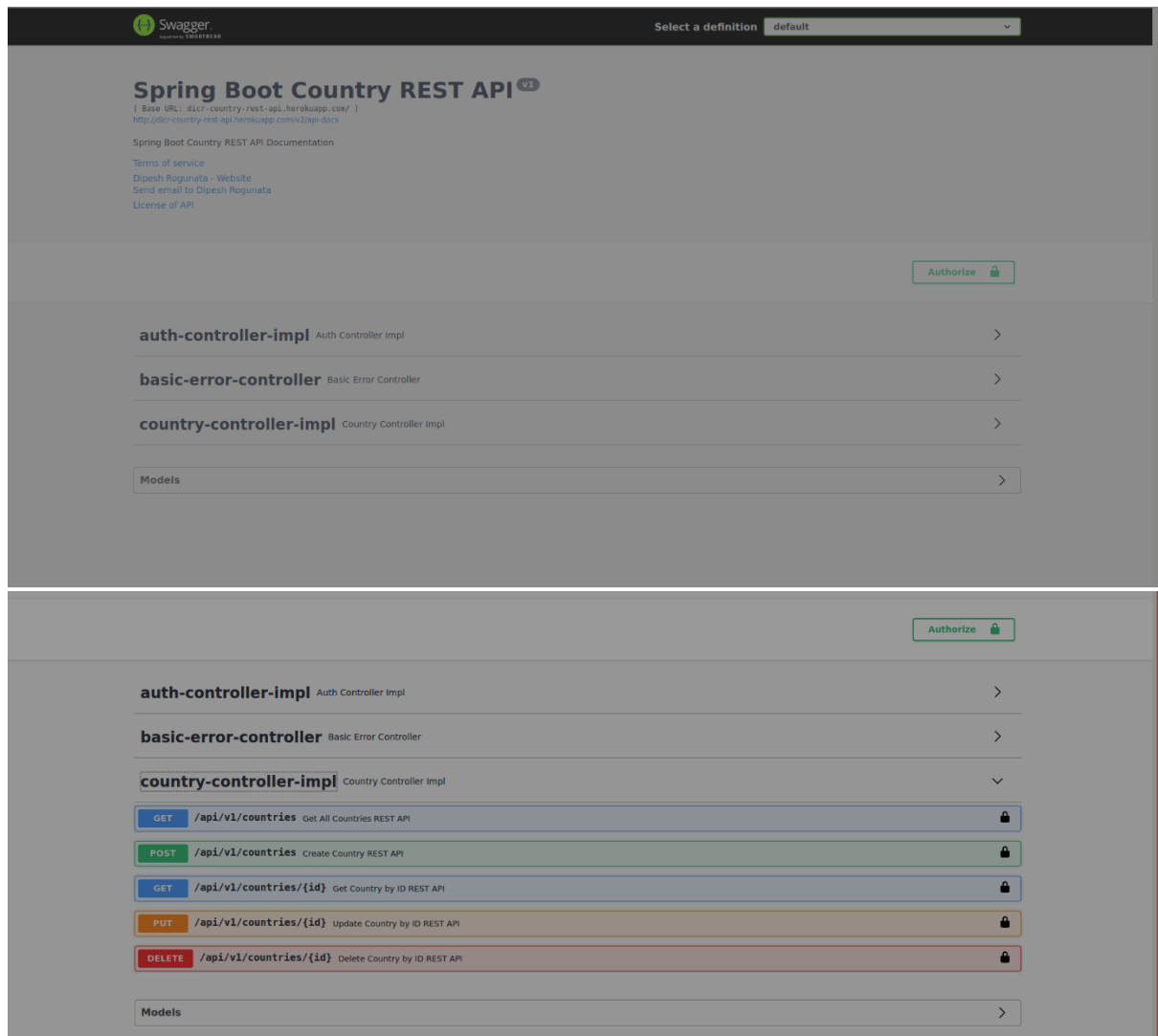


Spring Boot Country REST API SWAGGER

USER GUIDE



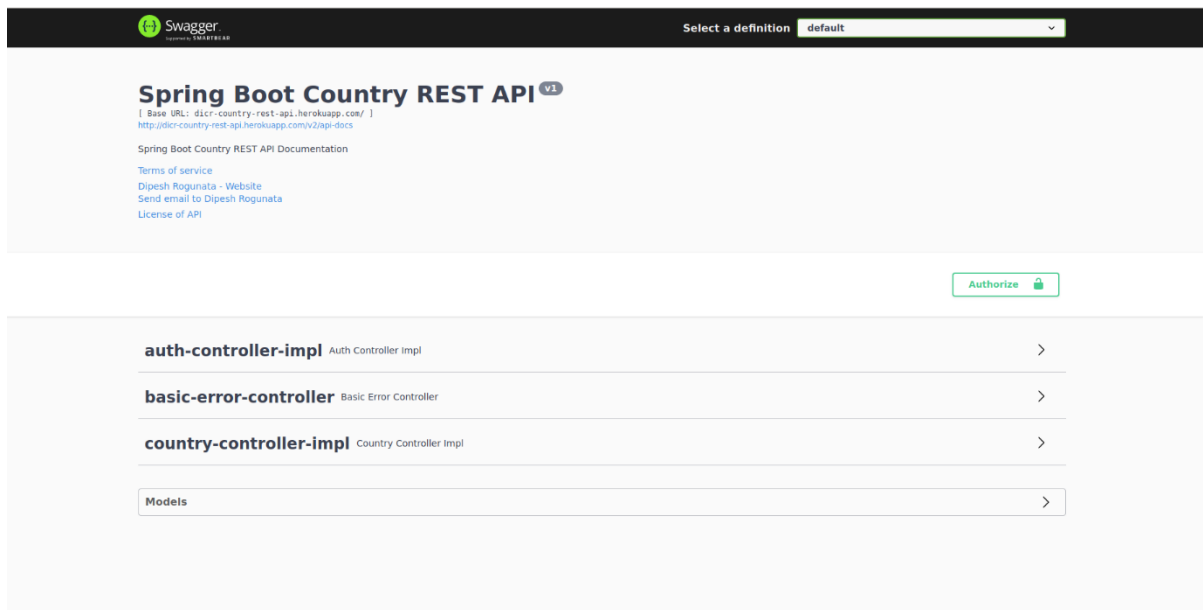
Author: Dipesh Rogunata

January 2022

Contents

1. Home Page.....	3
2. Auth-controller-impl.....	3
3. Country-controller-impl.....	6

1. Home Page



The image shows the Swagger UI home page for the 'Spring Boot Country REST API v1'. The header includes the Swagger logo and a 'Select a definition' dropdown set to 'default'. The main content area displays the API title and version, followed by links for documentation, terms of service, website, email, and license. Below this is an 