

Design & Test First Workflow For REST APIs

How can REST API specification projects help us?

“hand copy” workflow

 design

 code

 test

 document

“hand copy” workflow

design ≠ code ≠ test ≠ document



:(

need for a better workflow

less error prone

easier specification implementation

specification can be changed

REST API specification projects



api blueprint



raml



swagger

helpful tools

designing

coding

testing

documenting

designing

web - / text - editors

syntax highlighted

wysiwym

validator

coding

generate

server

client

models

testing






write client & server tests
validate implementation
import in tools e.g. postman

documenting

dynamic

static

tools are here to help

-  → generate test server
-  → generate test client
-  → code
-  → validate implementation
-  → document

“tooled” workflow

 design

 test

 code

 document

“tooled” workflow

design = test = code = document



:)

swagger as an example

swagger spec

swagger editor

swagger codegen

swagger ui

swagger spec

write in editor

generate from annotated code

put into version control system

use as exchange document

swagger spec

```
1  swagger: '2.0'
2  info:
3    version: 1.0.0
4    title: Swagger Petstore (Simple)
5    description: A sample API that uses a petstore as an example to
6      demonstrate features in the swagger-2.0 specification
7    termsOfService: 'http://helloverb.com/terms/'
8  contact:
9    name: Swagger API team
10   email: foo@example.com
11   url: 'http://swagger.io'
12  license:
13    name: MIT
14    url: 'http://opensource.org/licenses/MIT'
15  host: petstore.swagger.io
16  basePath: /api
17  schemes:
18    - http
19  consumes:
20    - application/json
21  produces:
22    - application/json
23  paths:
```


swagger spec

```
21   - application/json
22  paths:
23    /pets:
24      get:
25        description: Returns all pets from the system that the user has
26                     access to
27        operationId: findPets
28        produces:
29          - application/json
30          - application/xml
31          - text/xml
32          - text/html
33        parameters:
34          - name: tags
35            in: query
36            description: tags to filter by
37            required: false
38            type: array
39            items:
40              type: string
41              collectionFormat: csv
42          - name: limit
```

swagger editor

The image shows the Swagger Editor interface. On the left is a code editor with a Swagger 2.0 specification for a petstore API. On the right is the rendered API documentation page.

```
1 swagger: '2.0'
2 info:
3   version: 1.0.0
4   title: Swagger Petstore (Simple)
5   description: A sample API that uses a petstore as an example to
6     demonstrate features in the swagger-2.0 specification
7   termsOfService: 'http://helloverb.com/terms/'
8   contact:
9     name: Swagger API team
10    email: foo@example.com
11    url: 'http://swagger.io'
12  license:
13    name: MIT
14    url: 'http://opensource.org/licenses/MIT'
15  host: petstore.swagger.io
16  basePath: /api
17  schemes:
18  - http
19  consumes:
20  - application/json
21  produces:
22  - application/json
23  paths:
24  /pets:
25    get:
26      description: Returns all pets from the system that the user has
27        access to
28      operationId: findPets
29      produces:
30      - application/json
31      - application/xml
32      - text/xml
33      - text/html
34      parameters:
35      - name: tags
```

Swagger Petstore (Simple)

A sample API that uses a petstore as an example to demonstrate features in the swagger-2.0 specification

