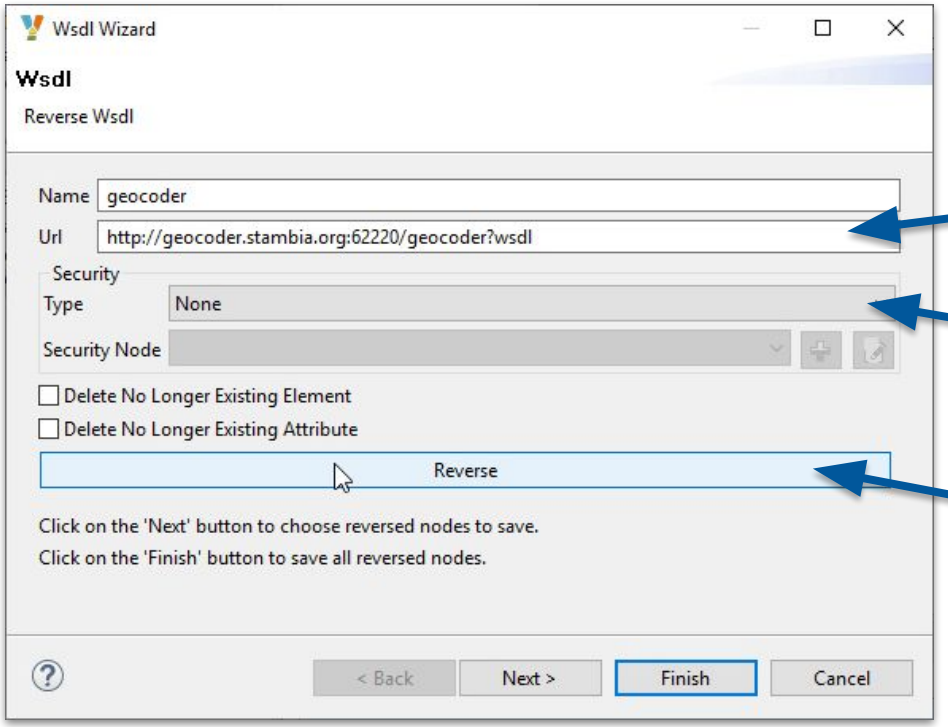
- REST mode uses a standard Uniform Resource Identifier (URI)
- Format of a REST Web Service call:
  https://www.mycompany.com/program/method?Parameters=value
- REST can use JSON in addition to single parameters and Xml
- Rest is easy to use and can be implemented with HTTP/HTTPS client or server
- The available methods in REST mode are standard HTTP methods :
  - GET to query
  - POST to create records
  - PUT/PATCH to update records
  - DELETE to delete records

Reverse engineering Web Services

Reverse engineering a SOAP Web Service

SOAP Web Services are reversed by using the WSDL (Web Services Description Language) Metadata type





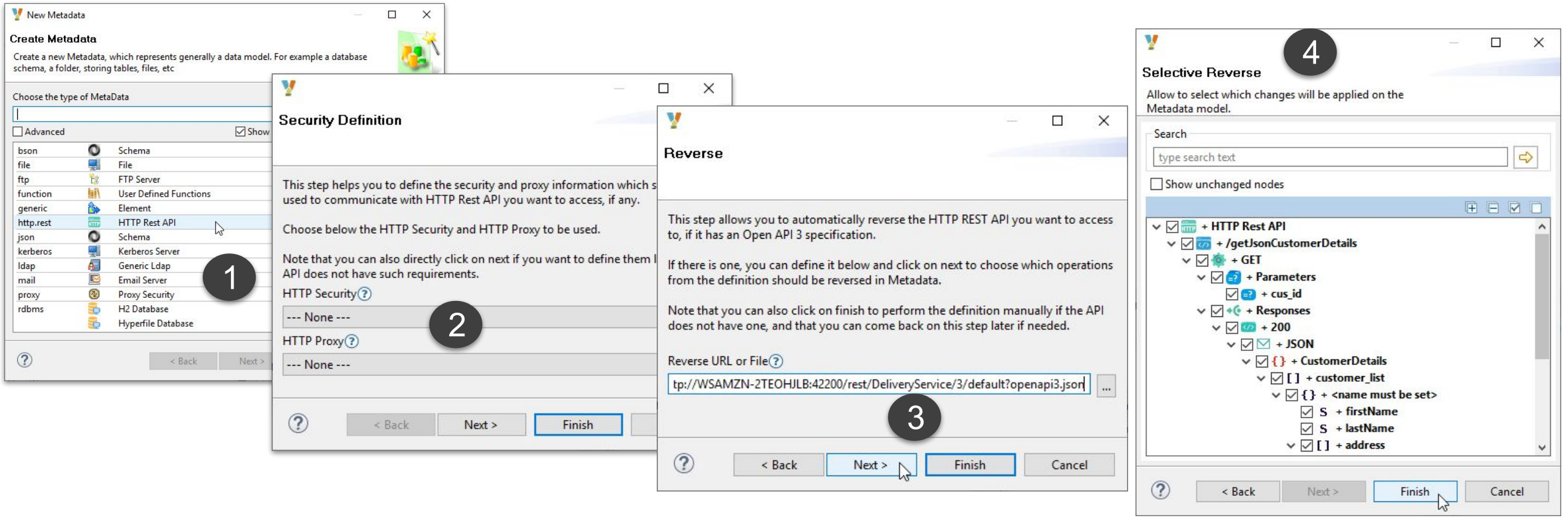
Url for the WSDL is required

Define optionally the Security

Push on "Reverse" button

Reverse engineering a REST Web Service

Reverse engineering with HTTP REST API metadata for an Open API 3 REST Web Service



Reverse engineering a REST Web Service

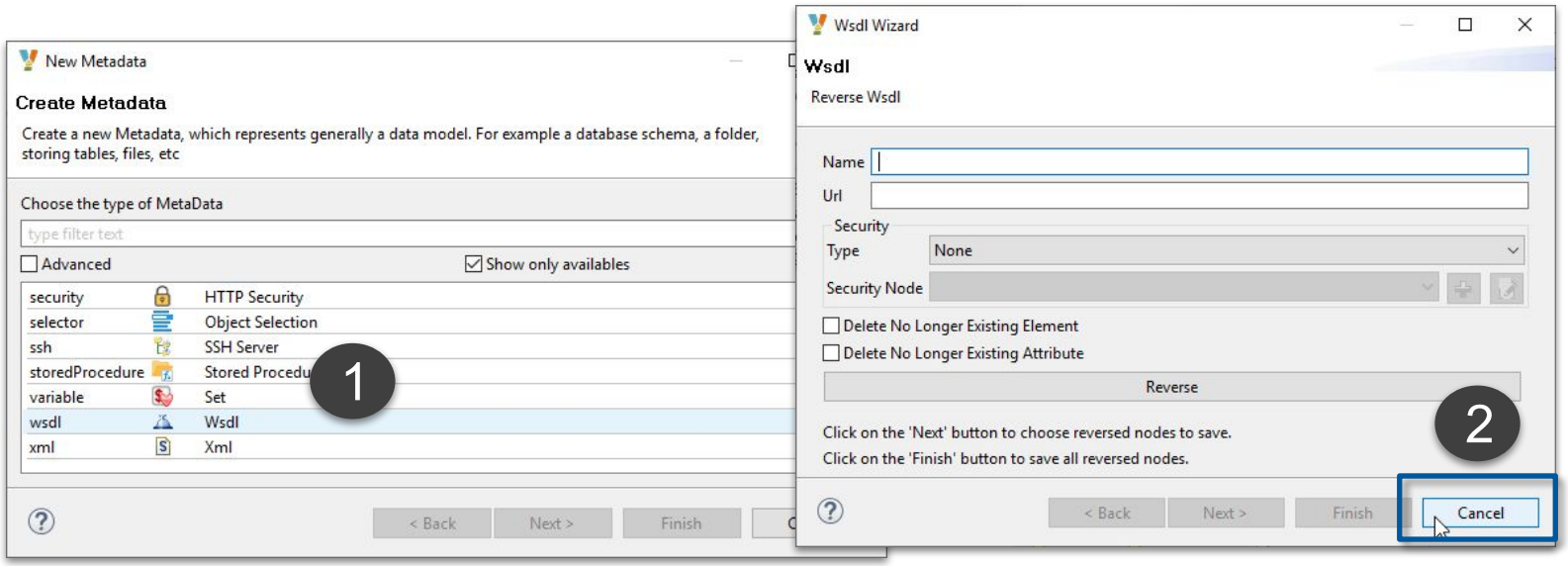
Otherwise, it's also possible to reverse engineering a REST Web Service