'Authorize' button. A list of API endpoints is shown, including 'auth-controller-impl', 'basic-error-controller', 'country-controller-impl', and 'Models', each with a right-pointing arrow.

Swagger
OpenAPI 3.0.1

Select a definition **default**

Spring Boot Country REST API ^{v1}

[Base URL: <http://localhost:8080/country-rest-api/>]
<http://localhost:8080/country-rest-api/v2/api-docs>

Spring Boot Country REST API Documentation

[Terms of service](#)
[Dipesh Rogunata - Website](#)
[Send email to Dipesh Rogunata](#)
[License of API](#)

Authorize

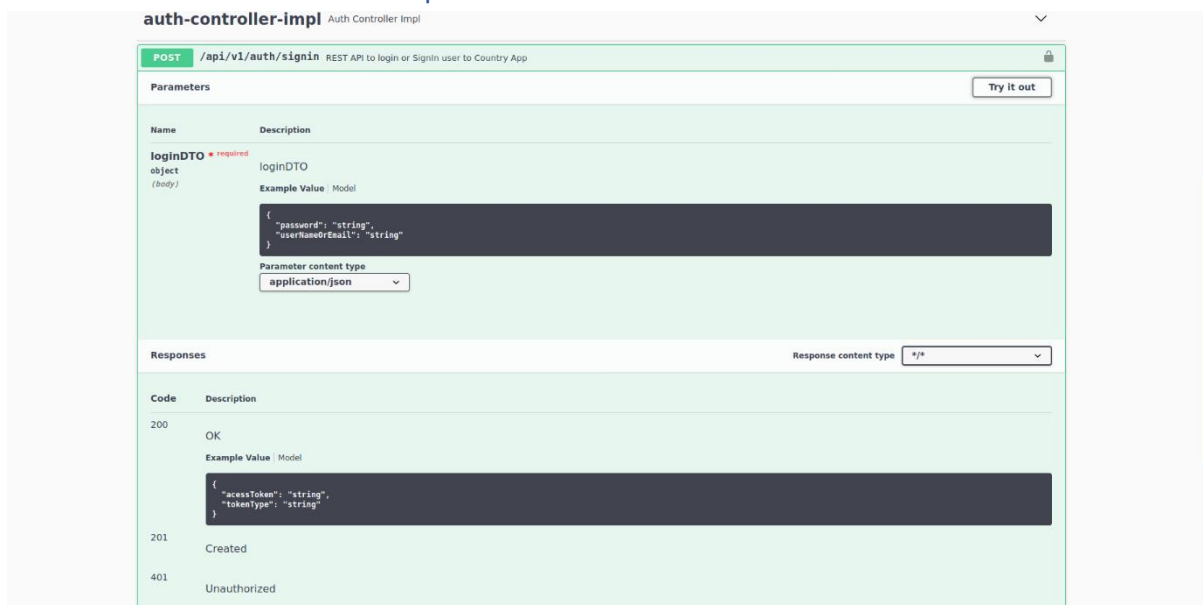
auth-controller-impl Auth Controller Impl >

