Swagger 2.0, a Primer

Swagger
A Primer

Introduction

This document covers only the basic offerings of Swagger.

Swagger has feature/functions that overlap the features that HATEOAS provides. Whether we can even benefit from HATEOAS is an open question.

Swagger Conventions

There are a few Swagger coding conventions to observe. These are invariants which, if missing raise runtime errors which resemble the following:

Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Wed Nov 13 09:08:52 CST 2019
There was an unexpected error (type=Not Found, status=404).
No message available

The Conventions

A. Line 52 overrides the `addResourceHandlers()` method. Within that method, a pair of resource handlers, one of which (Lines 54-6) must be added to the `ResourceHandlerRegistry` instance.

The registry is passed as a formal argument into the `addResourceHandlers()` method.

```
48    /**
49     * @param registry spring, serving static resources
50     */
51    @Override
52    public void addResourceHandlers(ResourceHandlerRegistry registry) {
53
54       registry.
55          addResourceHandler(CANONICAL_RESOURCE_CATALOG).
56             addResourceLocations("classpath:/META-INF/resources/");
57
58       registry.
59          addResourceHandler(CANONICAL_WEBJARS_PATH).
60             addResourceLocations("classpath:/META-INF/resources/webjars/");
61    }
```

B. The `Swagger2UiConfiguration.java` class must always have that name.

C. There is one annotation that the `Swagger2UiConfiguration` class must be decorated with, and one that is conditionally optional (see next section).

The annotations are, respectively:

`@Configuration`

`@EnableSwagger2`

```
24   @Configuration
25   @EnableSwagger2
26   public class Swagger2UiConfiguration extends WebMvcConfigurerAdapter {
27
28      private static final String CANONICAL_RESOURCE_CATALOG = "swagger-ui.html";
29      private static final String CANONICAL_WEBJARS_PATH = "/webjars/**";
30
31
```

D. There are two (interchangeable) abstract classes, one of which the `Swagger2UiConfiguration` class must extend:

`WebMvcConfigurationSupport`
(provides sensible defaults, no `@EnableWebMvc` annotation)

`WebMvcConfigurerAdapter`
(enables customization, requires `@EnableWebMvc` annotation*)

ref https://bit.ly/2Xh52vh

E. Finally, there's one specification (for the only mandated resource handler), the value of which is hard-wired into Swagger:

`swagger-ui.html`

Essentially, the value swagger-ui.html# (Line 28) names a canonical Swagger publication-page which serves as a "catalog" of resources. This catalog publishes endpoints/resources, and details about a resource's HTTP verbs).

Design, Discovery

With a deployed Swagger-enabled application (server) running, you can interrogate the status of the endpoints being published by Swagger.

The token:

```
/swagger
```

... in the link below is the application context path. The value of this path is specified in the canonical Spring `application.properties` file:
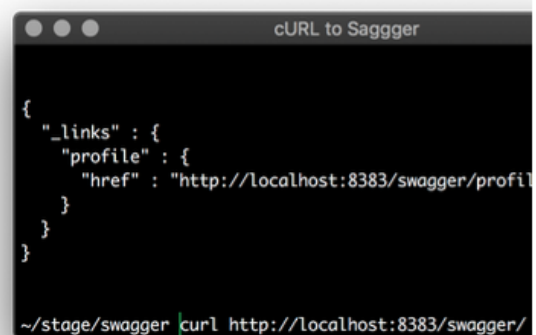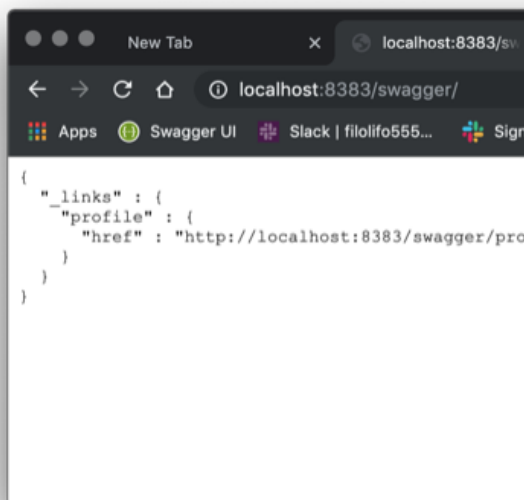
```
server.port=8383
server.contextPath=/swagger

spring.output.ansi.enabled=never
```

Specifying just the context path as an URI:

http://localhost:8383/swagger/

… you get the information shown below (a browser call and a cURL call are shown side-by-side).

