



Creating Java JAX-RS Servers from Swagger API definitions

London Java Community Meetup, Docklands, Oct 2015

David Garry, CTO
david.garry(at)nomos-software.com



Contents

- Introduction to Swagger and REST meta-data formats
- Sample Swagger Document
- Editing Swagger Document using Swagger Editor
- Code generation using Swagger CodeGen
- Swagger Documentation Generation using Swagger-Core
- Usage of Swagger Editor at Nomos
- Swagger Hub



Introduction to Swagger



Swagger (<http://swagger.io/>) is a project used to describe and document RESTful APIs.

Swagger project is open source and sponsored by **SMARTBEAR**

- Swagger Specification - <http://swagger.io/specification/>
- Swagger Tooling - <http://swagger.io/tools/>
- Swagger Community - <http://swagger.io/community/>

Introduction to Swagger

API Description Languages	Mainly uses XML Schema	Mainly uses JSON Schema	
<i>WADL</i>	✓		not used much
<i>Swagger</i>		✓	Sponsored by Smartbear, largest community, lots of tools
<i>RAML</i>		✓	Sponsored by Mulesoft
<i>API Blueprint</i>		✓	Sponsored by Apiary

API Description Languages - Alternatives



Introduction to Swagger

Tool	Description
Swagger Core	Java-related libraries for generating and reading Swagger definitions
Swagger Codegen	Command-line tool for generating both client and server side code from a Swagger definition
Swagger UI	Browser based UI for exploring a Swagger defined API
Swagger Editor	Browser based editor for authoring Swagger definitions in YAML or JSON format



Core Swagger Tools

<https://github.com/swagger-api>

https://github.com/swagger-api/swagger-codegen/tree/develop_2.0

<https://github.com/swagger-api/swagger-ui>

<https://github.com/swagger-api/swagger.io/blob/wordpress/tools/swagger-editor.md>

<https://github.com/swagger-api/swagger-core>

Sample Swagger Document – TMForum Product Mgt API

```
version: 14.5.0
title: AMKB Cloud RI - TMF Product Order Management REST API
termsOfService: 'http://amkbcloud.com/terms'
contact:
  name: support@amkbcloud.com
license:
  name: Apache 2.0
  url: 'http://www.apache.org/licenses/LICENSE-2.0.html'
host: '12.208.99.92:10006'
basePath: /productOrderManagement
schemes:
  - http
consumes:
  - application/json
produces:
  - application/json
paths:
  /productOrder:
    post: ➔
    get: ➔
  '/productOrder/{id}': ➔
  /productOrder/hub: ➔
  '/productOrder/hub/{id}': ➔
definitions:
  BundledProductOffering:
    description: 'May be a bundle ProductOffering, in this case, it will contain the list of bundled offers that are ordered.'
    required:
      - href
    properties:
      href:
        description: Reference of the product offering.
        type: string
      bundledProductOffering:
        description: Bundles that make up the bundled product offering.
        type: array
```

The paths / operations

POST product Order
GET product Order

The data model for API request parameters and responses

Based on JSON schema



Editing Swagger Document using Swagger Editor

The screenshot shows the Swagger Editor interface with a Swagger document for a Petstore API. The left pane displays the raw Swagger JSON code, and the right pane shows the generated API documentation and security settings.

Swagger Document (Left Pane):

```
swagger: '2.0'
info:
  description: >
    This is a sample server.
    It can be fully configured
    at http://editor.swagger.io/ or join the IRC channel '#swagger'.
  version: 1.0.0
  title: Swagger Petstore
  termsOfService: 'http://heloreverb.com/terms/'
  contact:
    name: apiteam@swagger.io
  license:
    name: Apache 2.0
    url: 'http://www.apache.org/licenses/LICENSE-2.0.html'
host: petstore.swagger.io
basePath: /v2
schemes:
  - http
paths:
  /pets:
    post:
      tags:
        - pet
      summary: Add a new pet to the store
      description: ''
      operationId: addPet
      consumes:
        - application/json
        - application/xml
      produces:
        - application/json
        - application/xml
      parameters:
        - in: body
          name: body
          description: Pet object that needs to be added to the store
          required: false
          schema:
            $ref: '#/definitions/Pet'
      responses:
```

