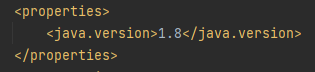
**Simple User Score App**

**Technologies:**

* Java (8,11,16 & 17 supported)
* Spring boot 2.7.0-SNAPSHOT
* Spring Data JPA
* MySQL
* DB migration tool: Liquibase
* Maven

**Steps to Run The APP:**

* System must have Java 8 installed
* Note: Currently using Java 8. But the app can be run using Java 11 (tested), 16 or 17 (tested). In that case need to change the Java version in the POM.xml file



* System must have MySQL server running at port 3306.
* If MySQL should run at a different port, then need to change port number in application.properties file
* Any system meeting the above requirements can immediately run the application without any other configuration

**API Specifications according to Requirement Document:**

1. **createUser**

**Description:** This API will create a new user and insert it into users table in the DB. Please note that, till now this API will keep the userProgress part as null which will be separately created with another api and will be mapped to the already created user. There is an OneToOne relationship between User and UserProgress entities.

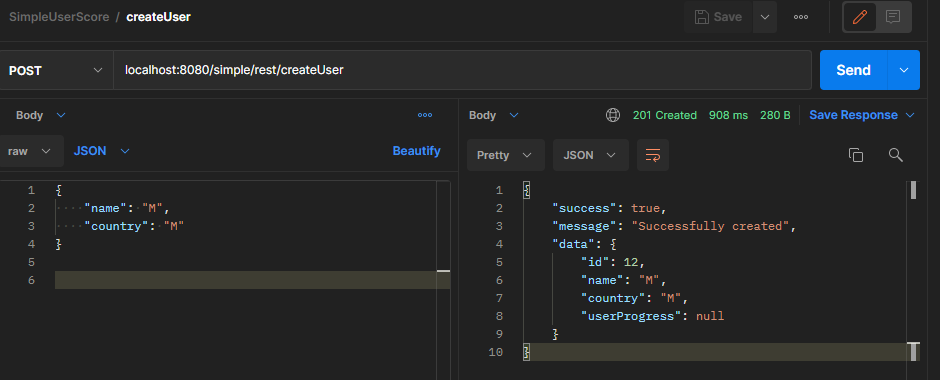
**Endpoint:** localhost:8080/simple/rest/createUser

**Request Method:** POST

**Path Variables:** N/A

**Request Body:** raw JSON request body

**Sample Request and Response:**



**Response Description:** As there is no specific response format provided for this API in the requirement, this response is designed to be wrapped with below fields to achieve meaningful insights:

* **success:** Represents if the operation is successful or not
* **message:** Represents the error message to this request.
* **data:** This field will contain the data resulted from the API request. In this case it is the newly created user in JSON format. Please see userProgress is null here as we haven’t yet created any userProgress for this user to be mapped with.

1. **getUserByID**

**Description:** This API will return an already created user in JSON format. As there is an OneToOne relationship between User and userProgress, userProgress filed will be shown as null if userProgress for this user is not yet created which can be handled by another api (see API createUserProgressbyUserId).

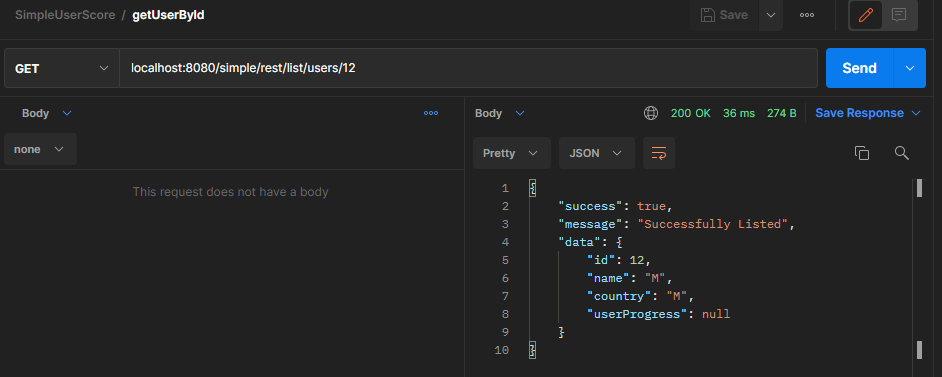
**Endpoint:** localhost:8080/simple/rest/list/users/{id}

**Request Method:** GET

**Path Variable:** id (type Long | represents the id of an already created user)

**Request Body:** None

**Sample Request and Response:**



**Response Description:** As there is no specific response format provided for this API in the requirement, this response is designed to be wrapped with below fields to achieve meaningful insights:

* **success:** Represents if the operation is successful or not
* **message:** Represents the error message to this request.
* **data:** This field will contain the data resulted from the API request. In above case it is the requested user in JSON format. Please see userProgress is null here as we haven’t yet created any userProgress for this user to be mapped with.

