



Documenting RESTful APIs

Andy Wilkinson, Pivotal
@ankinson

How RESTful is your API?

REST maturity model



<http://www.crummy.com/writing/speaking/2008-QCon/act3.html>
<http://martinfowler.com/articles/richardsonMaturityModel.html>

How RESTful do you want it to be?

Making your API easy to document

Consistency

HTTP status codes
HTTP verbs

Uniqueness

Use distinct relations on your links

```
{  
  "_links" : {  
    "notes" : {  
      "href" : "http://localhost:8080/notes"  
    },  
    "tags" : {  
      "href" : "http://localhost:8080/tags"  
    }  
  }  
}
```


What should you document?

Resources

What do they represent?

What input do they accept?

What output do they produce?

Links

Where can you go?

What will you find when you get there?

Cross-cutting concerns

HTTP status codes

HTTP verbs (PUT vs PATCH)

Authentication and authorisation

Rate limiting

...

What shouldn't you document?

URIs

What does it look when you get it wrong?

Learning from my mistakes

[GemFire](#)
[RabbitMQ](#)
[SQLFire](#)
[tc Server](#)
[vFabric](#)
[Web Server](#)
[Expand All](#)
[Collapse All](#)

GET	/tc-server/v1/	The main entry point to the API
-----	----------------	---------------------------------

GET	/tc-server/v1/groups/	List all groups
-----	-----------------------	-----------------

[READ](#)

POST	/tc-server/v1/groups/	Create a group
------	-----------------------	----------------

[WRITE](#)

GET	/tc-server/v1/groups/{}/	Detail about a group
-----	--------------------------	----------------------

[READ](#)

POST	/tc-server/v1/groups/{}/	Update the membership of a group
------	--------------------------	----------------------------------

[WRITE](#)

DELETE	/tc-server/v1/groups/{}/	Delete a group
--------	--------------------------	----------------

[WRITE](#)

GET	/tc-server/v1/groups/{}/installations/	List all installations in the group
-----	--	-------------------------------------

[READ](#)

POST	/tc-server/v1/groups/{}/installations/	Create an installation
------	--	------------------------

[WRITE](#)

And when you get it right?

Overview

This describes the resources that make up the official GitHub API v3. If you have any problems or requests please contact [support](#).

- i. [Current Version](#)
- ii. [Schema](#)
- iii. [Parameters](#)
- iv. [Root Endpoint](#)
- v. [Client Errors](#)
- vi. [HTTP Redirects](#)
- vii. [HTTP Verbs](#)

RESTful API documentation tools

Swagger

<http://swagger.io>

 **[martypitt/swagger-springmvc](https://github.com/martypitt/swagger-springmvc)**

It doesn't support hypermedia

Consider hypermedia for resource listing #97

 **Closed**

rgarcia opened this issue on Jan 2, 2013 · 8 comments



fehguay commented on Aug 30

closing as not per design

It's URI-centric

/pet : Operations about pets

[Show/Hide](#)[List Operations](#)[Expand Operations](#)[Raw](#)**PUT** /pet

Update an existing pet

POST /pet

Add a new pet to the store

POST /pet/{petId}

Updates a pet in the store with form data

GET /pet/{petId}

Find pet by ID

DELETE /pet/{petId}

Deletes a pet

It's leaky

Response Class

Model | Model Schema

ResponseEntity«Album» {

```
statusCode (string, optional) = ['100' or '101' or '102' or '103' or '200' or '201' or  
'202' or '203' or '204' or '205' or '206' or '207' or '208' or '226' or '300' or '301' or  
'302' or '302' or '303' or '304' or '305' or '307' or '308' or '400' or '401' or '402' or  
'403' or '404' or '405' or '406' or '407' or '408' or '409' or '410' or '411' or '412' or  
'413' or '414' or '415' or '416' or '417' or '418' or '419' or '420' or '421' or '422' or  
'423' or '424' or '426' or '428' or '429' or '431' or '500' or '501' or '502' or '503' or  
'504' or '505' or '506' or '507' or '508' or '509' or '510' or '511'],  
headers (HttpHeaders, optional),  
body (Album, optional)  
}
```

It's huge

Spring Boot
Tomcat
Hibernate
25MB

Swagger
31MB

HAL and HAL Browser

http://stateless.co/hal_specification.html



mikekelly/hal-browser

A new alternative

 **wilkinsona/spring-restdocs**

**Write as much as
possible in a format that's
designed for writing**

**Don't use the
implementation to provide
the documentation**

**Provide some guarantees
that the documentation
is accurate**

Asciidoctor

Spring MVC Test

Maven or Gradle

 wilkinsona/spring-restdocs

@Test

```
public void indexExample() throws Exception {  
1  this.mockMvc.perform(get("/"))  
2    .andExpect(status().isOk())  
3    .andExpect(jsonPath("_links.notes",  
4                          is(notNullValue())))  
5    .andExpect(jsonPath("_links.tags",  
                          is(notNullValue())))  
5    .andDo(document("index-example"));  
}
```

[source,http]

HTTP/1.1 200 OK

Content-Type: application/hal+json

```
{
  "_links" : {
    "tags" : {
      "href" : "http://localhost:8080/tags"
    },
    "notes" : {
      "href" : "http://localhost:8080/notes"
    }
  }
}
```

Example response

HTTP

HTTP/1.1 200 OK

Content-Type: application/hal+json

```
{
  "_links" : {
    "notes" : {
      "href" : "http://localhost:8080/notes"
    },
    "tags" : {
      "href" : "http://localhost:8080/tags"
    }
  }
}
```

@Test

```
public void indexExample() throws Exception {  
    1 this.mockMvc.perform(get("/"))  
        .andExpect(status().isOk())  
    2 .andDo(document("index-example").withLinks(  
    3     linkWithRel("notes").description(  
        "The <<resources-notes,Notes resource>>"),  
    4     linkWithRel("tags").description(  
        "The <<resources-tags,Tags resource>>"))));  
}
```

|===

Relation	Description
----------	-------------

notes	
-------	--

The <<resources-notes,Notes resource>>	
--	--

tags	
------	--

The <<resources-tags,Tags resource>>	
--------------------------------------	--

|===

Links

Relation	Description
notes	The Notes resource
tags	The Tags resource

Please try it out

I'd love your feedback

 **wilkinsona/spring-restdocs**

Thank you



@ankinson



wilkinsona



awilkinson@pivotal.io