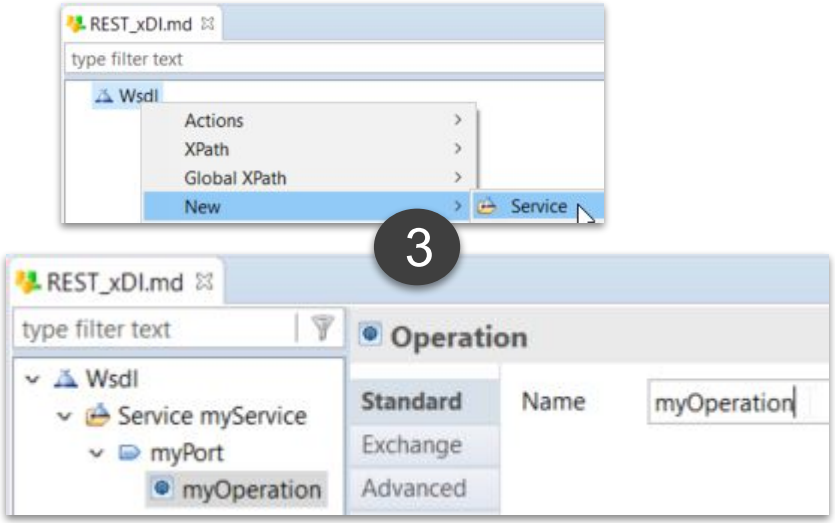
- Beginning in the same way as a SOAP Web service

1. Create a new metadata, choosing "wsdl"

2. Push on the "Cancel" button without specifying any field

3. Create a new service with a name. In the same way, create a port under the new service and a new operation under the new port