basic-error-controller Basic Error Controller >

country-controller-impl Country Controller Impl >

Models >

2. Auth-controller-impl



The image shows the Swagger UI details for the 'auth-controller-impl' endpoint. The endpoint is a POST request to '/api/v1/auth/signin'. The description is 'REST API to login or Signin user to Country App'. The parameters section shows a required 'loginDTO' object with an example value: {'password': 'string', 'usernameOrEmail': 'string'}. The responses section shows three possible responses: 200 OK, 201 Created, and 401 Unauthorized. The response content type is set to '*/*'. The example value for the 200 response is: {'accessToken': 'string', 'tokenType': 'string'}.

auth-controller-impl Auth Controller Impl

POST /api/v1/auth/signin REST API to login or Signin user to Country App

Parameters Try it out

Name	Description
loginDTO * required object (body)	loginDTO Example Value Model <pre>{ "password": "string", "usernameOrEmail": "string" }</pre> Parameter content type application/json

Responses Response content type */*

Code	Description
200	OK Example Value Model <pre>{ "accessToken": "string", "tokenType": "string" }</pre>
201	Created
401	Unauthorized

Input the username and password information and click Execute. The data shown below refers to a user that was previously recorded.

auth-controller-impl

Auth Controller Impl

POST

/api/v1/auth/signin

REST API to login or SignIn user to Country App

📋

Parameters

Cancel

Name	Description
loginDTO required	loginDTO
object	
(body)	<div>Edit ValueModel</div> <div><pre>{ "password": "admin123", "userEmail": "dinesh.regunata"}</pre></div> <div>Cancel</div>
Parameter content type	<div>application/json</div>