Generated JAX-RS Code (Top Left):

```
JAX-RS
Inflector
Node.js
Scalatra
Silex PHP
Sinatra
Spring MVC
```

Swagger Petstore (Right Pane):

This is a sample server Petstore server.
Learn about Swagger or join the IRC channel `#swagger` on irc.freenode.net.
For this sample, you can use the api key `special-key` to test the authorization filters
Version 1.0.0
Contact information
apiteam@swagger.io
Terms of service
<http://heloreverb.com/terms/>
License
Apache 2.0

Security (Bottom Right):

api_key (API Key) Authenticate

Name	api_key
In	header

petstore_auth (OAuth 2.0) Authenticate

Flow	implicit
Authorization URL	http://petstore.swagger.io/api/oauth/dialog

Scopes

Show all downloads... 17.5/17.5 KB, 0 secs left

Play with the editor at <http://editor.swagger.io/#/>

Code Generation using Swagger Codegen

PetsApi.java StoresApi.java UsersApi.java

```

package io.swagger.api;

import io.swagger.model.*;

@Path("/users")

@io.swagger.annotations.Api(value = "/users", description = "the u")
@javax.annotation.Generated(value = "class io.swagger.codegen.lan
public class UsersApi {

    private final UsersApiService delegate = Users ApiServiceFactory
        .createUsersApiService();

    @POST

    @Produces({ "application/json", "application/xml" })
    @io.swagger.annotations.ApiOperation(value = "Create user", no
    @io.swagger.annotations.ApiResponses(value = {
        @io.swagger.annotations.ApiResponse(code = 200, message =
            "Successful operation"
    })

    public Response createUser(@ApiParam(value = "Created user obj
        throws NotFoundException {
            return delegate.createUser(body);
        }
    }

    @POST
    @Path("/createWithArray")

    @Produces({ "application/json", "application/xml" })
    @io.swagger.annotations.ApiOperation(value = "Creates list of
    @io.swagger.annotations.ApiResponses(value = {
        @io.swagger.annotations.ApiResponse(code = 200, message =
            "Successful operation"
    })

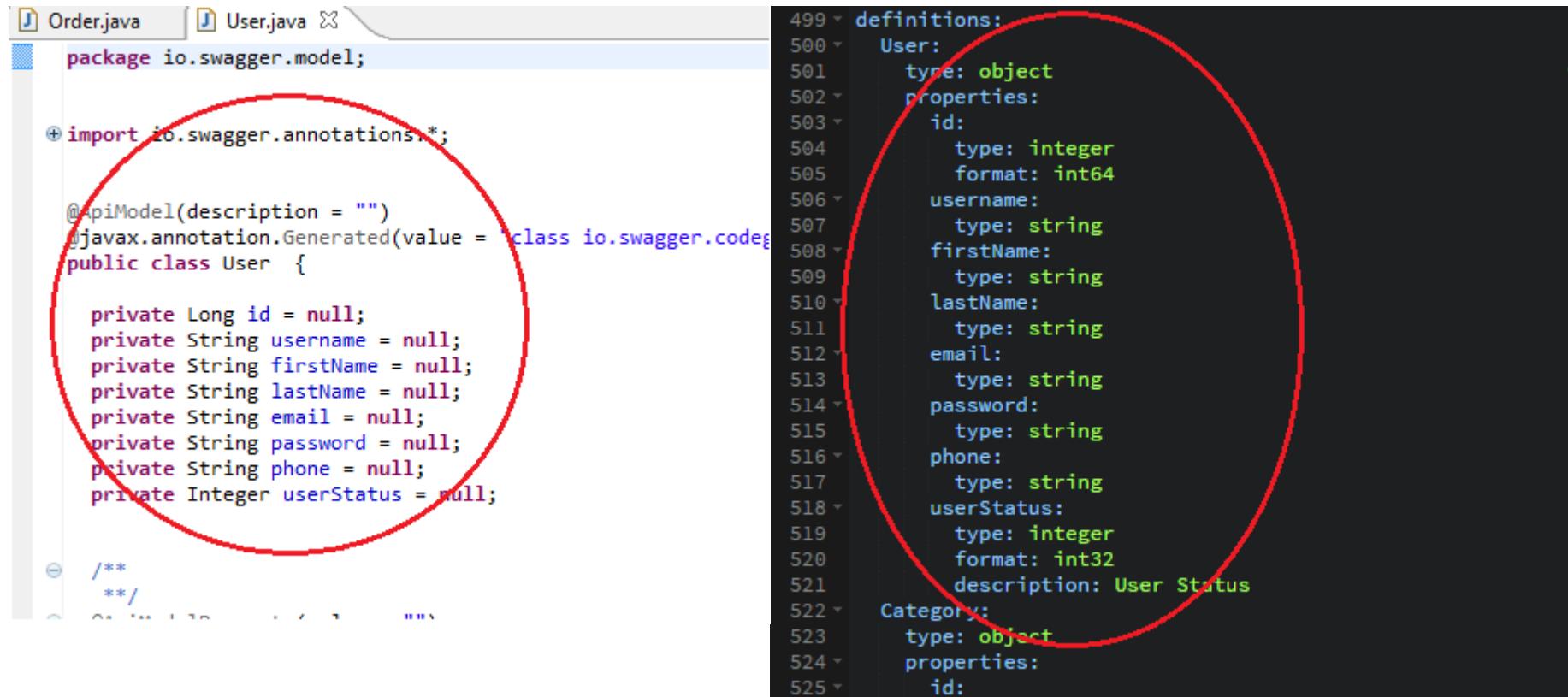
    public Response createUsersWithArrayInput(@ApiParam(value = "L
        throws NotFoundException {
            return delegate.createUsersWithArrayInput(body);
        }
    }

    @POST
    @Path("/createWithList")
}