1. **updateUserById**

**Description:** This API will update an already created user. It doesn’t expect any userProgress filed for the user in API request and only tends to update the name and country fields of the user. Although if there is an already mapped userProgress associated with the user, this api will not change anything to that and keep that part as it is. userProgress can be updated by another API described below (check updateUserProgressByUserId). It requires a field ‘id’ in the request body representing the id of the user to be update.

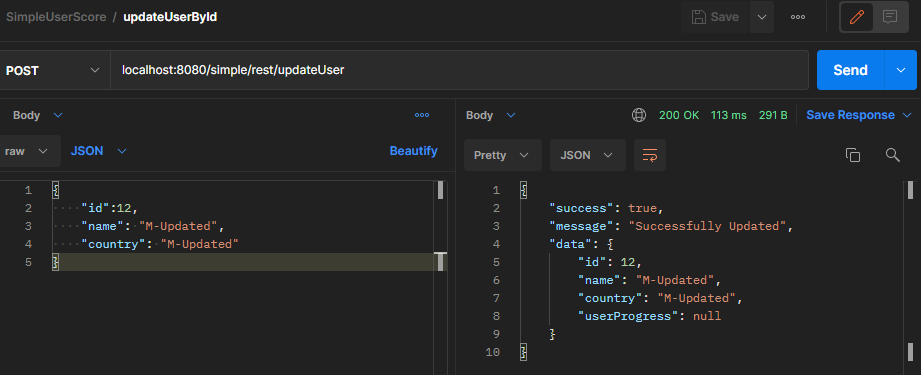
**Endpoint:** localhost:8080/simple/rest/updateUser

**Request Method:** POST

**Path Variable:** N/A

**Request Body:** raw JSON request body

**Sample Request and Response:**



**Response Description:** As there is no specific response format provided for this API in the requirement, this response is designed to be wrapped with below fields to achieve meaningful insights:

* **success:** Represents if the operation is successful or not
* **message:** Represents the error message to this request.
* **data:** This field will contain the data resulted from the API request. In above case it is the updated user in JSON format.

1. **createUserProgressByUserId**

**Description:** This API will create a new userProgress record and insert it into the DB. This api expects a reference (user\_id) to an already created user to which this userProgress will be mapped by the API. If that existing user already has a userProgress mapped, it will just update the existing userProgress with newly provided data and will not create any new userProgress record in DB.

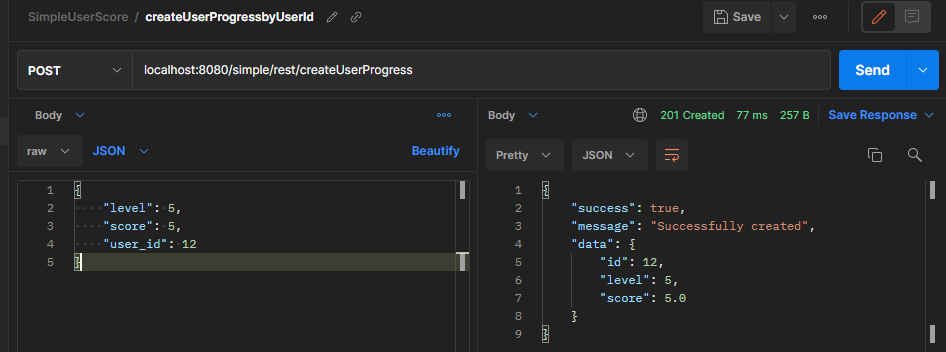
**Endpoint:** localhost:8080/simple/rest/createUserProgress

**Request Method:** POST

**Path Variables:** N/A

**Request Body:** raw JSON request body

**Sample Request and Response:**



**Response Description:** As there is no specific response format provided for this API in the requirement, this response is designed to be wrapped with below fields to achieve meaningful insights:

* **success:** Represents if the operation is successful or not
* **message:** Represents the error message to this request.
* **data:** This field will contain the data resulted from the API request. In this case it is the newly created userProgress record in JSON format.

1. **getUserProgressByUserId**

**Description:** This API will return a userProgress in JSON format. It expects a reference id (as path variable) to an already created user to which the requested userProgress is associated with.

