



# 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



- 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



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/>

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
<a href="#">Swagger Core</a>	Java-related libraries for generating and reading Swagger definitions
<a href="#">Swagger Codegen</a>	Command-line tool for generating both client and server side code from a Swagger definition
<a href="#">Swagger UI</a>	Browser based UI for exploring a Swagger defined API
<a href="#">Swagger Editor</a>	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-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>

```
4  version: 14.5.0
5  title: AMKB Cloud RI - TMF Product Order Management REST API
6  termsOfService: 'http://amkbcloud.com/terms'
7  contact:
8    name: support@amkbcloud.com
9  license:
10   name: Apache 2.0
11   url: 'http://www.apache.org/licenses/LICENSE-2.0.html'
12 host: '12.208.99.92:10006'
13 basePath: /productOrderManagement
14 schemes:
15   - http
16 consumes:
17   - application/json
18 produces:
19   - application/json
20 paths:
21   /productOrder:
22     post:
23     get:
24     '/productOrder/{id}':
25     /productOrder/hub:
26     '/productOrder/hub/{id}':
27 definitions:
28   BundledProductOffering:
29     description: 'May be a bundle ProductOffering, in this case, it will contain the list of
30       bundled offers that are ordered.'
31     required:
32     - href
33     properties:
34     href:
35     description: Reference of the product offering.
36     type: string
37     bundledProductOffering:
38     description: Bundles that make up the bundled product offering.
```

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 displays the Swagger Editor interface. On the left, a Swagger JSON document is shown in a dark-themed editor. The 'Generate Server' dropdown menu is open, with 'JAX-RS' selected. The JSON document includes fields like 'swagger: 2.0', 'info', 'description', 'version: 1.0.0', 'title: Swagger Petstore', 'termsOfService', 'contact', 'license', 'host', 'basePath', 'schemes', 'paths', and 'definitions'. A download bar at the bottom shows 'jaxrs-server-generated.zip' with a size of 7.5/17.5 KB and 0 secs left. On the right, the rendered Swagger UI is displayed, showing the 'Swagger Petstore' title, a description, contact information, terms of service, license, and security sections. The security section includes 'api\_key (API Key)' and 'petstore\_auth (OAuth 2.0)'. The 'api\_key' section has a table with columns 'Name' and 'In', and rows for 'api\_key' and 'header'. The 'petstore\_auth' section has a table with columns 'Flow' and 'Authorization URL', and rows for 'implicit' and 'http://petstore.swagger.io/api/oauth/dialog'. The 'Scopes' section is also visible.

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

# Code Generation using Swagger Codegen

```

PetsApi.java StoresApi.java UsersApi.java X
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.lang
public class UsersApi {

    private final UsersApiService delegate = UsersApiServiceFactory

    @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 =

    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 =

    public Response createUsersWithArrayInput(@ApiParam(value = "u
    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://helloverb.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

```



```
Order.java | User.java ✕
package io.swagger.model;

import io.swagger.annotations.*;

@ApiModel(description = "")
@javax.annotation.Generated(value = "class io.swagger.codegen")
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;

    /**
     **/
}

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

```
@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.getPetsApi();

    @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) })

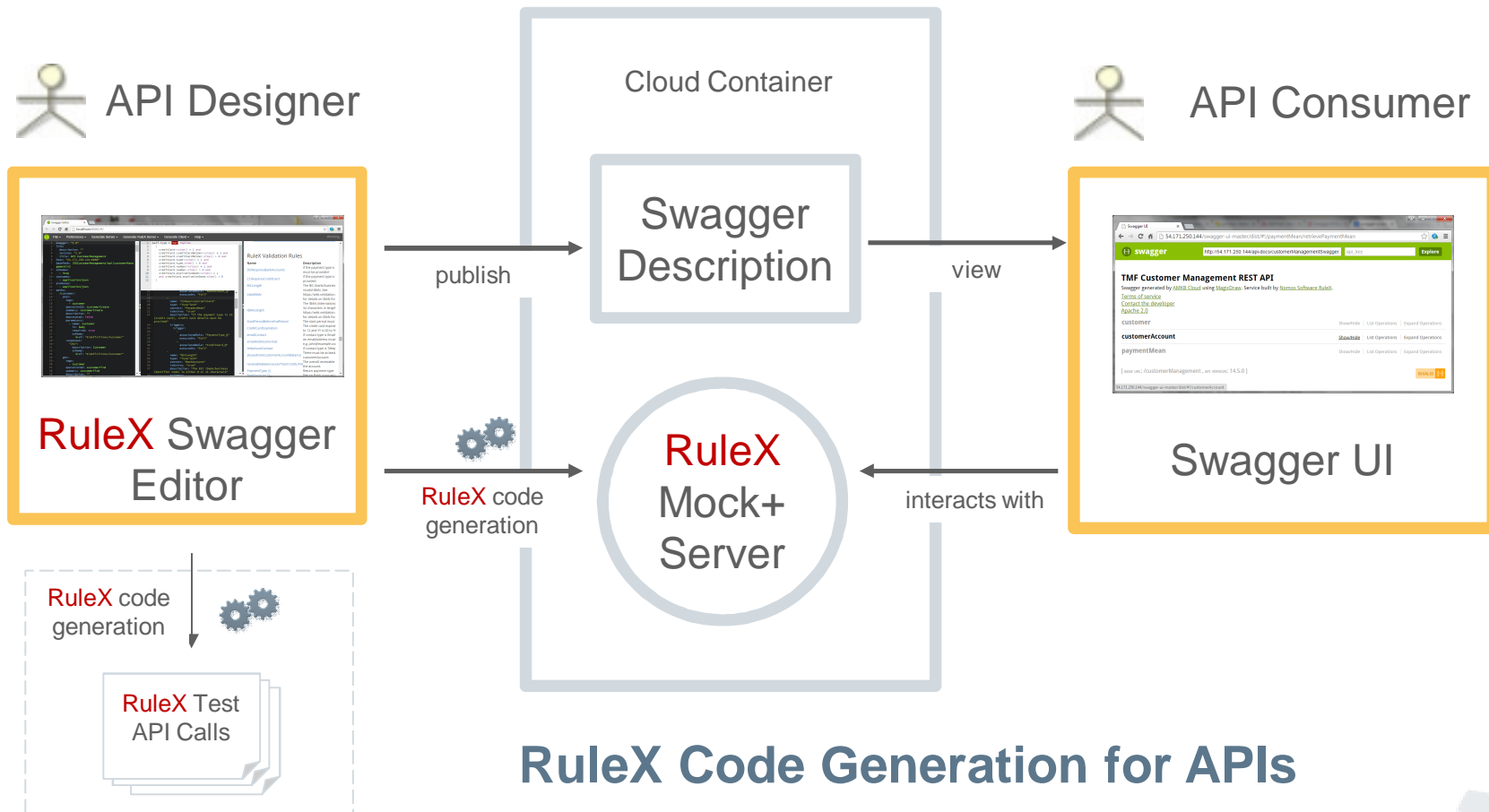
    public Response updatePet(@ApiParam(value = "Pet object that needs to be added to the store" ) Pet body)
    throws NotFoundException {
        return delegate.updatePet(body);
    }