Execute

Responses

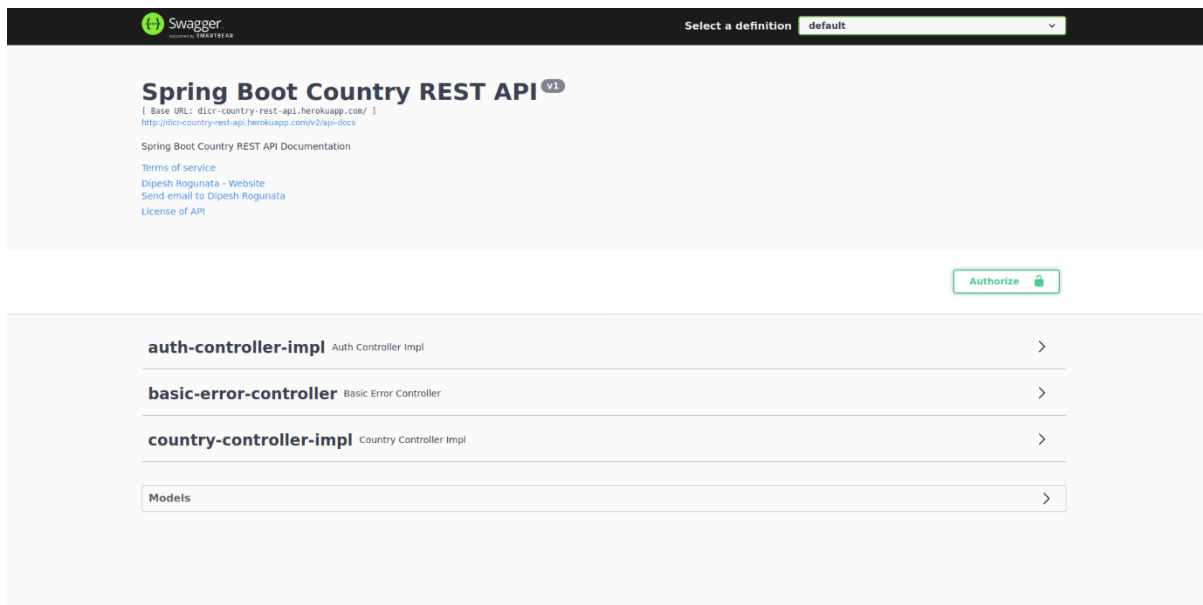
Response content type */*

Code	Description
------	-------------

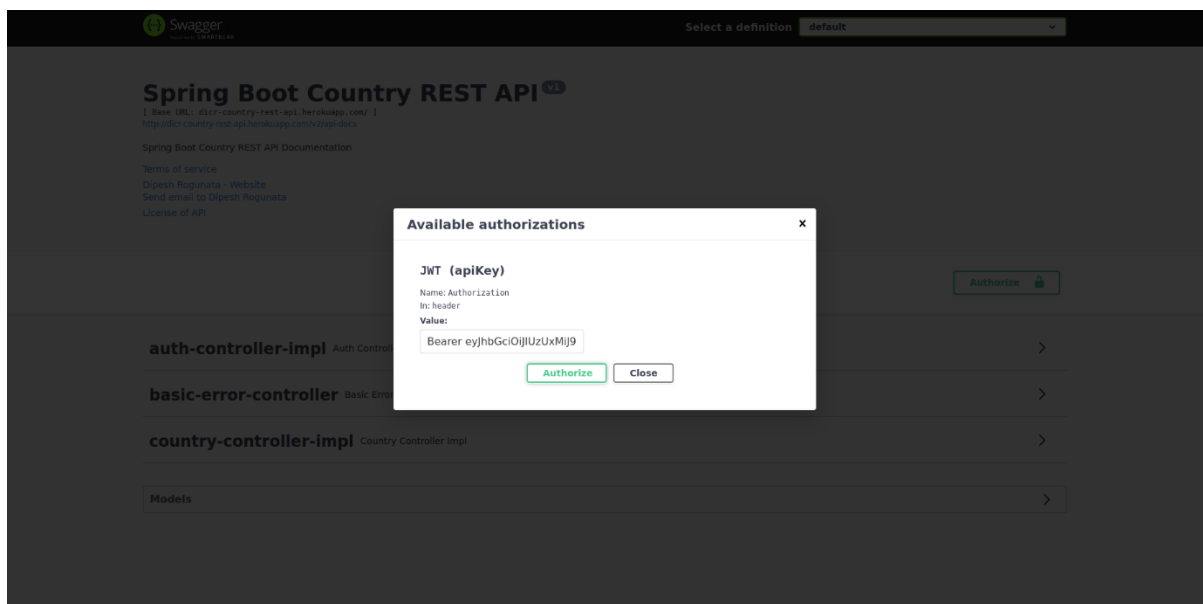
Copy the access token and please note that the token type is Bearer.

[illegible]