The Swagger-aware Controller

Several Swagger annotations can be added to any Spring Controller to enable the publication of endpoints/resources that exist in the Controller.

Besides the usual `@RestController` annotation, there must be an additional Swagger annotation:

`@Api(value="Controller")`

```
16   @Api(value = "Controller")
17   @RestController
18   public class Controller {
19
```

The Swagger-aware Controller, annotating methods

The `String` value within the `@ApiOperation` annotation's `value` attribute (Line 44) appears as a descriptor associated w/ each HTTP verb shown on the Swagger resource-rendering page (named: `swagger-ui.html#`).

Reference

The `swagger-ui.html#` page appears below the Spring `Controller` screen shot:

```
43
44      @ApiOperation(value = "Value Key:  Get One Resource ",
45                      response = Resource.class, tags = "getResource")
46      @RequestMapping(value = "/getResource/{name}")
47      public Resource getResource(@PathVariable(value = "name") String name) {
```

getResource   ⌄

GET	/getResource/{name} Value Key: Get One Resource
HEAD	/getResource/{name} Value Key: Get One Resource
POST	/getResource/{name} Value Key: Get One Resource
PUT	/getResource/{name} Value Key: Get One Resource
DELETE	/getResource/{name} Value Key: Get One Resource
OPTIONS	/getResource/{name} Value Key: Get One Resource
PATCH	/getResource/{name} Value Key: Get One Resource

Another attribute of the `@ApiOperation` annotation is named `response` (Line 45); it maps an internal "model" for Swagger to represent.

The last attribute `tags` (Line 45) maps the Java implementation method with the logical path for each of the seven (7) HTTP verbs that Swagger recognizes.

About the Resource (the embedded model)

The `@ApiProperty` Swagger annotation has a `value` attribute (Line 7, below) which provides a default value for the Java property/attribute with which it associated/attached (it's of limited utility, IMHO).

By comparing the two annotations in the Java class shown in the next screen shot (Lines 7, 11) with the Swagger output, you can see that – when the `value` mapping/attribute is not specified in the Java class' annotation – Swagger only reports the field's data type, and lacks an assigned value.

Two pairs of screen shots contrast/compare the results of using
the value attribute of the @ApiModelProperty annotation

Resource.java

```java
package org.ascension.swagger.model;

import io.swagger.annotations.ApiModelProperty;

public class Resource {

    @ApiModelProperty(name="name", required=true) //, value= "austin")
    private String name;
    public String getName() { return name; }

    @ApiModelProperty(name="country", required=true, value= "texas")
    private String country;
    public String getCountry() { return country; }

    public Resource(String nameVal, String countryVal) {

        super();

        name = nameVal;
        country = countryVal;
    }

    @Override
    public String toString() {
        return "Resource [name=" + name + ", country=" + country + "]";
    }
}
```

```
Resource ⌄ {
    country*           string
                       texas
    name*              string
}
```

```java
package org.ascension.swagger.model;

import io.swagger.annotations.ApiModelProperty;

public class Resource {

    @ApiModelProperty(name="name", required=true, value= "austin")
    private String name;
    public String getName() { return name; }

    @ApiModelProperty(name="country", required=true, value= "texas")
    private String country;
    public String getCountry() { return country; }

    public Resource(String nameVal, String countryVal) {

        super();

        name = nameVal;
        country = countryVal;
    }

    @Override
    public String toString() {
        return "Resource [name=" + name + ", country=" + country + "]";
    }
}
```

src/main/java/org/ascension/swagger/model/Resource   CRLF   UTF-8   Java   GitHub   Git (0)

```
Resource ⌄ {
    country*            string
                        texas
    name*               string
                        austin
}
```