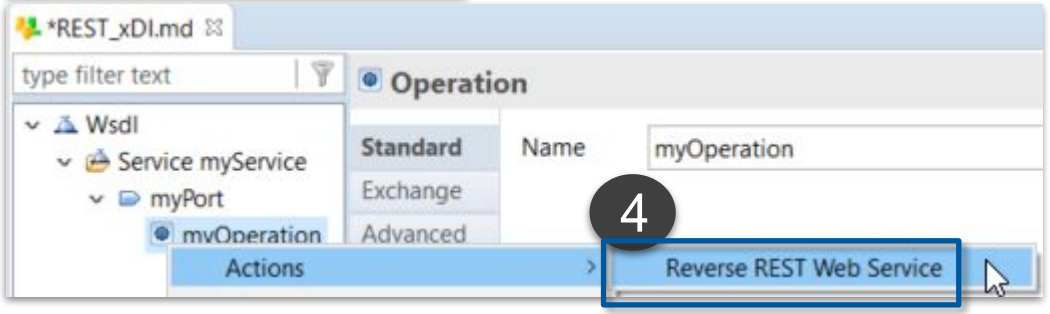




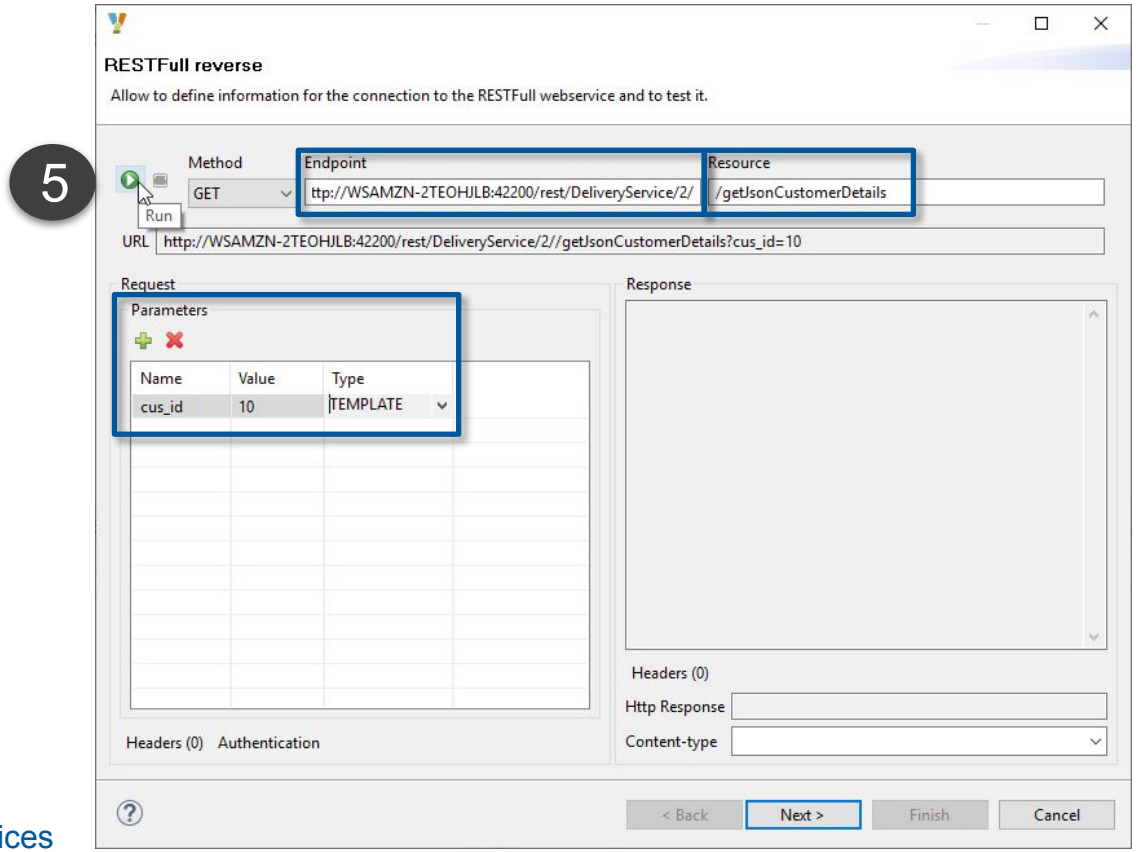
Reverse engineering a REST Web Service

At the operation level, a wizard can be launched

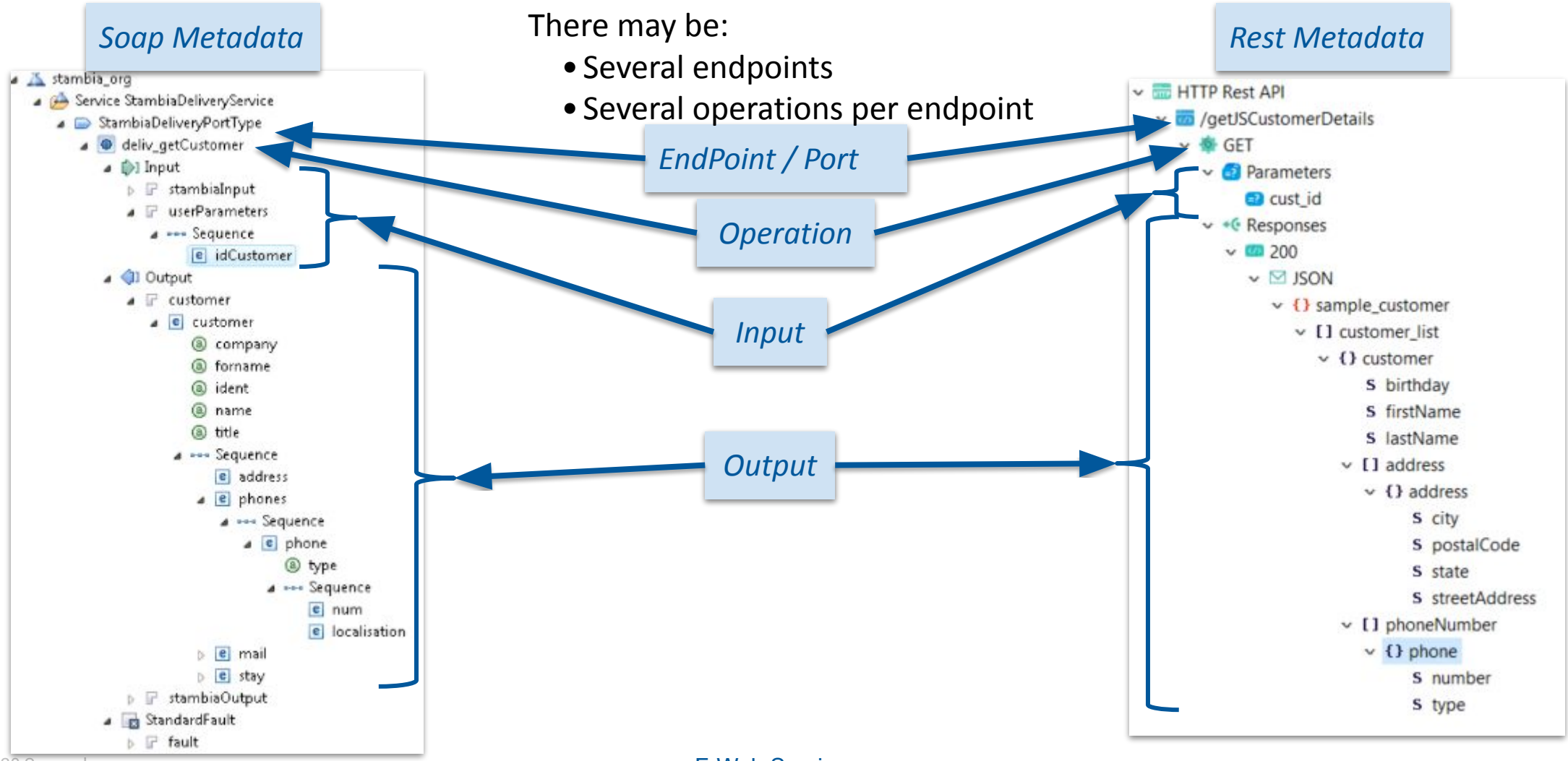
4. Start the wizard

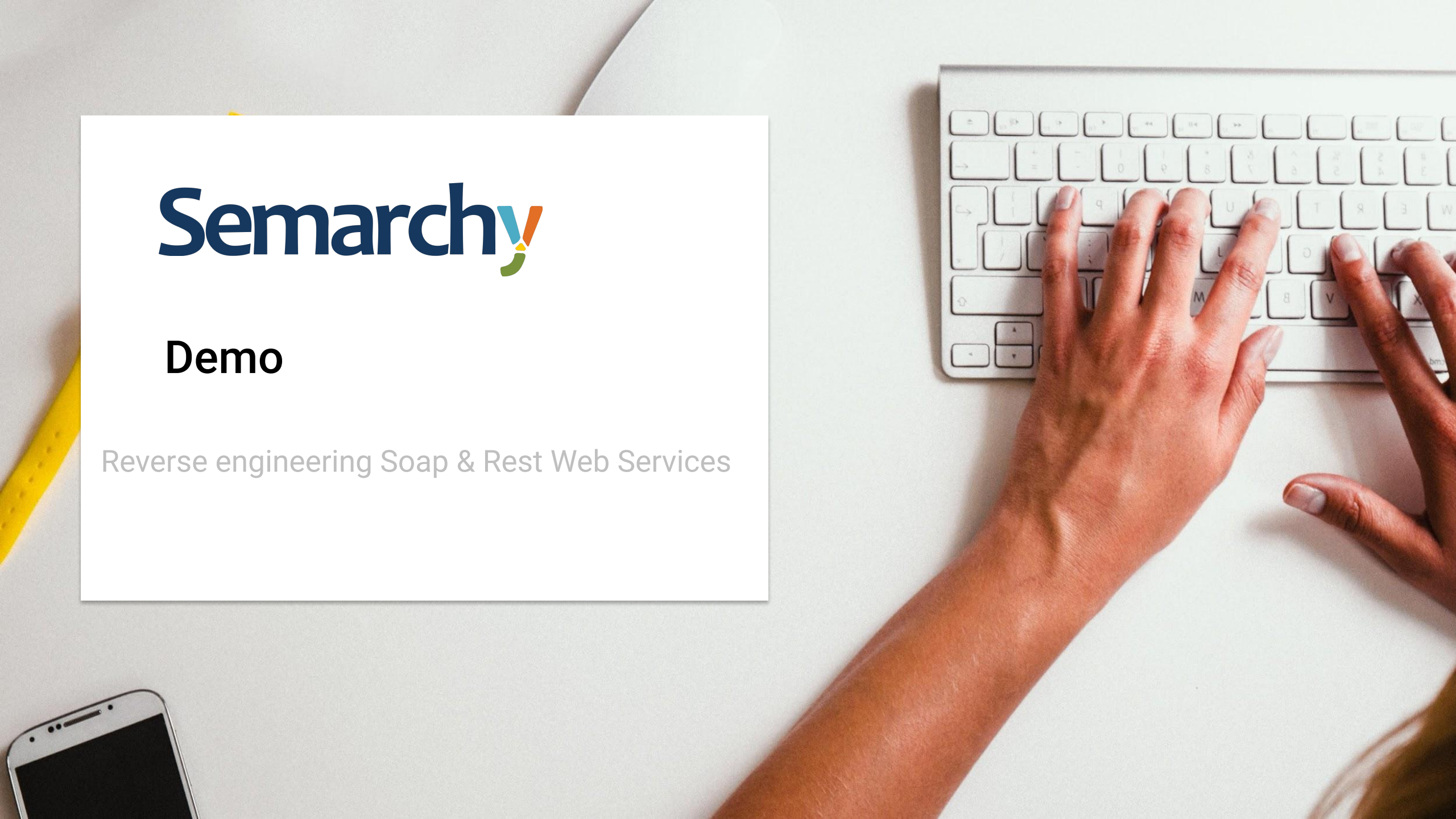


5. Specify the different fields and push on the "Run" button allowing to retrieve the Response



Soap & Rest Metadatas





Demo

