Vidal De La Rosa, Jose Luis

My Tourist Server

Requirements

Contents

[API Requirements 2](#_Toc3297238)

[POST to create a new place. 2](#_Toc3297239)

[GET list of places 2](#_Toc3297240)

[GET an existing place. 2](#_Toc3297241)

[PUT an existing place. 3](#_Toc3297242)

[DELETE an existing place. 3](#_Toc3297243)

# API Requirements

## POST to create a new place.

|  |  |
| --- | --- |
| Create an endpoint to upload new places | |
| Description | As a user  I want to send places to the API server  So that I can manage them somehow |
| Acceptance Criteria | 1. Server should enable a new endpoint under /api/places 2. This endpoint should be a POST 3. It should only accept application/json as Content-Type 4. We should validate the body received:    1. Body should not be null, undefined or empty.    2. Body should be of a previously defined type Place. 5. We should manage errors to return codes properly:    1. 201 CREATED when the place is valid and it has been stored.    2. 400 Bad Request if body is not valid, returning a message with the invalid or empty fields.    3. 500 Internal Server Error for any other reason. |

## GET list of places

|  |  |
| --- | --- |
| Create an endpoint to return a list of places | |
| Description | As a user  I want to retrieve a list of places from the API server  So that I can present them in a client |
| Acceptance Criteria | 1. Server should enable a new endpoint under /api/places 2. This endpoint should be a GET 3. It should only return application/json as Content-Type 4. We should manage errors to return codes properly:    1. 200 OK when there is a list of places to return.    2. 404 Not found if the list of places is empty.    3. 500 Internal Server Error for any other reason. |

## GET an existing place.

|  |  |
| --- | --- |
| Create an endpoint to return a given place by its id | |
| Description | As a user  I want to retrieve a known place by its id from the API server  So that I can present them in a client |
| Acceptance Criteria | 1. Server should enable a new endpoint under /api/places/{id} 2. This endpoint should be a GET 3. It should only return application/json as Content-Type 4. We should manage errors to return codes properly:    1. 200 OK when there is a place to return.    2. 400 Bad Request if id is not present or invalid.    3. 404 Not found if the place has not been found.    4. 500 Internal Server Error for any other reason. |

## PUT an existing place.

|  |  |
| --- | --- |
| Create an endpoint to update a given place by its id | |
| Description | As a user  I want to update a known place by its id  So that I can modify some properties somehow |
| Acceptance Criteria | 1. Server should enable a new endpoint under /api/places/{id} 2. This endpoint should be a PUT 3. It should only return application/json as Content-Type 4. We should validate new body. 5. We should manage errors to return codes properly:    1. 200 OK when the place has been updated properly.    2. 400 Bad Request if the body or id is not valid.    3. 404 Not found if the place to update has not been found.    4. 500 Internal Server Error for any other reason. |

## DELETE an existing place.

|  |  |
| --- | --- |
| Create an endpoint to update a given place by its id | |
| Description | As a user  I want to delete a known place by its id  So that it does no longer managed by the server |
| Acceptance Criteria | 1. Server should enable a new endpoint under /api/places/{id} 2. This endpoint should be a DELETE 3. It should only return application/json as Content-Type 4. We should validate parameters. 5. We should manage errors to return codes properly:    1. 200 OK when the place has been deleted properly.    2. 400 Bad Request if the id is not valid.    3. 404 Not found if the place to update has not been found.    4. 500 Internal Server Error for any other reason. |