This completes this section.
Examples (using canonical Swagger links) follow

Swagger, Exposition

The URL below reveals the wealth of information about the endpoints, which Swagger publishes.
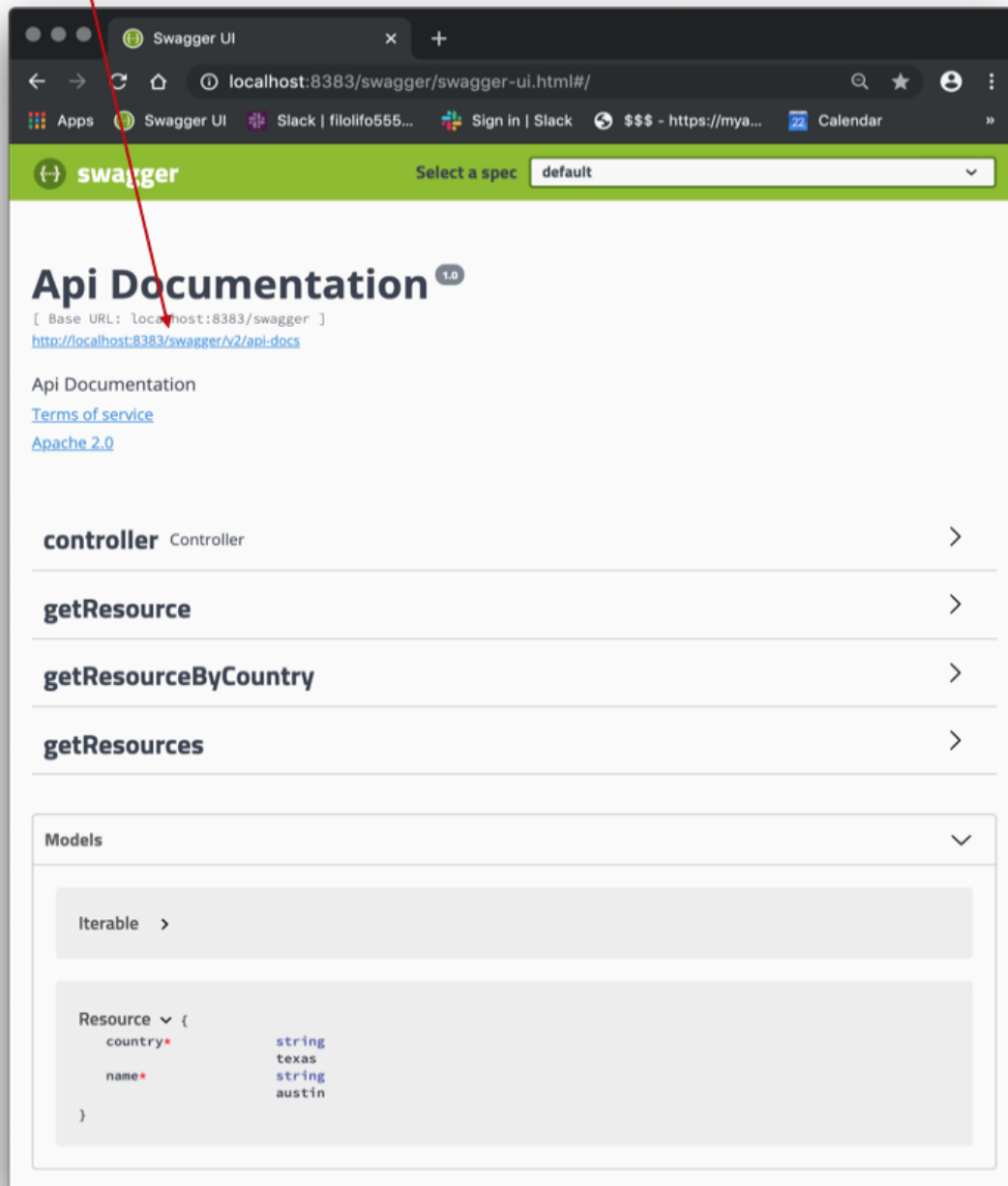