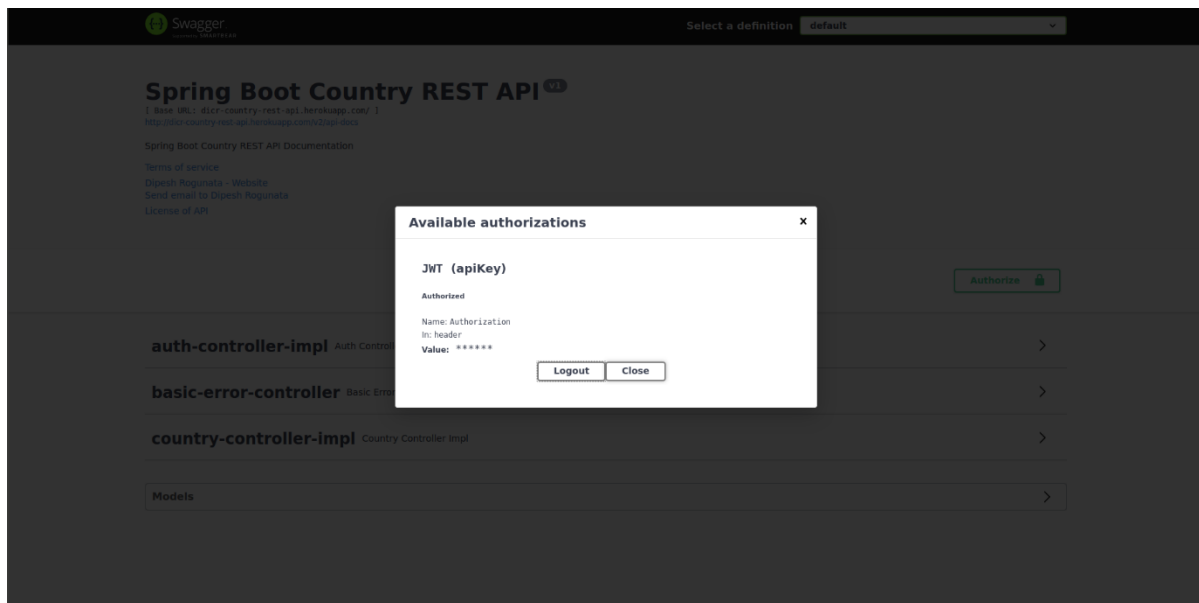
Click on Authorize



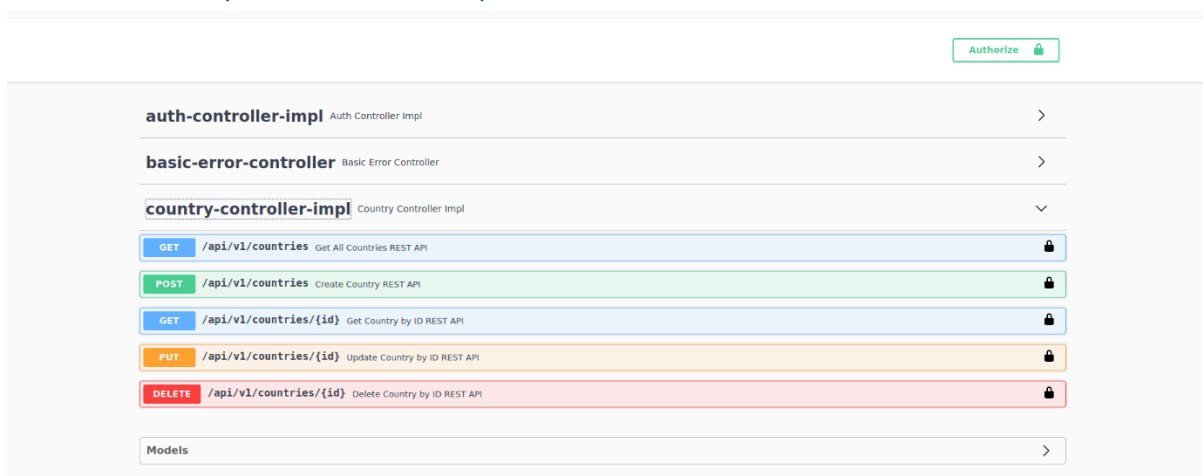
On the value field, type: Bearer. Give a space and past the token which was given on the login step and click Authorize.



The screen below must be shown so it means that the authorize process was successfully occurred. Click the Close button. If something unexpected happens please try to follow the previous stpes.



3. Country-controller-impl



Click on Get All Countries REST API and after click on Try it out.

country-controller-impl Country Controller Impl

GET /api/v1/countries Get All Countries REST API

Try it out

Name	Description
pageNo integer(int32) (query)	pageNo Default value : 0
pageSize integer(int32) (query)	pageSize Default value : 10
sortBy string (query)	Available sorting options: id, name, capital, region, subRegion, area Default value : id
sortDir string (query)	Available sorting options: asc, desc Default value : asc

Responses

Response content type: */*

Insert the sort parameters according to your requirements and click Execute

country-controller-impl Country Controller Impl

GET /api/v1/countries Get All Countries REST API

Cancel

Name	Description
pageNo integer(int32) (query)	pageNo
pageSize integer(int32) (query)	pageSize
sortBy string (query)	Available sorting options: id, name, capital, region, subRegion, area
sortDir string (query)	Available sorting options: asc, desc

Execute

The result of the operation will be shown like the image below.

Request URL
http://dicr-country-rest-api.herokuapp.com/api/v1/countries?pageNo=0&pageSize=10&sortBy=name&sortDir=asc

Server response

Code Details

200

Response body

```
{
  "content": [
    {
      "id": 4,
      "name": "Mozambique",
      "capital": "Maputo",
      "region": "Africa",
      "subRegion": "Southern Africa",
      "area": 801537
    },
    {
      "id": 14,
      "name": "Portugal",
      "capital": "Lisboa",
      "region": "Europe",
      "subRegion": "Southern Europe",
      "area": 92090
    }
  ],
  "pageNo": 0,
  "pageSize": 10,
  "totalElements": 2,
  "totalPages": 1,
  "last": true
}
```

Response headers

```
cache-control: no-cache=storemax-age=0must-revalidate
connection: keep-alive
content-type: application/json
date: Wed26 Jan 2022 11:30:57 GMT
expires: 0
pragma: no-cache
server: Cowboy
transfer-encoding: chunked
via: 1.1 vegur
x-content-type-options: nosniff
x-frame-options: DENY
x-xss-protection: 1; mode=block
```

Responses

Code	Description
200	OK

Click on Create Country REST API.