```

11
12
13 For this sample, you can use the api key `special-key` to test
14 authorization filters
15 version: 1.0.0
16 title: Swagger Petstore
17 termsOfService: 'http://helloreverb.com/terms/'
18 contact:
19 name: apiteam@swagger.io
20 license:
21 name: Apache 2.0
22 url: 'http://www.apache.org/licenses/LICENSE-2.0.html'
23 host: petstore.swagger.io
24 basePath: /v2
25 schemes:
26 - http
27 paths:
28 /pets:
29 /pets/findByStatus:
30 /pets/findByTags:
31 '/pets/{petId}':
32 /stores/order:
33 '/stores/order/{orderId}':
34 /users:
35 '/users/createWithArray':
36 '/users/createWithList':
37 '/users/login':
38 '/users/logout':
39 '/users/{username}':
40 securityDefinitions:
41 api_key:
42 type: apiKey
43 name: api_key
44 in: header

Code Generation using Swagger Codegen



The screenshot shows two side-by-side code editors. The left editor displays a Java file named `User.java` with annotations for Swagger code generation. The right editor displays the corresponding `definitions` section from a Swagger JSON file.

User.java (Left):

```
package io.swagger.model;

import io.swagger.annotations.*;

@ApiModel(description = "")
@javax.annotation.Generated(value = "class io.swagger.codegen.languages.JavaClientCodegen", name = "User")
public class User {

    private Long id = null;
    private String username = null;
    private String firstName = null;
    private String lastName = null;
    private String email = null;
    private String password = null;
    private String phone = null;
    private Integer userStatus = null;

    /**
     */
}
```

Definitions (Right):

```
499  definitions:
500    User:
501      type: object
502      properties:
503        id:
504          type: integer
505          format: int64
506        username:
507          type: string
508        firstName:
509          type: string
510        lastName:
511          type: string
512        email:
513          type: string
514        password:
515          type: string