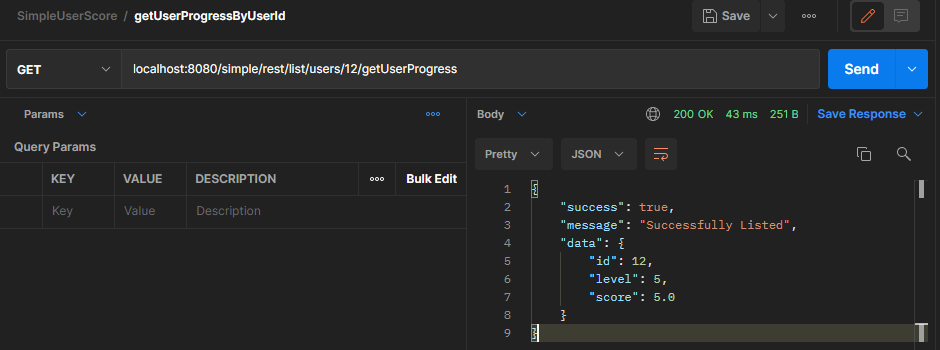
**Endpoint:** localhost:8080/simple/rest/list/users/{id}/getUserProgress

**Request Method:** GET

**Path Variable:** id (type Long | represents the id of an already created user)

**Request Body:** None

**Sample Request and Response:**



**Response Description:** As there is no specific response format provided for this API in the requirement, this response is designed to be wrapped with below fields to achieve meaningful insights:

* **success:** Represents if the operations is successful or not
* **message:** Represents the error message for this request.
* **data:** This field will contain the data resulted from the API request. In above case it is the requested userProgress in JSON format.

1. **updateUserProgressByUserId**

**Description:** This API will update an already created userProgress. Need to provide the user\_id to which this userProgress is associated with.

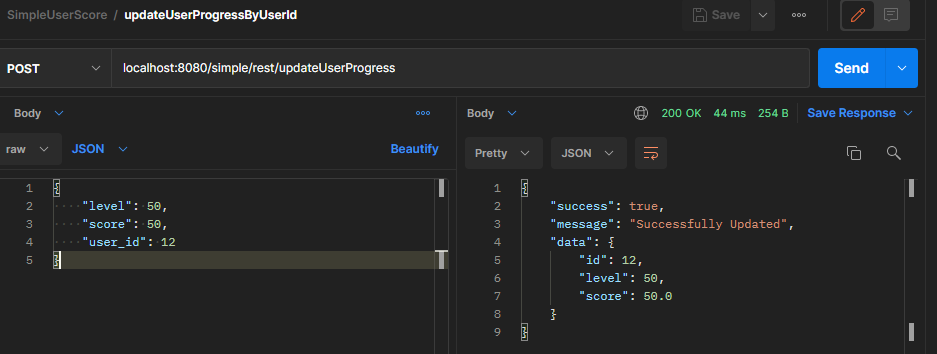
**Endpoint:** localhost:8080/simple/rest/updateUserProgress

**Request Method:** POST

**Path Variable:** N/A

**Request Body:** raw JSON request body

**Sample Request and Response:**



**Response Description:** As there is no specific response format provided for this API in the requirement, this response is designed to be wrapped with below fields to achieve meaningful insights:

* **success:** Represents if the operations is successful or not
* **message:** Represents the error message for this request.
* **data:** This field will contain the data resulted from the API request. In above case it is the updated userProgress in JSON format.

1. **getLeaders**

**Description:** This API will return a list of top user progresses in JSON format ranked by their score value up to rank 10.

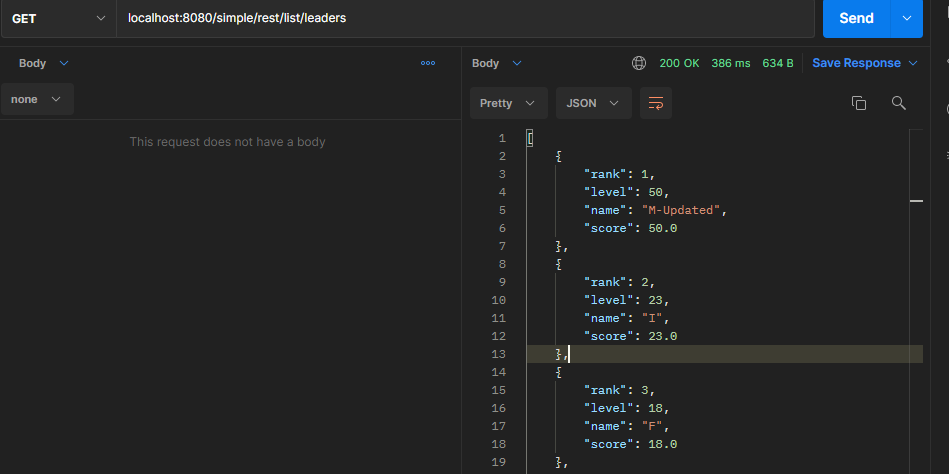
**Endpoint:** localhost:8080/simple/rest/list/leaders