Authorize

auth-controller-impl Auth Controller Impl

basic-error-controller Basic Error Controller

country-controller-impl Country Controller Impl

GET /api/v1/countries Get All Countries REST API

POST /api/v1/countries Create Country REST API

GET /api/v1/countries/{id} Get Country by ID REST API

PUT /api/v1/countries/{id} Update Country by ID REST API

DELETE /api/v1/countries/{id} Delete Country by ID REST API

Models

Click on Try it out

POST /api/v1/countries Create Country REST API

Parameters Try it out

Name	Description
countryDTO required	countryDTO
object (body)	
Example Value	Model
<pre>{ "area": 0, "capital": "string", "id": 0, "name": "string", "region": "string", "subRegion": "string"}</pre>	
Parameter content type	application/json

Responses Response content type */*

Code	Description
200	OK
Example Value	Model
<pre>{ "area": 0, "capital": "string", "id": 0, "name": "string", "region": "string", "subRegion": "string"}</pre>	
201	Created

Remove the id field, because when creating a Country the ID is automatically generated, so there is no need to inform it.

POST /api/v1/countries Create Country REST API

Parameters Cancel

Name	Description
countryDTO required	countryDTO
object (body)	
Example Value	Model
<pre>{ "area": 0, "capital": "string", "id": 0, "name": "string", "region": "string", "subRegion": "string"}</pre>	
Parameter content type	application/json

Cancel

Execute

Responses Response content type */*

Code	Description
200	OK

It should look like the image below after removing the id field

The screenshot shows a REST client interface for a POST request to `/api/v1/countries`. The request body is a `countryDTO` object with the following JSON structure:

```
{  "area": 0,  "capital": "string",  "name": "string",  "region": "string",  "subRegion": "string"}
```

The interface includes a `Cancel` button, a `Parameter content type` dropdown set to `application/json`, and an `Execute` button. The `Responses` section shows a `200 OK` status.

Insert the data of the country to be created and click Execute

The screenshot shows the same REST client interface, but the `countryDTO` object now contains specific data for Brazil:

```
{  "area": 8515767.0,  "capital": "Brasilia",  "name": "Brazil",  "region": "Americas",  "subRegion": "South America"}
```

The `Execute` button is highlighted, indicating it has been clicked. The `Responses` section still shows a `200 OK` status.

It should return a response with a code 201 and a response body must have the country information including the id that was automatically generated.

The screenshot displays a REST client interface with the following details:

- Request URL:** `http://dicr-country-rest-api.herokuapp.com/api/v1/countries`
- Server response:**
 - Code:** 201
 - Details:**
 - Response body:**

```
{
  "id": 24,
  "name": "Brazil",
  "capital": "Brasilia",
  "region": "Americas",
  "subRegion": "South America",
  "area": 8515767
}
```
 - Response headers:**

```
cache-control: no-cache=storemax-age=86400;must-revalidate
connection: keep-alive
content-type: application/json
date: Wed, 01 Jan 2020 11:37:00 GMT
expires: 0
pragma: no-cache
server: Cowboy
transfer-encoding: chunked
via: 1.1 vevor
x-content-type-options: nosniff
x-frame-options: DENY
x-ssr-protection: 1; mode=block
```
- Responses:**
 - Code:** 200, **Description:** OK
 - Example Value | Model:**

```
{
  "area": 0,
  "capital": "string",
  "id": 0,
  "name": "string",
  "region": "string",
  "subRegion": "string"
}
```
 - Code:** 201, **Description:** Created

Click on Get Country by ID REST API

The screenshot shows a REST client interface with an **Authorize** button in the top right corner. The main area lists the following controllers and their endpoints:

- auth-controller-impl** Auth Controller Impl >
- basic-error-controller** Basic Error Controller >
- country-controller-impl** Country Controller Impl ▾
 - GET** `/api/v1/countries` Get All Countries REST API
 - POST** `/api/v1/countries` Create Country REST API
 - GET** `/api/v1/countries/{id}` Get Country by ID REST API
 - PUT** `/api/v1/countries/{id}` Update Country by ID REST API
 - DELETE** `/api/v1/countries/{id}` Delete Country by ID REST API
- Models** >

Click on Try it out

The screenshot shows a REST client interface for the endpoint `GET /api/v1/countries/{id}` with the description "Get Country by ID REST API". In the "Parameters" section, the `id` parameter is defined as a required integer (int64) in the path, with a placeholder value `id - id`. A "Try it out" button is located in the top right corner of the parameters section. The "Responses" section shows a table of status codes and descriptions: 200 OK, 401 Unauthorized, 403 Forbidden, and 404 Not Found. An example JSON response for the 200 status is displayed, showing fields like `area`, `capital`, `id`, `name`, `region`, and `subRegion`.

Insert the id of the country that is required to see the information and click Execute.