Reverse engineering Soap & Rest Web Services

- Web Service

Semarchy

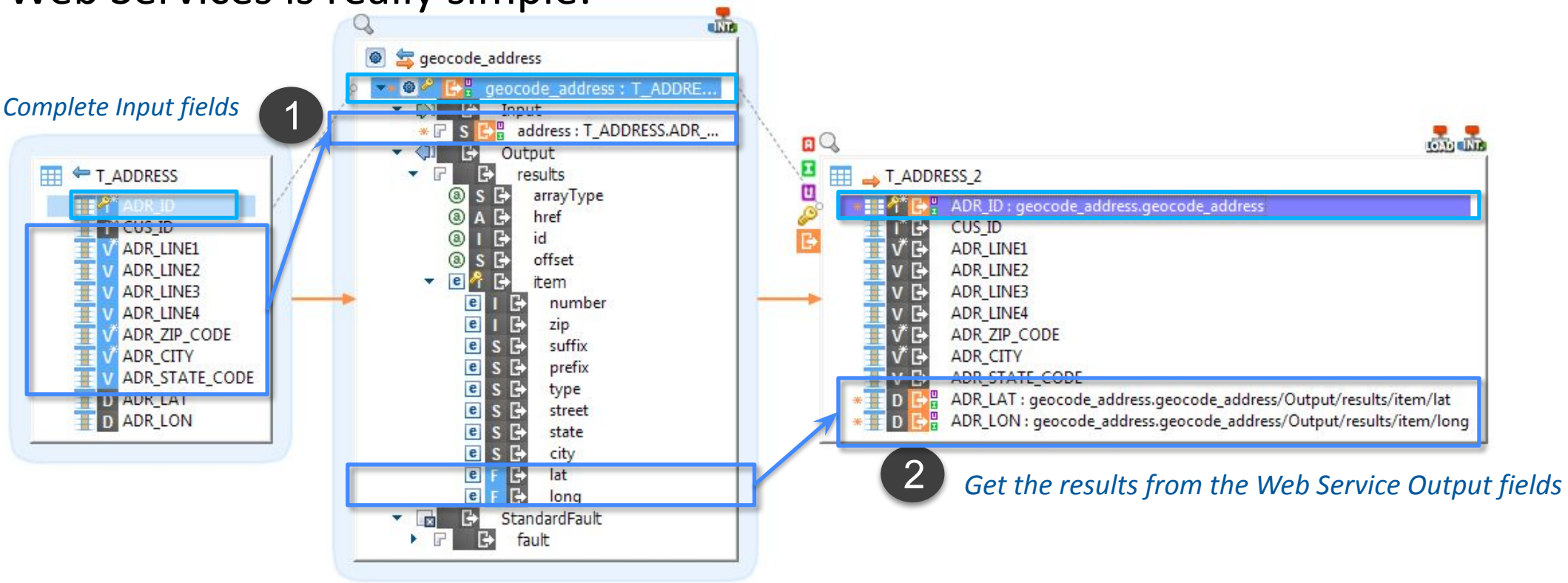
xDI DEV

F Web Services

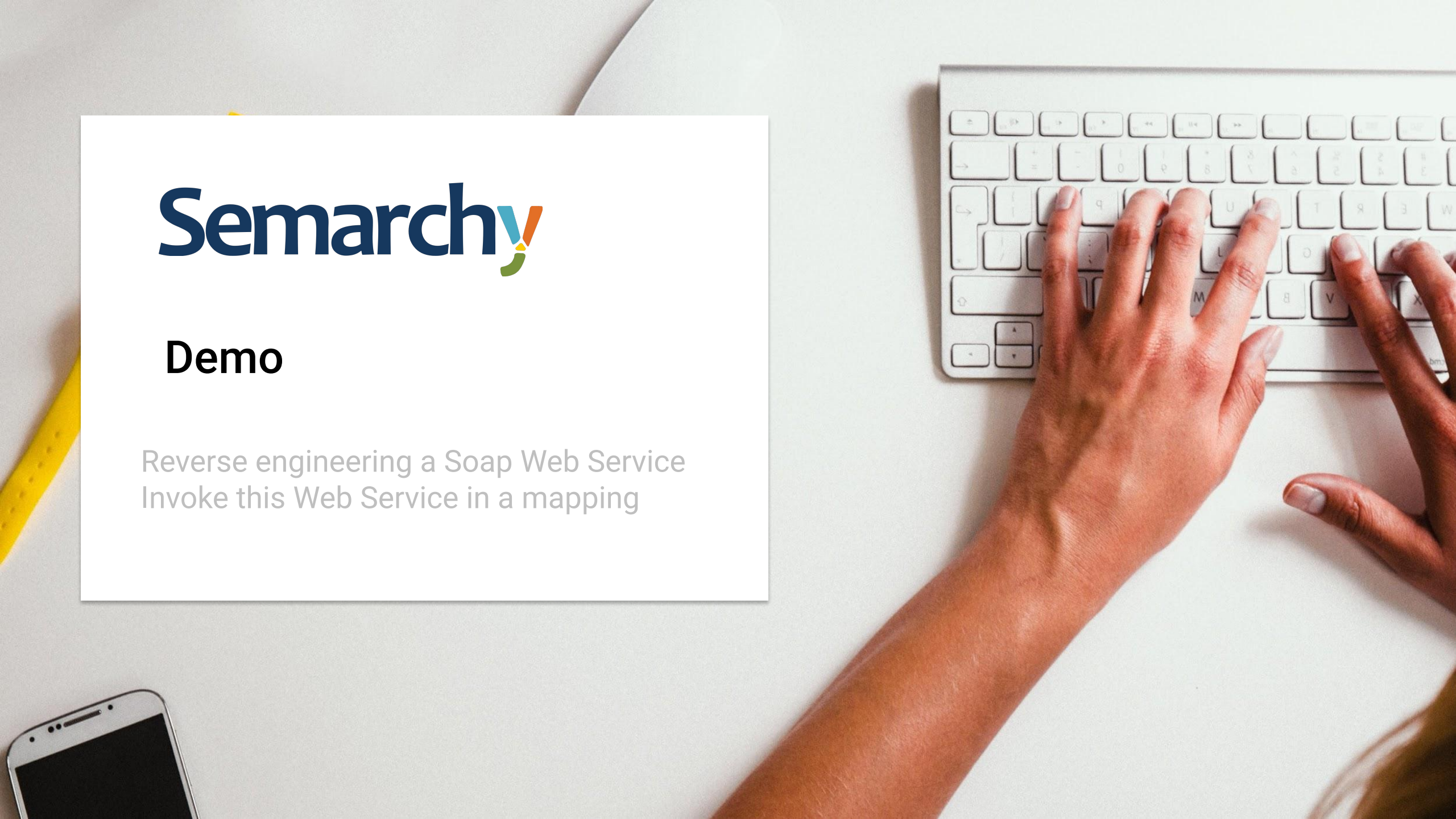
F2 - Invoke Web Services

Using Web Services with Mappings

Using Web Services is really simple:



Note : *the synchronization is done on the Web Service Operation field (Web Service node, highlighted with a key) populated with the source identifier*