http://localhost:8383/swagger/swagger-ui.html#
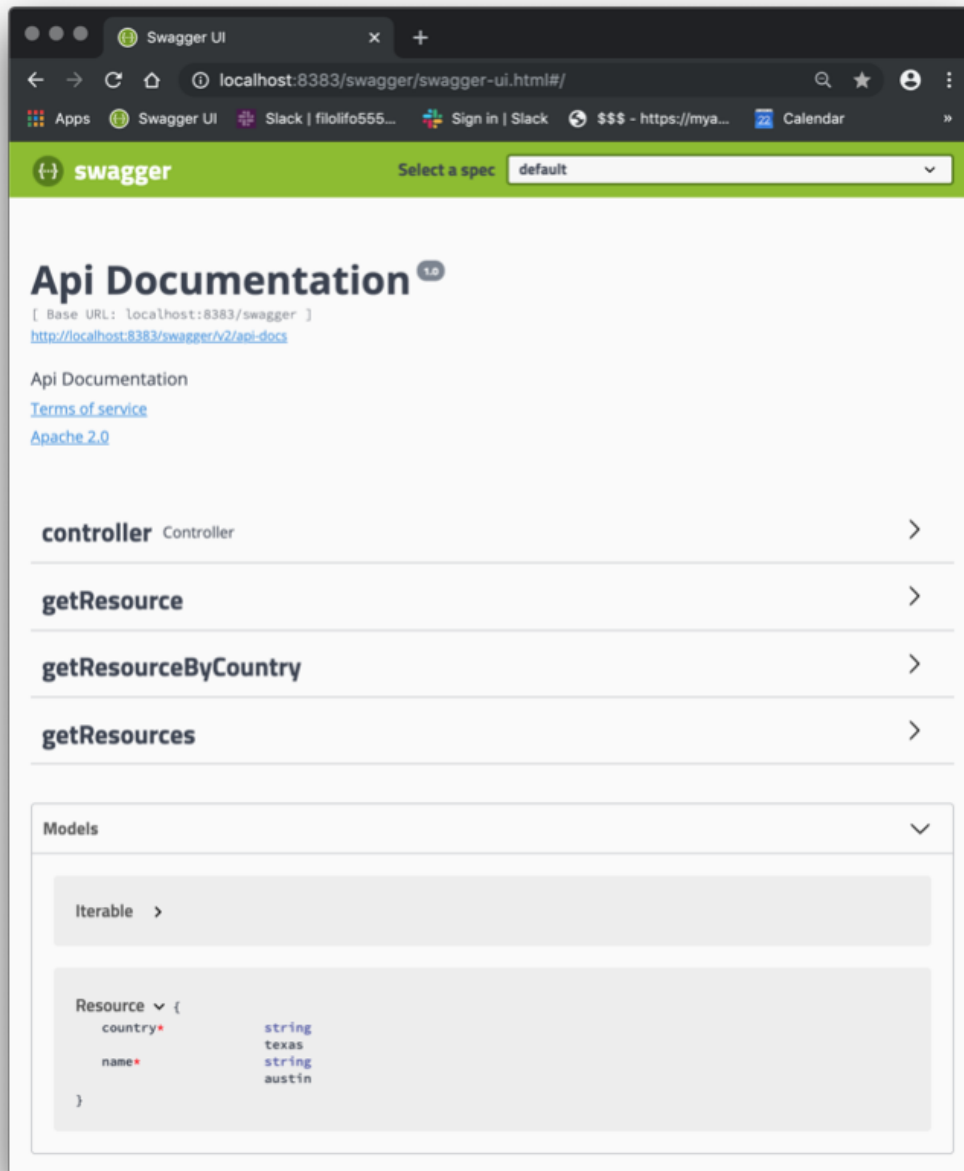
Each category that is listed above, is depicted below
(with a reference to the Java annotation to which it corresponds).