This screenshot shows the same REST client interface, but now the `id` parameter in the path is filled with the value `24`. A blue "Execute" button is prominently displayed below the parameters section. The "Responses" section remains the same, showing the status codes and the example JSON response for the 200 status.

It should return a response with a code 200 and a response body must have the country information of the provided id.

The screenshot shows a REST client interface with the following details:

- Request URL:** `http://dic-country-rest-api.herokuapp.com/api/v1/countries/24`
- Server response:**
 - Code:** 200
 - Response body:**

```
{
  "id": 24,
  "name": "Brazil",
  "capital": "Brasilia",
  "region": "Americas",
  "subRegion": "South America",
  "area": 8515767
}
```
 - Response headers:**

```
cache-control: no-cache;no-store;max-age=0;must-revalidate
connection: keep-alive
content-type: application/json
date: Wed, 25 Jun 2022 11:41:23 GMT
expires: 0
pragma: no-cache
server: Cowboy
transfer-encoding: chunked
via: 1.1 vaper
x-content-type-options: nosniff
x-frame-options: DENY
x-xss-protection: 1; mode=block
```
- Responses:**
 - Code:** 200
 - Description:** OK
 - Example Value | Model:**

```
{
  "area": 0,
  "capital": "string",
  "id": 0,
  "name": "string",
  "region": "string",
  "subRegion": "string"
}
```
- 401:** Unauthorized

Click on Update Country by ID REST API

The screenshot shows a REST client interface with the following details:

- Authorize** button (green)
- auth-controller-impl** Auth Controller impl >
- basic-error-controller** Basic Error Controller >
- country-controller-impl** Country Controller Impl v
- GET** `/api/v1/countries` Get All Countries REST API
- POST** `/api/v1/countries` Create Country REST API
- GET** `/api/v1/countries/{id}` Get Country by ID REST API
- PUT** `/api/v1/countries/{id}` Update Country by ID REST API

Click on Try it out.

PUT /api/v1/countries/{id} Update Country by ID REST API

Parameters

Try it out

Name	Description
countryDTO * required	countryDTO
object (body)	Example Value Model
	<pre>{ "area": 0, "capital": "string", "id": 0, "name": "string", "region": "string", "subregion": "string"}</pre>
	Parameter content type: application/json
id * required	id
integer(\$int64) (path)	id - id

Responses

Response content type: */*

Code	Description
200	OK
	Example Value Model
	<pre>{ "area": 0, "capital": "string", "id": 0, "name": "string", "region": "string", "subregion": "string"}</pre>

Remove the id field.

PUT /api/v1/countries/{id} Update Country by ID REST API

Parameters

Cancel

Name	Description
countryDTO * required	countryDTO
object (body)	Edit Value Model
	<pre>{ "area": 0, "capital": "string", "id": 0, "name": "string", "region": "string", "subregion": "string"}</pre>
	Parameter content type: application/json
id * required	id
integer(\$int64) (path)	id - id

Execute

Responses

Response content type: */*

Insert the update information of the country that is required to update and inform the id on the required field and click Execute.

The screenshot shows a REST client interface for a PUT request to the endpoint `/api/v1/countries/{id}` with the description "Update Country by ID REST API".

Parameters:

Name	Description
countryDTO * required object (body)	countryDTO Edit Value Model

```
{  "area": 8515767.049,  "capital": "Brasilia",  "name": "Brazil",  "region": "Americas",  "subRegion": "South America"}
```

id * required
integer(\$int64)
(path)

id:

Parameter content type:

Execute

Responses: Response content type:

It should return a response with a code 200 and a response body must have the country with the updated information.

The screenshot shows the response of the PUT request. The Request URL is `http://dicr-country-rest-api.herokuapp.com/api/v1/countries/24`.

Server response:

Code	Details
200	<p>Response body</p> <pre>{ "id": 24, "name": "Brazil", "capital": "Brasilia", "region": "Americas", "subRegion": "South America", "area": 8515767.049}</pre> <p>Response headers</p> <pre>cache-control: no-cache, no-store, max-age=0, must-revalidate connection: keep-alive content-type: application/json date: Wed, 12 Jan 2022 11:46:34 GMT expires: 0 pragma: no-cache server: Cowboy transfer-encoding: chunked via: 1.1 weav x-content-type-options: nosniff x-frame-options: DENY x-xss-protection: 1; mode=block</pre>

Responses:

Code	Description
200	OK

Example Value | Model

```
{  "area": 0,  "capital": "string",  "id": 0,  "name": "string",  "region": "string",  "subRegion": "string"}
```