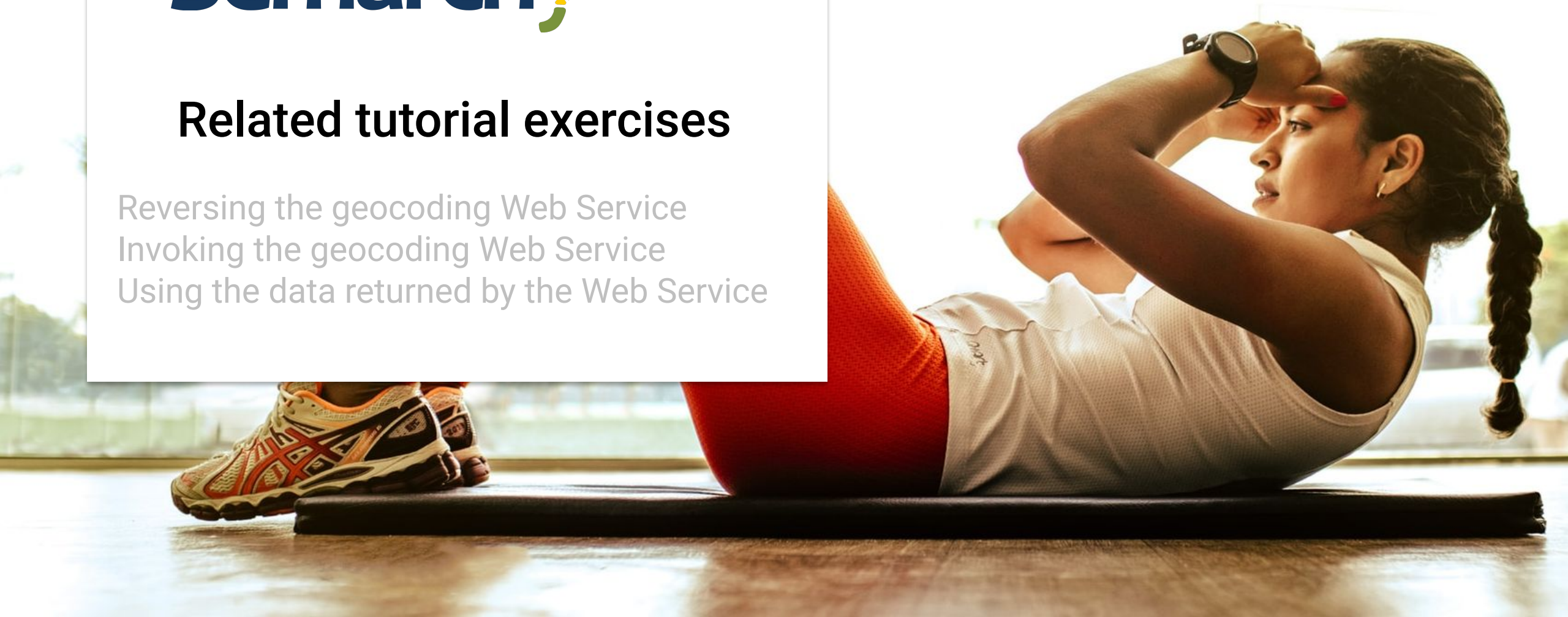
Demo

Reverse engineering a Soap Web Service
Invoke this Web Service in a mapping

Semarchy

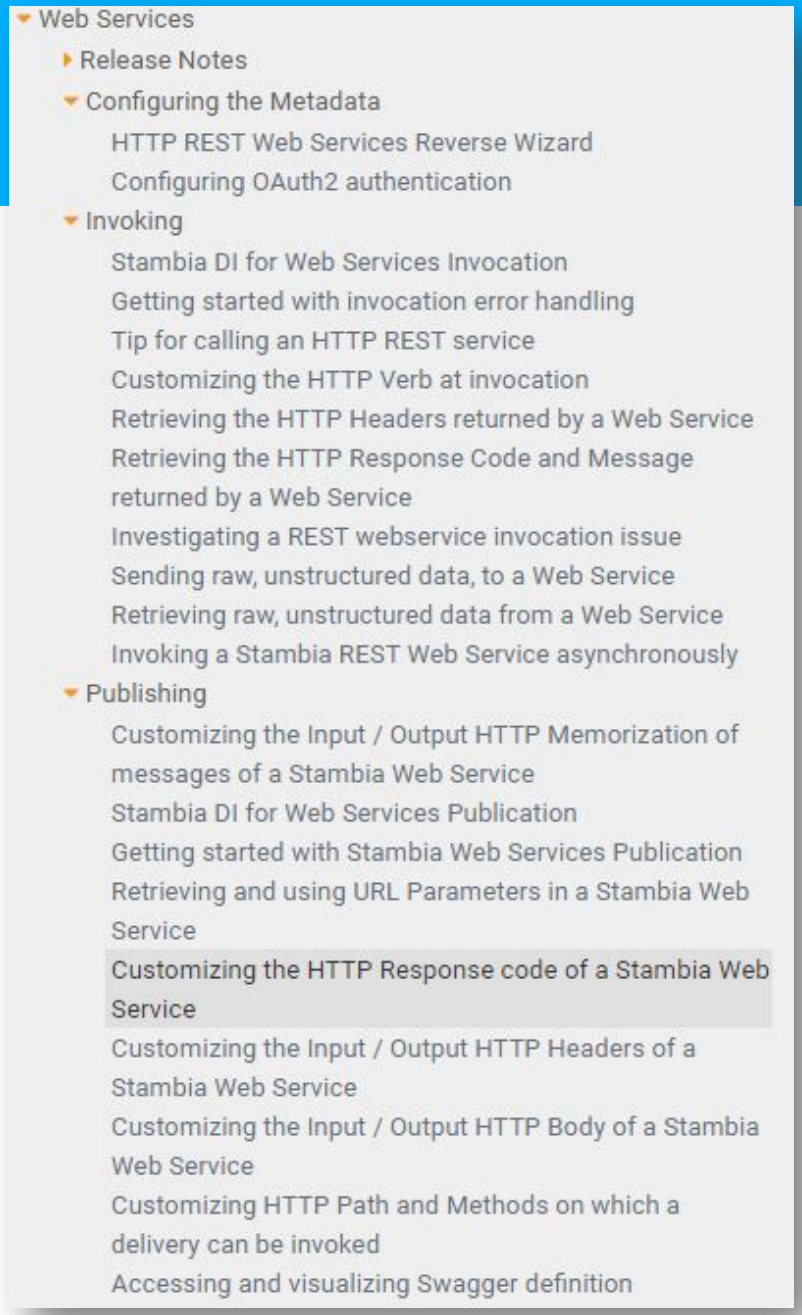
Related tutorial exercises

Reversing the geocoding Web Service
Invoking the geocoding Web Service
Using the data returned by the Web Service



Some interesting links

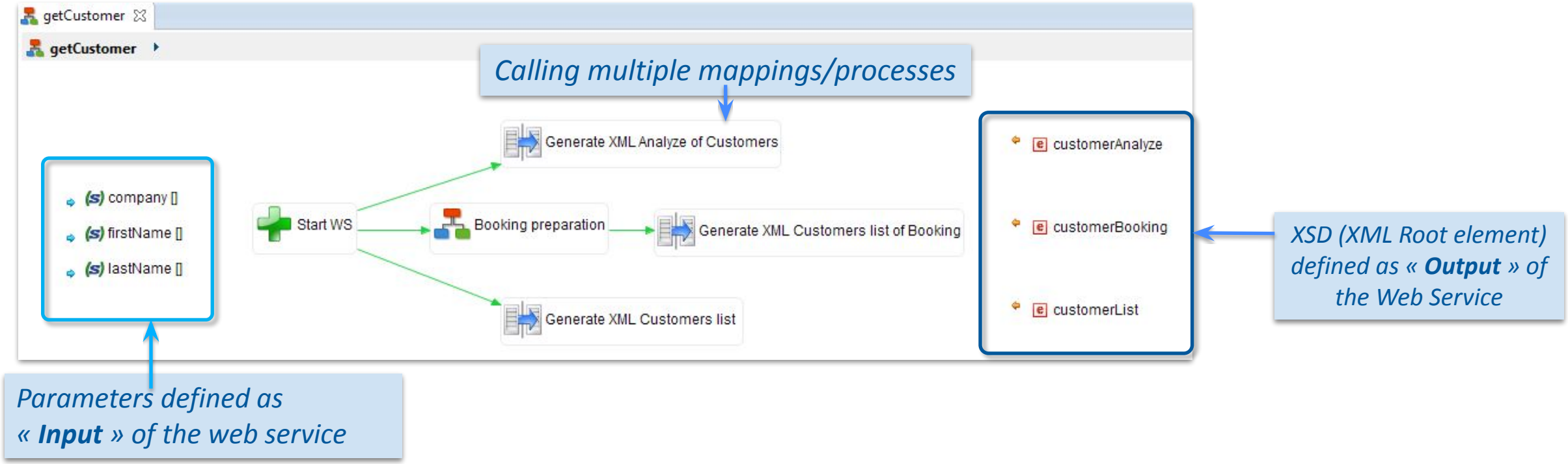
- Invocation error handling
  - https://stambia.org/doc/65-technology-articles/web-services/invoking/634-getting-started-with-invocation-error-handling
- Http response code and message returned
  - https://stambia.org/doc/65-technology-articles/web-services/invoking/431-retrieving-the-http-response-code-and-message-returned-by-a-web-service