516        phone:
517          type: string
518        userStatus:
519          type: integer
520          format: int32
521          description: User Status
522        Category:
523          type: object
524          properties:
525            id:
```

Data model generated from Definitions Section



Swagger Documentation Generation using Swagger-Core

```
@io.swagger.annotations.Api(value = "/pets", description = "the pets API")
@javax.annotation.Generated(value = "class io.swagger.codegen.languages.JaxRSServerCodegen", date = "2015-10-09T15
public class PetsApi {

    private final PetsApiService delegate = PetsApiServiceFactory.getPetApi();

    @PUT

        @Consumes({ "application/json", "application/xml" })
        @Produces({ "application/json", "application/xml" })
        @io.swagger.annotations.ApiOperation(value = "Update an existing pet", notes = "", response = Void.class)
        @io.swagger.annotations.ApiResponses(value = {
            @io.swagger.annotations.ApiResponse(code = 400, message = "Invalid ID supplied", response = Void.class),
            @io.swagger.annotations.ApiResponse(code = 404, message = "Pet not found", response = Void.class),
            @io.swagger.annotations.ApiResponse(code = 405, message = "Validation exception", response = Void.class) })

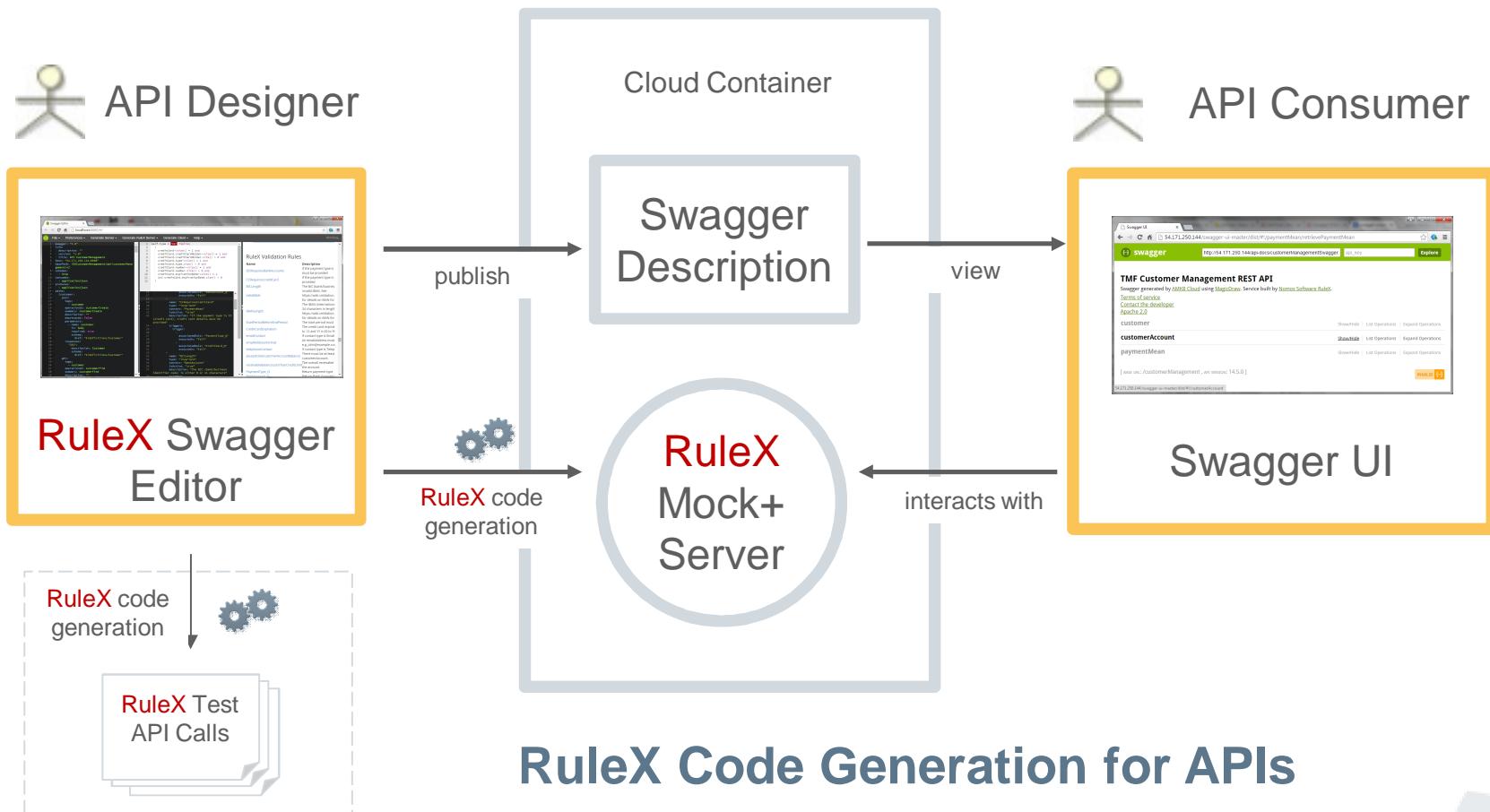
    public Response updatePet(@ApiParam(value = "Pet object that needs to be added to the store" ) Pet body)
        throws NotFoundException {
        return delegate.updatePet(body);
    }

    @POST

        @Consumes({ "application/json", "application/xml" })
        @Produces({ "application/json", "application/xml" })
        @io.swagger.annotations.ApiOperation(value = "Add a new pet to the store", notes = "", response = Void.class)
        @io.swagger.annotations.ApiResponses(value = {
            @io.swagger.annotations.ApiResponse(code = 405, message = "Invalid input", response = Void.class) })
```