    @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

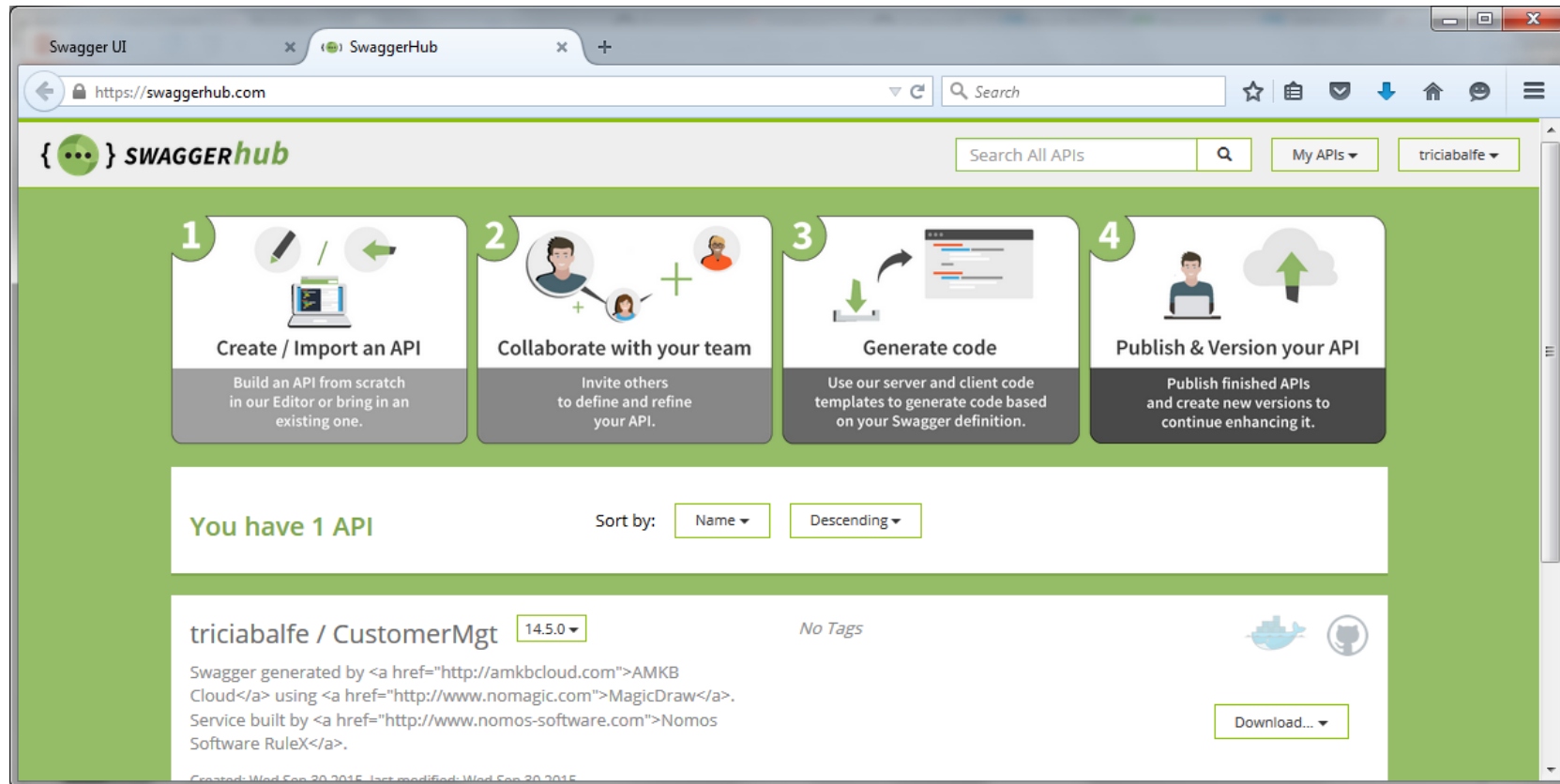


More on RuleX Swagger Editor : <http://nomos-software.com/blog/rulex-swagger-editor>  
 Available on github at <https://github.com/nomos-software/swagger-editor-rulex>



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

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





## 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](mailto:david.garry@nomos-software.com)

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