This link exposes a very comprehensive view of what Swagger publishes – the JSON for each HTTP verb.

Since the meta-information when clicking the link is presented in a mimified JSON format, it is best viewed in something like JSON Lint. See the 2 screen shots after the one directly below).

{"swagger":"2.0","info":{"description":"Api Documentation","version":"1.0","title":"Api
Documentation","termsOfService":"urn:tos","contact":{},"license":{"name":"Apache
2.0","url":"http://www.apache.org/licenses/LICENSE-2.0"}},"host":"localhost:8383","basePath":"/swagger","tags":
[{"name":"controller","description":"Controller"}],"paths":{"/getResource/{name}":{"get":{"tags":
["getResource"],"summary":"Value Key: Get One Resource ","operationId":"getResourceUsingGET","produces":
["*/*"],"parameters":
[{"name":"name","in":"path","description":"name","required":true,"type":"string"}],"responses":{"200":
{"description":"OK","schema":{"$ref":"#/definitions/Resource"}},"401":{"description":"Unauthorized"},"403":
{"description":"Forbidden"},"404":{"description":"Not Found"}},"deprecated":false},"head":{"tags":
["getResource"],"summary":"Value Key: Get One Resource ","operationId":"getResourceUsingHEAD","consumes":
["application/json"],"produces":["*/*"],"parameters":
[{"name":"name","in":"path","description":"name","required":true,"type":"string"}],"responses":{"200":
{"description":"OK","schema":{"$ref":"#/definitions/Resource"}},"204":{"description":"No Content"},"401":
{"description":"Unauthorized"},"403":{"description":"Forbidden"}},"deprecated":false},"post":{"tags":
["getResource"],"summary":"Value Key: Get One Resource ","operationId":"getResourceUsingPOST","consumes":
["application/json"],"produces":["*/*"],"parameters":
[{"name":"name","in":"path","description":"name","required":true,"type":"string"}],"responses":{"200":
{"description":"OK","schema":{"$ref":"#/definitions/Resource"}},"201":{"description":"Created"},"401":

JSON **JSONLint** - The JSON Validator   **Try the New Pro**   ☰ More Developer Tools

```
52              },
53 ▾            "head": {
54                  "tags": ["getResource"],
55                  "summary": "Value Key:  Get One Resource ",
56                  "operationId": "getResourceUsingHEAD",
57                  "consumes": ["application/json"],
58                  "produces": ["*/*"],
59 ▾                "parameters": [{
60                      "name": "name",
61                      "in": "path",
62                      "description": "name",
63                      "required": true,
64                      "type": "string"
65                  }],
66 ▾                "responses": {
67 ▾                    "200": {
68                          "description": "OK",
69 ▾                        "schema": {
70                              "$ref": "#/definitions/Resource"
71                          }
72                      },
73 ▾                    "204": {
74                          "description": "No Content"
75                      },
76 ▾                    "401": {
```