- Web Service

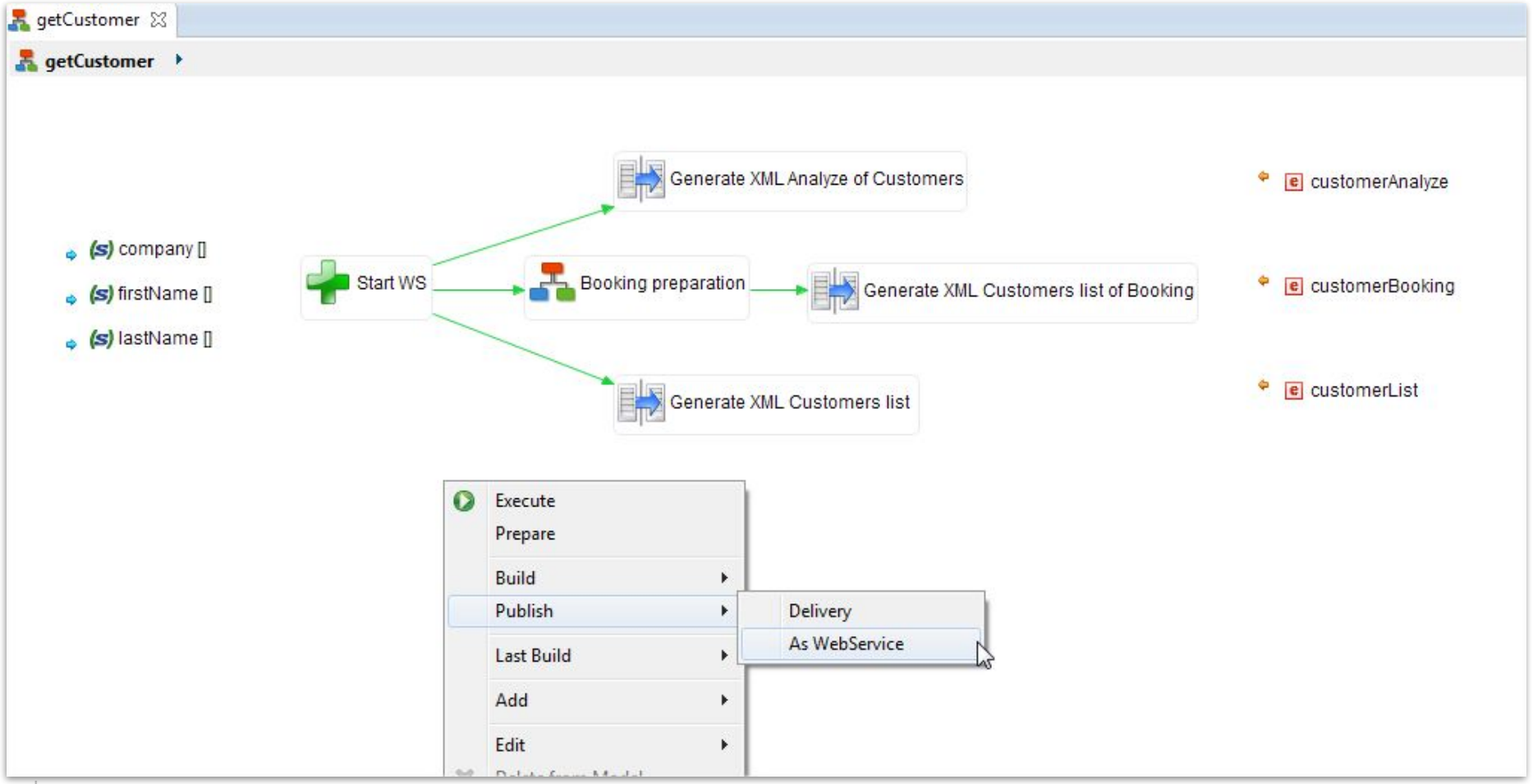
Publishing a Process as a web service

- A Process can also be published as a Web service
- They are published on the runtime which exposes them and generates automatically a WSDL that can be retrieved through a url



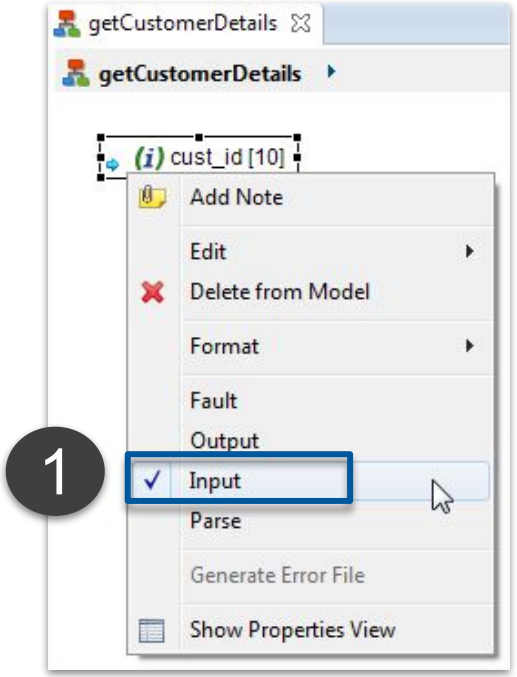
Publishing the process as Web Service

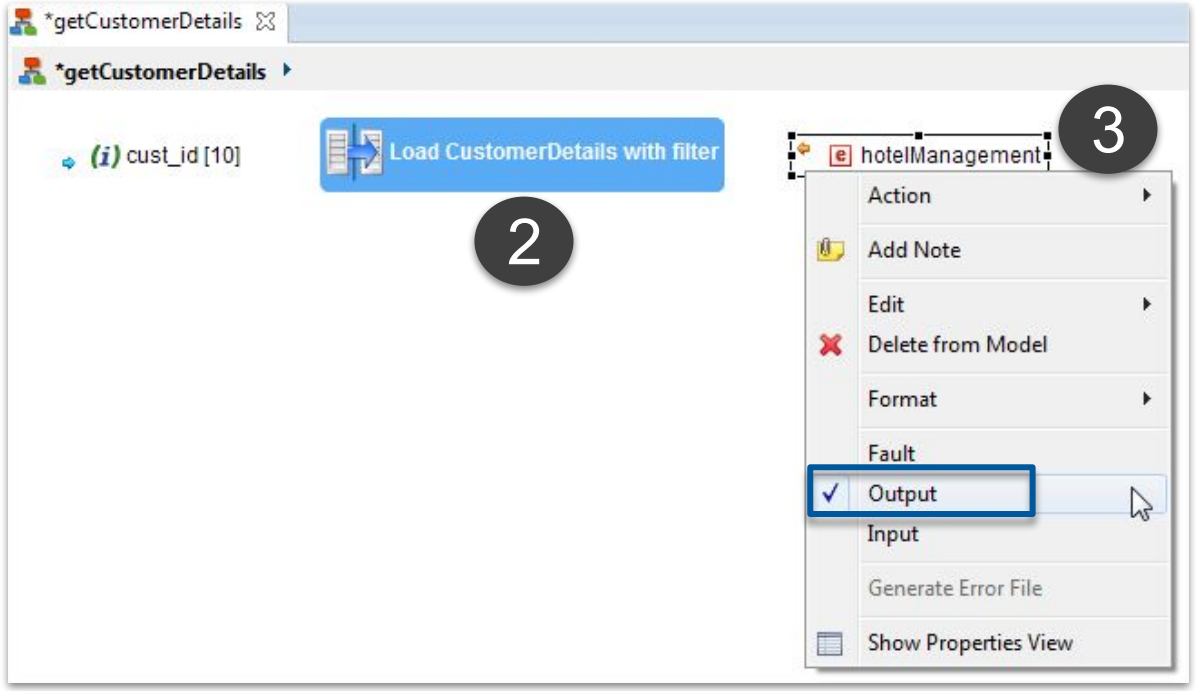
To finalize the creation of the Web Service, publish the process "*As WebService*"



Publish a process as a Web Service step by step

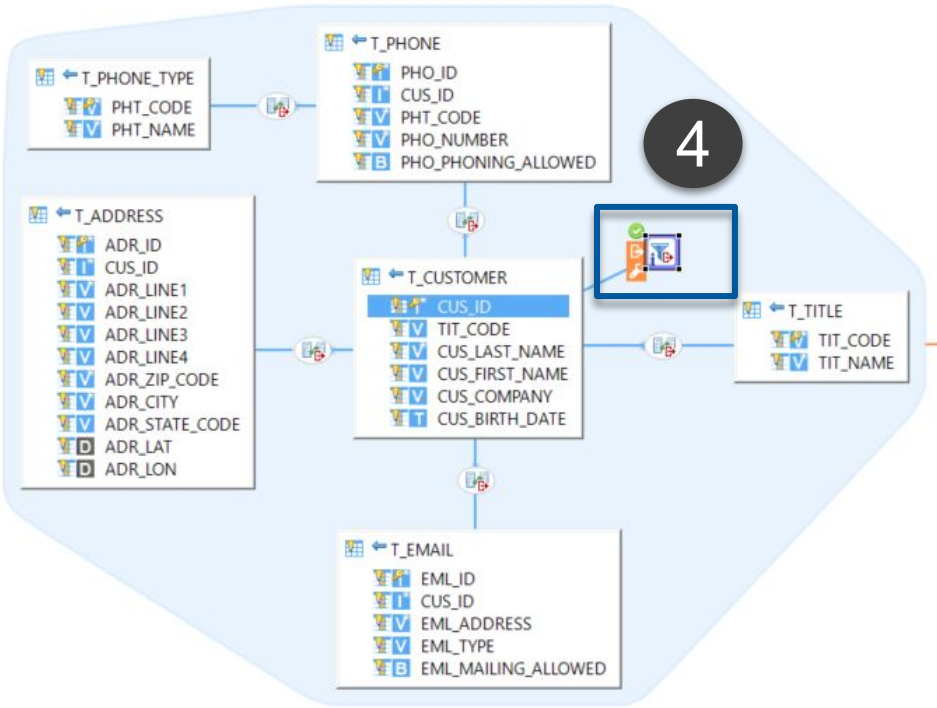
1. Create a process named “getCustomerDetails” (name of the Web Service operation)
   Create a parameter cust_id with default value 10, defined as an input (right click/Input)
2. Drag & Drop the mapping and the root element used as an output
3. Right click on the root element and define it as an output for the Web Service