Swagger-core exposes JAX-RS providers and one JAX-RS resource to generate and serve the Swagger API documentation

Usage of Swagger at Nomos



RuleX Code Generation for APIs

More on RuleX Swagger Editor : <http://nomos-software.com/blog/rulex-swagger-editor>
 Available on github at <https://github.com/nomos-software/swagger-editor-rulex>



SwaggerHub : <https://swaggerhub.com/>

Hosted environment for your swagger : launched by Smartbear to provide Collaboration and Coordination across API Lifecycle

The screenshot shows the SwaggerHub homepage. At the top, there's a navigation bar with tabs for "Swagger UI" and "SwaggerHub". The address bar shows the URL <https://swaggerhub.com>. The main content area has a green header with the "SWAGGERhub" logo. Below the header, there are four numbered cards:

- Create / Import an API**: Build an API from scratch in our Editor or bring in an existing one.
- Collaborate with your team**: Invite others to define and refine your API.
- Generate code**: Use our server and client code templates to generate code based on your Swagger definition.
- Publish & Version your API**: Publish finished APIs and create new versions to continue enhancing it.

Below these cards, a message says "You have 1 API". A dropdown menu shows "Sort by: Name ▾ Descending ▾". Underneath, a card for "triciabalf / CustomerMgt" is displayed, showing its version "14.5.0", "No Tags", and a note: "Swagger generated by AMKB Cloud using MagicDraw. Service built by Nomos Software RuleX". It also includes a "Download..." button and icons for GitHub and Docker.



About Nomos Software



Software Vendor

Our RuleX product
supports Swagger



API Consultancy
Services

Design, Test, Deploy,
Document, API Tools
Ecosystem etc

Contact: david.garry@nomos-software.com

(slides available at <http://www.slideshare.net/nomossoftware>)