Version 1.0.0

Contact information
Swagger API team

foo@example.com
<http://swagger.io>

Terms of service
<http://helloverb.com/terms/>

License
MIT

Paths

/pets

GET /pets

Description
Returns all pets from the system that the user has access to

Parameters

```
File ▾ Preferences ▾ Generate Server ▾ Generate Client ▾ Help ▾
58 ▾ post:
59 ▾   description: Creates a new pet in the store. Duplicates are allowed
60 ▾   operationId: addPet
61 ▾   produces:
62 ▾     - application/json
63 ▾   parameters:
64 ▾     - name: pet
65 ▾       in: body
66 ▾       description: Pet to add to the store
67 ▾       required: true
68 ▾       schema:
69 ▾         $ref: '#/definitions/newPet'
70 ▾   responses:
71 ▾     '200':
72 ▾       description: pet response
73 ▾       schema:
74 ▾         $ref: '#/definitions/pet'
75 ▾     default:
76 ▾       description: unexpected error
77 ▾       schema:
78 ▾         $ref: '#/definitions/errorModel'
79 ▾ '/pets/{id}':
80 ▾   get:
81 ▾     description: 'Returns a user based on a single ID, if the user does
82 ▾       not have access to the pet'
83 ▾     operationId: findPetById
84 ▾     produces:
85 ▾       - application/json
86 ▾       - application/xml
87 ▾       - text/xml
88 ▾       - text/html
89 ▾     parameters:
90 ▾       - name: id
91 ▾         in: path
92 ▾         description: ID of pet to fetch
93 ▾         required: true
94 ▾         type: integer
95 ▾         format: int64
96 ▾     responses:
97 ▾       '200':
98 ▾         description: pet response
99 ▾         schema:
100 ▾           $ref: '#/definitions/pet'
```

POST /pets

Description

Creates a new pet in the store. Duplicates are allowed

Parameters

| Name | Located in | Description | Required | Schema |
|------|------------|-------------------------|----------|------------------------------------------------------------------------------|
| pet | body | Pet to add to the store | Yes | <pre> newPet { id: integer (int64) name: string * tag: string } </pre> |

Responses

| Code | Description | Schema |
|---------|------------------|-----------------------------------------------------------------------------|
| 200 | pet response | <pre> pet { id: integer * (int64) name: string * tag: string } </pre> |
| default | unexpected error | <pre> errorModel { code: integer * (int32) message: string * } </pre> |

Try this operation

/pets/{id}

swagger codegen

possible targets

android, async-scala, csharp, dart, flash, java, jaxrs, inflector, nodejs, objc, perl, php, python, python3, qt5cpp, retrofit, ruby, scala, scalatra, silex-PHP, sinatra, spring-mvc, dynamic-html, html, swagger, swagger-yaml, swift, tizen, typescript-angular, typescript-node, akka-scala, CsharpDotNet2

swagger codegen

implement generated interfaces
use generated clients in tests
get to know different frameworks

swagger ui

reads swagger.json

dynamic html page

try out

swagger ui

The screenshot shows the Swagger UI interface. At the top, there is a green header bar with the Swagger logo on the left, a text input field containing the URL 'http://petstore.swagger.io/v2/swagger.json', another text input field labeled 'api_key', and an 'Explore' button on the right. Below the header, the main content area has a title 'Swagger Petstore' followed by a paragraph of introductory text. There are three links: 'http://swagger.io', 'Contact the developer', and 'Apache 2.0'. Below this is a section titled 'pet : Everything about your Pets' with three sub-headers: 'Show/Hide', 'List Operations', and 'Expand Operations'. The main part of the interface is a list of API endpoints, each with a colored header indicating the HTTP method and a description of the operation.

| Method | Endpoint | Description |
|--------|-------------------|----------------------------|
| POST | /pet | Add a new pet to the store |
| PUT | /pet | Update an existing pet |
| GET | /pet/findByStatus | Finds Pets by status |
| GET | /pet/findByTags | Finds Pets by tags |
| DELETE | /pet/{petId} | Deletes a pet |
| GET | /pet/{petId} | Find pet by ID |

swagger ui

<http://swagger.io>
[Contact the developer](#)
[Apache 2.0](#)

pet : Everything about your Pets

Show/Hide | List Operations | Expand Operations

| | | |
|--------|--------------------------|-------------------------------------------|
| POST | /pet | Add a new pet to the store |
| PUT | /pet | Update an existing pet |
| GET | /pet/findByStatus | Finds Pets by status |
| GET | /pet/findByTags | Finds Pets by tags |
| DELETE | /pet/{petId} | Deletes a pet |
| GET | /pet/{petId} | Find pet by ID |
| POST | /pet/{petId} | Updates a pet in the store with form data |
| POST | /pet/{petId}/uploadImage | uploads an image |

store : Access to Petstore orders

Show/Hide | List Operations | Expand Operations

user : Operations about user

Show/Hide | List Operations | Expand Operations

swagger ui

DELETE /pet/{petId} Deletes a pet

GET /pet/{petId} Find pet by ID

Implementation Notes
Returns a single pet

Response Class (Status 200) OFF 

Model | Model Schema

```
{
  "id": 0,
  "category": {
    "id": 0,
    "name": "string"
  },
  "name": "doggie",
  "photoUrls": [
    "string"
  ],
  "tags": [
```