201 Created

Click on Delete Country REST API

Authorize

auth-controller-impl Auth Controller Impl >

basic-error-controller Basic Error Controller >

country-controller-impl Country Controller Impl v

GET /api/v1/countries Get All Countries REST API

POST /api/v1/countries Create Country REST API

GET /api/v1/countries/{id} Get Country by ID REST API

PUT /api/v1/countries/{id} Update Country by ID REST API

DELETE /api/v1/countries/{id} Delete Country by ID REST API

Click on Try it out

DELETE /api/v1/countries/{id} Delete Country by ID REST API

Try it out

Name	Description
id	integer(int64) id (path)

id - id

Responses

Response content type */*

Code	Description
200	OK
	Example Value Model
	string
204	No Content
401	Unauthorized
403	Forbidden

Models >

Insert the id the country that is required to delete and click Execute.

DELETE

/api/v1/countries/{id} Delete Country by ID REST API

Parameters

Cancel

Name	Description
id required integer(int64)id (path) <input type="text" value="24"/>	

Execute

Responses

Response content type */*

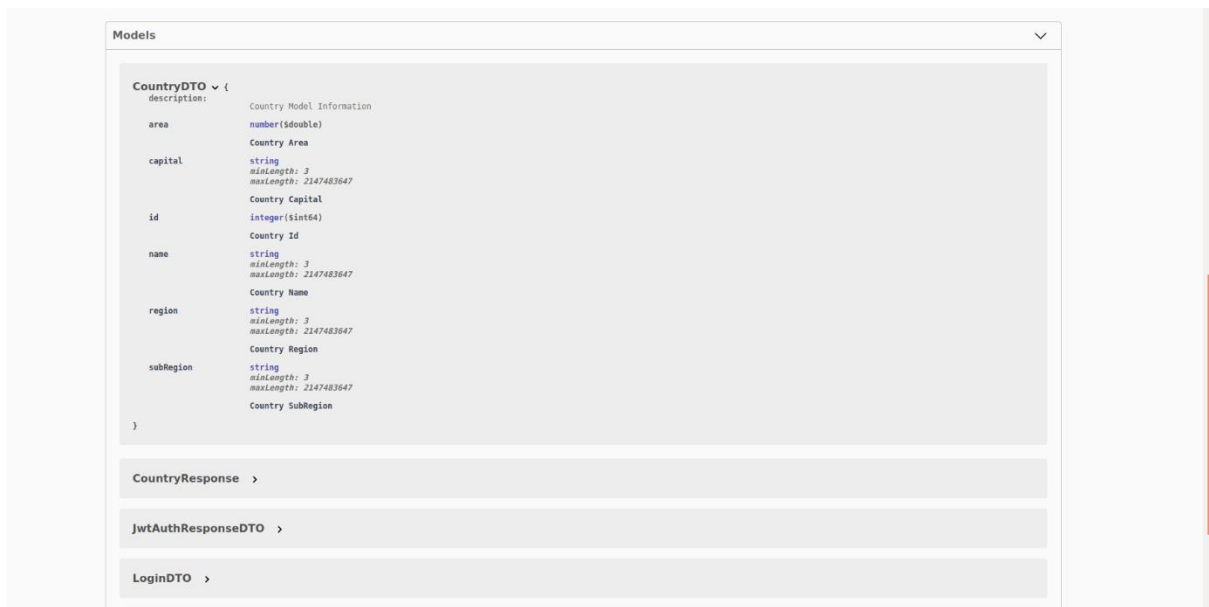
Code	Description
200	OK Example Value Model <div>string</div>
204	No Content
401	Unauthorized
403	Forbidden

Models >

It should return a message informing that the country was successfully deleted.

[illegible]

Click on Models.



The image shows the 'Models' section of a Swagger UI interface. The title 'Models' is at the top left, and a dropdown arrow is at the top right. The main content area displays a list of models. The first model, 'CountryDTO', is expanded, showing its description and fields. Below it are three other models: 'CountryResponse', 'JwtAuthResponseDTO', and 'LoginDTO', each with a right-pointing arrow indicating they can be expanded. The interface has a light gray background with a vertical orange bar on the right side.

```
CountryDTO {  
  description: Country Model Information  
  area: number($double)  
    Country Area  
  capital: string  
    minLength: 3  
    maxLength: 2147483647  
    Country Capital  
  id: integer($int64)  
    Country Id  
  name: string  
    minLength: 3  
    maxLength: 2147483647  
    Country Name  
  region: string  
    minLength: 3  
    maxLength: 2147483647  
    Country Region  
  subRegion: string  
    minLength: 3  
    maxLength: 2147483647  
    Country SubRegion  
}
```

CountryResponse >

JwtAuthResponseDTO >

LoginDTO >