RESTful Processes
Extending the OpenAPI Specification to Enable the Documentation of RESTful Processes
Client Server Architektur

Client

Server

Funktionen

Nutzer

28.09.2018
DevCamp 2018, Berlin

Steffen Viebrock
REST - Representational State Transfer
RESTful Processes

Client → Request → Response ← Server
RESTful Processes

One Business Goal

Request 1
Response 1
Request 2
Response 2
Request 3
Response 3

Client

Server
Pay 100€ Purchase

GET /{userId}/balance

200 {balance:40}

if balance < 100

POST /{userId}/credit {amount:60}

201 {newBalance:100}

POST /{userId}/payment {amount:100}

201
RESTful Processes

Multiple connect resources, one goal

- owns the process
- decides which requests are available

- initiates the process and each request
- must know which request to send next
Hypermedia As The Engine Of Application State

- core concept of REST
- server guides client
- decouples client and server
- client must not be aware of control flow
- no need for hardcoded URIs
- increased development effort
- hardly known and implement

Pay 100€ Purchase

GET /{userId}/balance

if balance < 100

200 {balance:40, rel:credit, ref:/credit}

else

200 {balance:300, rel:pay, ref:/payment}
RESTful Processes w/o Hypermedia

- server does not guide client
- client needs to decide which requests can be send
- documentation for client developers required

Pay 100€ Purchase

200 {balance:40}

if balance < 100

POST /{userId}/credit {amount:60}

else

POST /{userId}/payment {amount:100}
RESTful Processes In Reality
Analysis of 35,000 APIs and two hypermedia APIs

• 50% simple CRUD APIs

• complex APIs implement on average 3 processes and 25 operations

• execution splits based on request/response body

• parallel execution (optional requests)

• OpenAPI Link objects are not used
Capabilities of Swagger/OpenAPI

Allows low level interface (HTTP) documentation

- request (URI, parameter, body)
- response (statuscode, body,
- authentication & authorization

Does not allow high level process documentation

- control flows
- requests dependencies/availability
Enabling Process Documentation in OpenAPI

- based on BPMN (gateways, events)
- new schema object to document processes
- does not break existing documentation/tooling
- using existing schema objects/documentation
Enabling Process Documentation in OpenAPI

- visualization based on RESTalk (Pautasso et al.)
- integration into SwaggerUI via bpmn-js and converter
What’s next

• implement the visualization

• document real world APIs with the extension

• research
  • if developers can easily write the documentation
  • if documentation and visualization helps with integration of an API

• tooling (e.g. Automatically discover processes based on URI structure)

• propose extension to community