Validate JSON   Clear   **Support JSONLint for $2/Month**

Results

Valid JSON

Returning to the Swagger welcome page, the section labeled **getResource** when expanded, shows the URI for each HTTP verb:
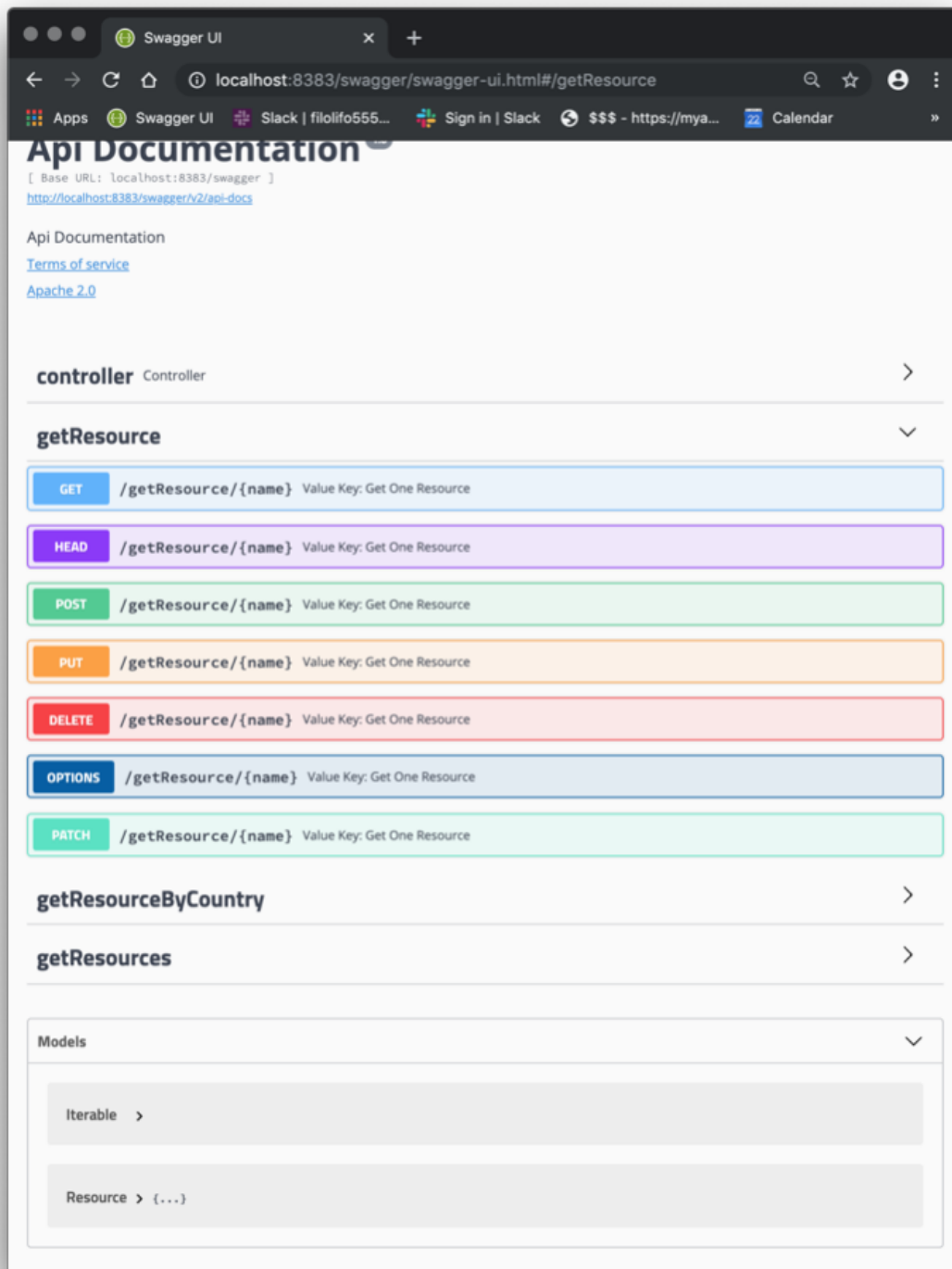
Returning again to the welcome page, the section labeled **getResourceByCountry** when expanded, shows the URI for each HTTP verb:

