REST services in Jolie

Ivan Lanese
Computer Science Department
University of Bologna/INRIA
Italy

REST services

- REST services are used a lot nowadays
  - Thanks to their simplicity
  - SOAP used when advanced features are needed
- You already discussed REST principles with Davide Rossi
- Here we will discuss how to use REST in Jolie
  - Jolie can invoke and expose REST services
  - We need to bridge the gap between REST resources and Jolie operations
- We need to define operations for each REST verb and tell Jolie how to extract parameters from the invocation

REST features

- REST is based on http (or https)
- Hence a REST request includes
  - An http verb (GET, PUT, POST, DELETE, …)
  - An url of a resource
  - Additional data
- A REST reply includes
  - A status code (success or failure)
  - Additional data

Exposing a Jolie service as REST

- Protocol must be http (or https)
- Data types must have void root and only first level leaves
- As parameters of the http protocol we need to specify
  - format = ”json”
  - osc (operation specific configurations): for each operation we need to specify a method (the http verb) as well as a template, giving the URL of the corresponding resource

OSCs: method and status code

- ViewUser << {
  template = "api/user/{username}"
  method = "get"
  statusCodes.UserNotFound = 404
  }
- Method refers to the HTTP verb
- Status codes are used to translate Jolie notifications of remote exceptions

OSCs: template

- ViewUser << {
  template = "api/user/{username}"
  method = "get"
  statusCodes.UserNotFound = 404
  }
- Template allows to derive the URL of the resource
- It can contain a fixed part, but also be based on parameters of the Jolie invocation
- Other parameters are turned into parameters of the REST invocation (with ?par1=val1,par2=val2 if the invocation is from a browser)

Testing the service

- We will use again soapUI
- We will use openAPI to describe the interface to soapUI
- We can automatically generate it using jolie2openapi
  - jolie2openapi service.ol port location targetFolder
  - E.g.: jolie2openapi notes.ol Notes localhost:8000 .
  - An opanAPI file named Notes.json will appear
- Create a soapUI empty project
- Right-click on the project name and select Import Swagger/OpenAPI Definition

A missing bit...

- We need to provide a file rest_template.json to jolie2openapi
- It cannot extract REST verbs from osc
- {

  "login":"method=post",

  "amount":"method=get"

  }

Invoking external services

- The mapping between operations and resources/verbs stays the same
  - We use it in the opposite direction
- Most services around require https
  - All what is said for http applies, plus one needs to set up a key store
  - See Jolie documentation, Security with SSL topic, if interested
- Most https services use the 443 port

Exercise

- Define a REST services managing notes
- You can add a new note, ask for all the notes, and delete a note (by providing an identifier of the note)
- Invoke your REST service using
  - SoapUI
  - A Jolie client