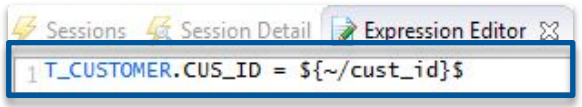


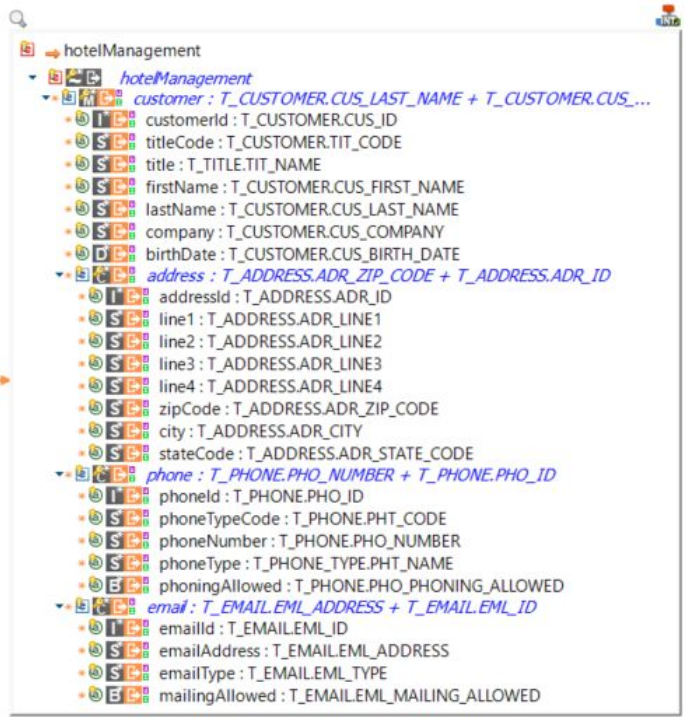


Publish a process as a Web Service step by step

4. Add a filter in the mapping on CUS_ID and test the equality with the value of the “cust_id” parameter



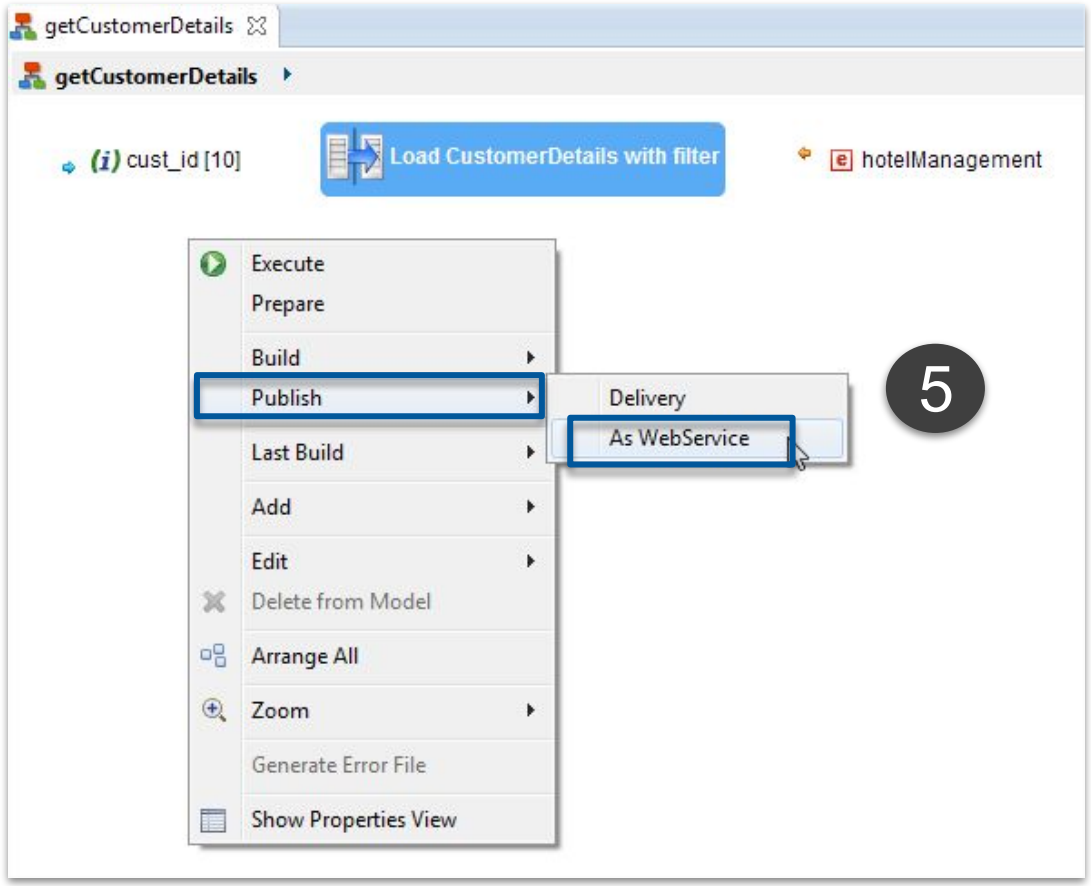




Note : *No Xml path (in metadata or in Integration Template) must be defined for a Web Service*