Returning to the welcome page, the section labeled **getResources** when expanded, shows the URI for each HTTP verb:
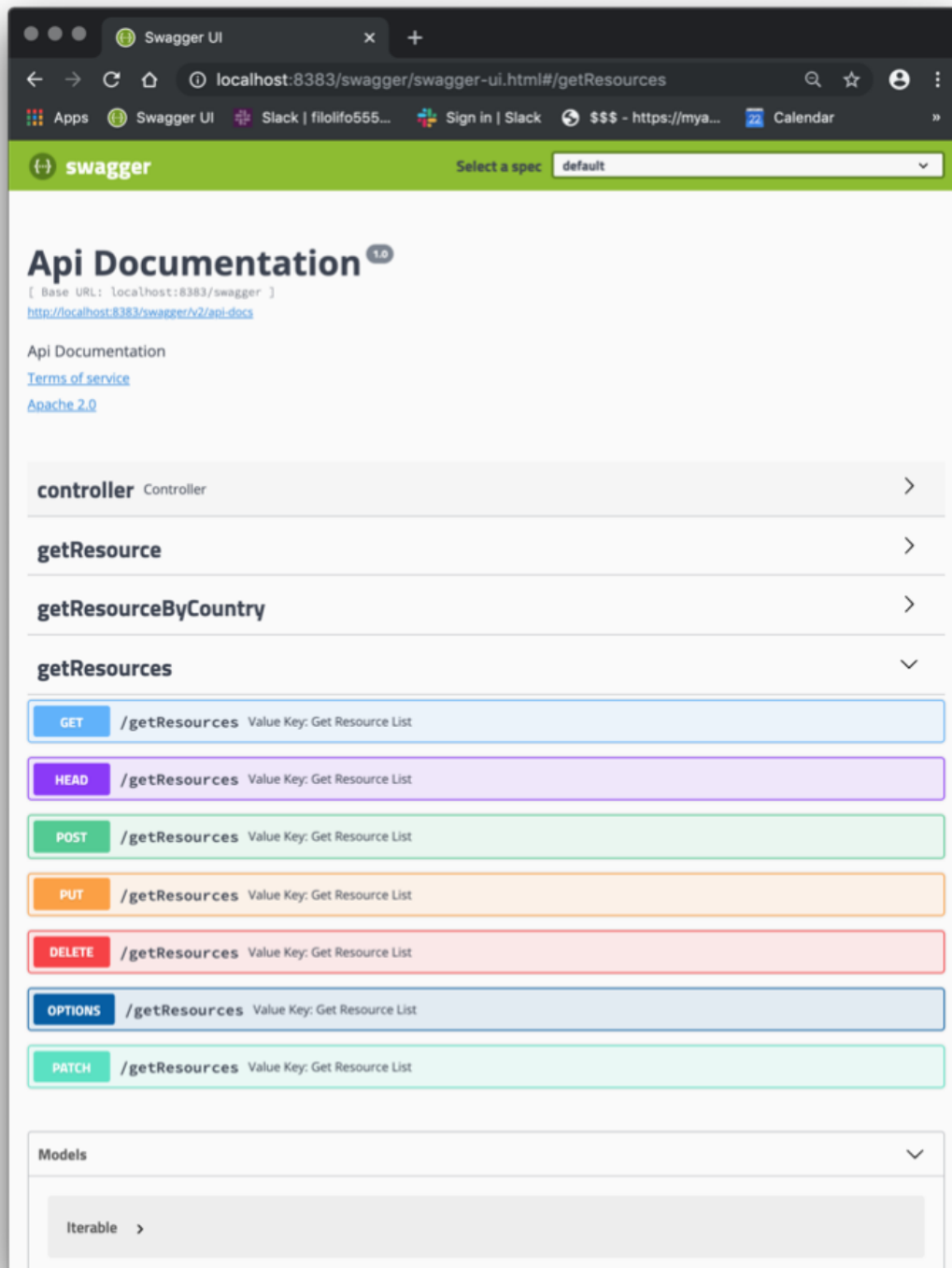
Returning to the welcome page one last time, the section labeled **Models** when expanded, shows a subsection labeled *Iterable* (which is empty in this case).
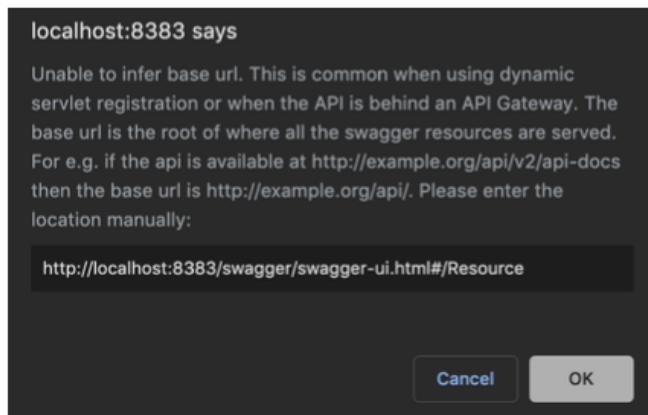
There is another subsection of **Models** labeled *Resource*:

```
Resource ∨ {
    country*          string
                      texas
    name*             string
                      austin
}
```

Details about this feature of Swagger were covered in the beginning of this section.

Appendix

* When the `@EnableWebMvc` annotation is missing from the `Swagger2UiConfiguration` class – and when `Swagger2UiConfiguration` extends `WebMvcConfigurerAdapter` – then, this runtime error surfaces:

localhost:8383 says

Unable to infer base url. This is common when using dynamic servlet registration or when the API is behind an API Gateway. The base url is the root of where all the swagger resources are served. For e.g. if the api is available at http://example.org/api/v2/api-docs then the base url is http://example.org/api/. Please enter the location manually:

http://localhost:8383/swagger/swagger-ui.html#/Resource

Cancel   OK

On Swagger Configuration

https://stackoverflow.com/questions/17898606/difference-between-webmvcconfigurationsupport-and-webmvcconfigureradapter

https://bit.ly/2O71Nm1

https://www.dineshonjava.com/webmvcconfigureradapter-vs-webmvcconfigurationsupport-in-spring-mvc/

https://bit.ly/34SGcV4