Response Content Type

Parameters

| Parameter | Value | Description | Parameter Type | Data Type |
|-----------|-------|-------------|----------------|-----------|
|-----------|-------|-------------|----------------|-----------|

swagger ui

```
{
  "id": 0,
  "category": {
    "id": 0,
    "name": "string"
  },
  "name": "doggie",
  "photourls": [
    "string"
  ],
  "tags": [
```

Response Content Type

Parameters

| Parameter | Value | Description | Parameter Type | Data Type |
|--------------|-----------------------------------------|----------------------------|----------------|-----------|
| petId | <input type="text" value="(required)"/> | ID of pet to return | path | long |

Response Messages

| HTTP Status Code | Reason | Response Model | Headers |
|------------------|---------------------|----------------|---------|
| 400 | Invalid ID supplied | | |
| 404 | Pet not found | | |

“tooled” workflow

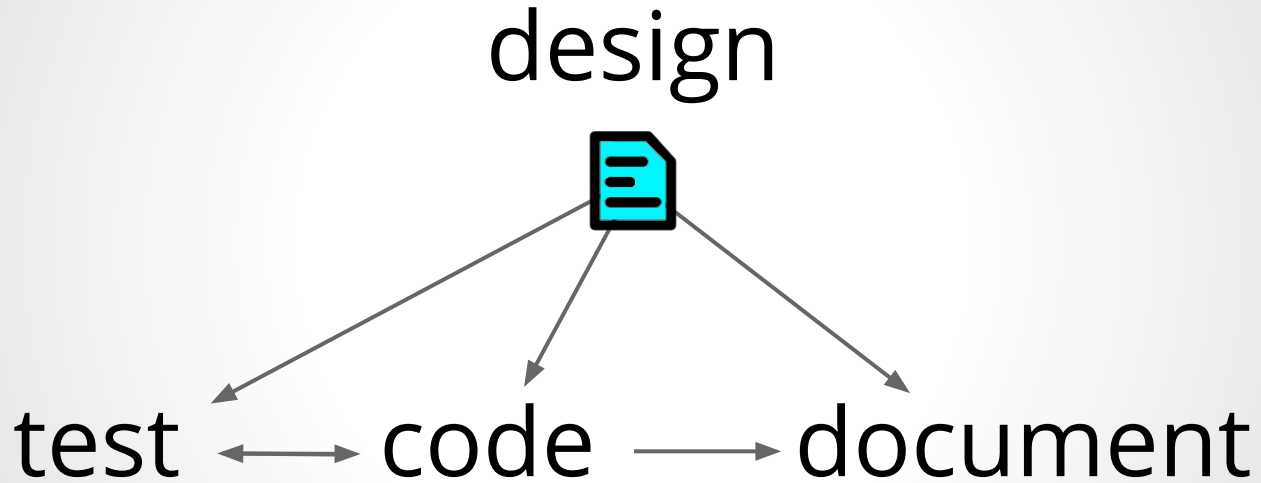
 design

 test





 code

 document

artefact relations



automation

- design  swagger-validation
- test 
- code  assertj-swagger
- document  swagger-codegen

“tooled” workflow

design = test = code = document



:)

“design and test first” - workflow

design = **test** = code = document



Thank you for listening

Questions?

References

examples are taken from swagger.io

<http://petstore.swagger.io/>

<http://editor.swagger.io/>

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

related blog posts

<http://swagger.io/writing-apis-with-the-swagger-inflector/>

<http://blog.raml.org/guest-post-api-first-with-raml-development-and-documentation-of-rest-apis/>