Publish a process as a Web Service step by step

5. Right click on the process and choose "*Publish/As Web Service*"



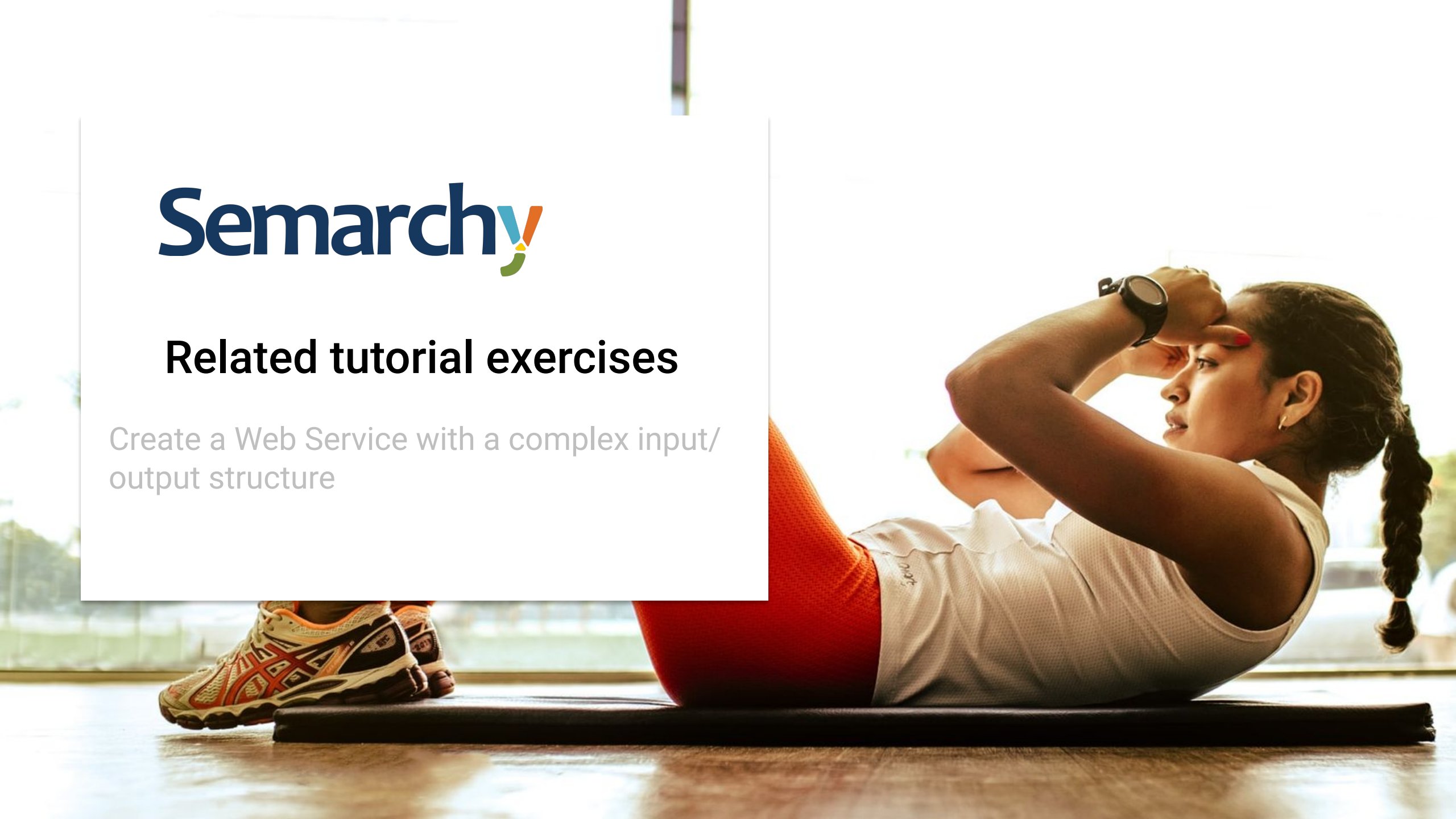
Demo

Publish a process as a Web Service

Semarchy

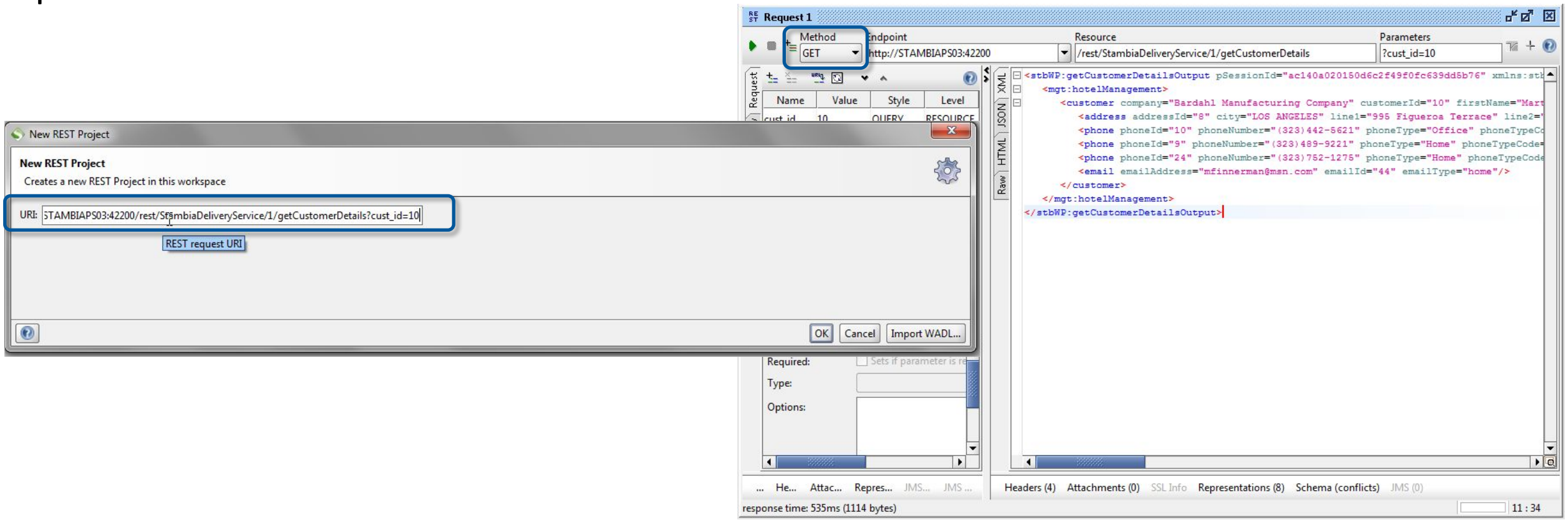
Related tutorial exercises

Create a Web Service with a complex input/ output structure



Test the Web Services

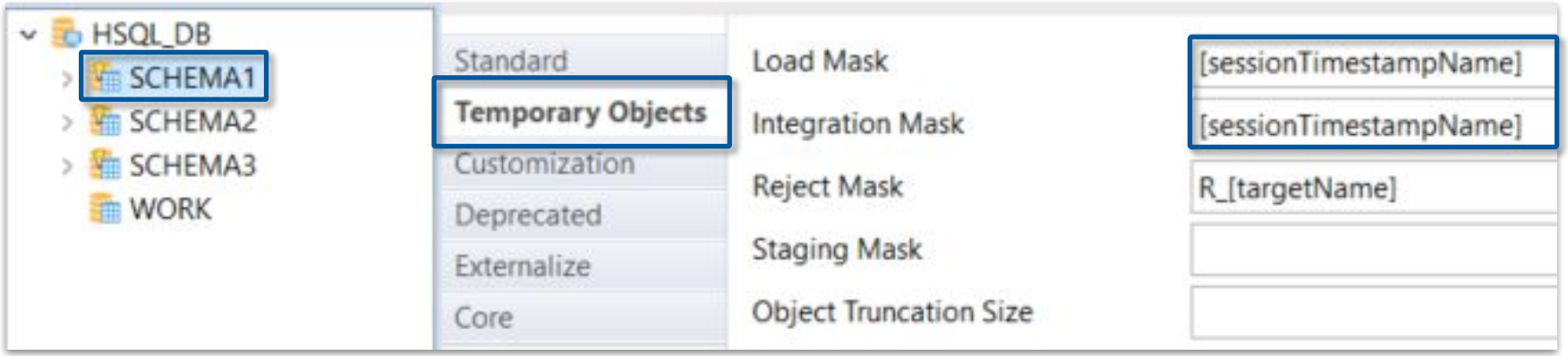
SoapUI, Postman or directly an Internet browser can be used to test the Web Services published on a Runtime



Parallel execution on Published Web Service

To allow the parallel execution of Web Services

- Change the mask of temporary objects created
  - In the metadata properties of schemas & files

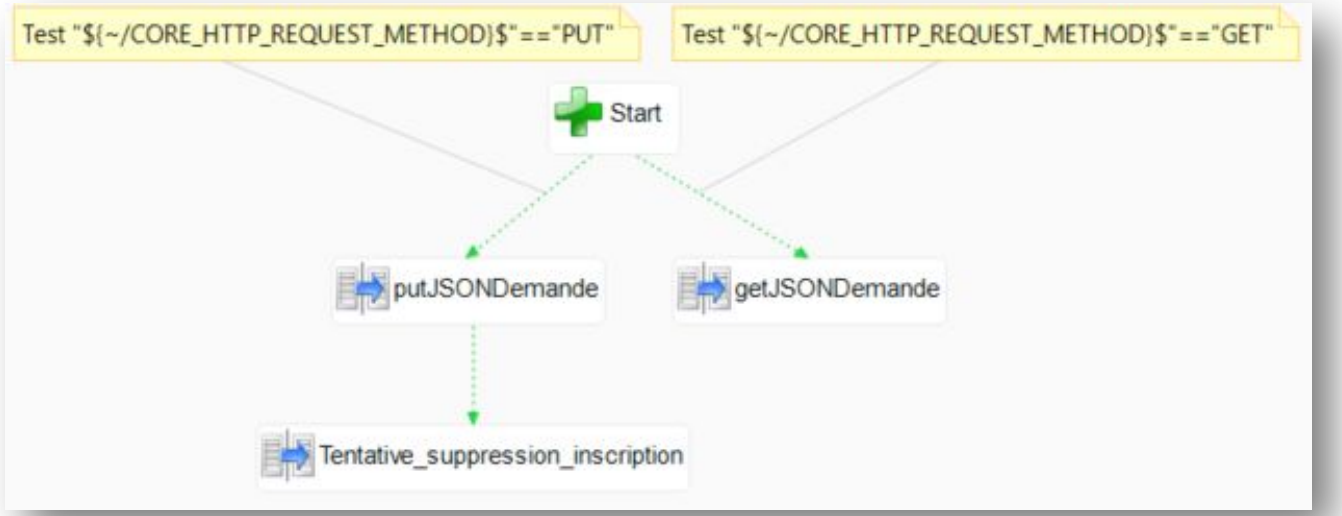


The change of masks can also be done on parallel execution of other mappings with same target tables

Test the HTTP request method in REST

It’s possible to test, inside a process, the HTTP request method in REST

- With CORE_HTTP_REQUEST_METHOD variable



To go further

Document Type	Link
French video How to use SOAP & REST Web Services	https://www.youtube.com/watch?v=66_ABJhQ7Ag
Stambia.org article HTTP REST Web Services reverse wizard	https://stambia.org/doc/224-technology-articles/web-services/configuring-the-metadata/543-http-rest-web-services-reverse-wizard
Stambia.org article Getting started with invocation error handling	https://stambia.org/doc/65-technology-articles/web-services/invoking/634-getting-started-with-invocation-error-handling
Stambia.org article Retrieving the http response code and message returned by a web service	https://stambia.org/doc/65-technology-articles/web-services/invoking/431-retrieving-the-http-response-code-and-message-returned-by-a-web-service


Semarchy
Questions?