**Request Method:** GET

**Path Variable:** N/A

**Request Body:** None

**Sample Request and Response:**



**Response Description:** As per the requirement document, this will be a list of objects in JSON format having below fields:

* rank
* level
* name
* score

1. **getUserScoreByUserId**

**Description:** This API will return a user’s score along with the rank that the user achieved by the current score compared to the other existing users.

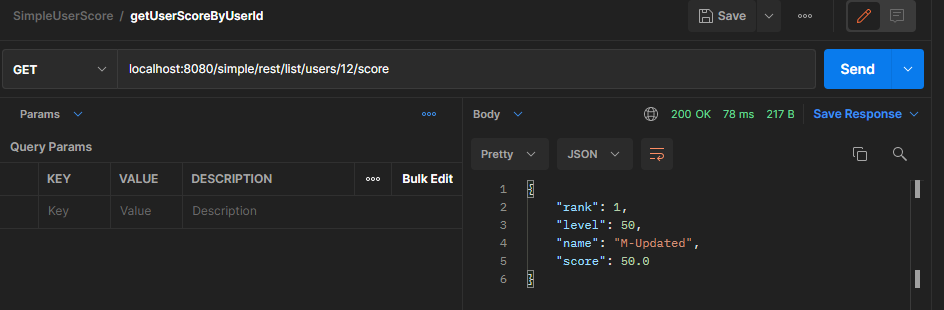
**Endpoint:** localhost:8080/simple/rest/list/users/{id}/score

**Request Method:** GET

**Path Variable:** id (type Long | represents the id of the user whose score is requested)

**Request Body:** None

**Sample Request and Response:**



**Response Description:** As per the requirement document, this will be an object in JSON format having below fields:

* rank
* level
* name
* score

**Additional APIs and Specifications:**

* If we want to create or update a user including his userProgress at one go, we can use below APIs:

1. **createUserIncludingProgress**

**Description:** This API will create a new user and insert it into users table in the DB. Please note that, there is an OneToOne mapping with the user and userProgress. For this API, providing userProgress data in the request is mandatory. This will insert the user and his progress in the DB at one go.

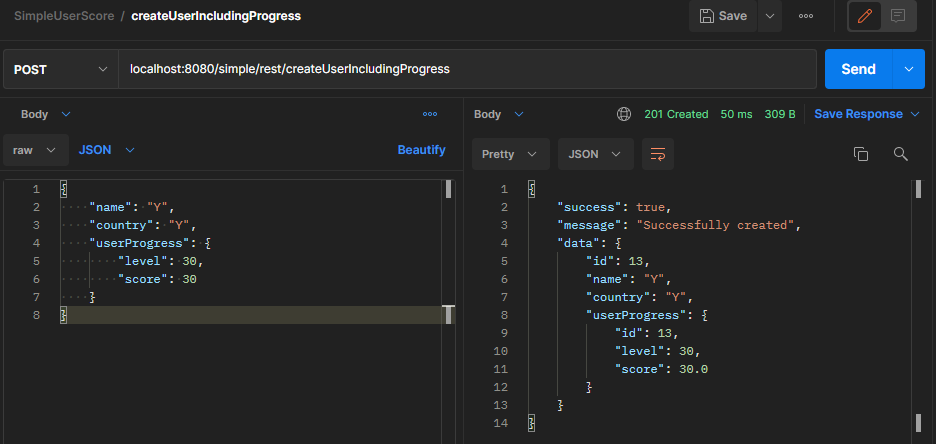
**Endpoint:** localhost:8080/simple/rest/createUserIncludingProgress

**Request Method:** POST

**Path Variables:** N/A

**Request Body:** raw JSON request body

**Sample Request and Response:**



**Response Description:** This response is designed to be wrapped with below fields to achieve meaningful insights:

* **success:** Represents if the operations is successful or not
* **message:** Represents the error message for this request.
* **data:** This field will contain the data resulted from the API request. In this case it is the newly created user in JSON format.

1. **updateUserIncludingProgress**

**Description:** This API will update an already created user. This can update either the data associated with user himself or user’s progress at one go. In request body it expects a field ‘id’ representing which user to update.

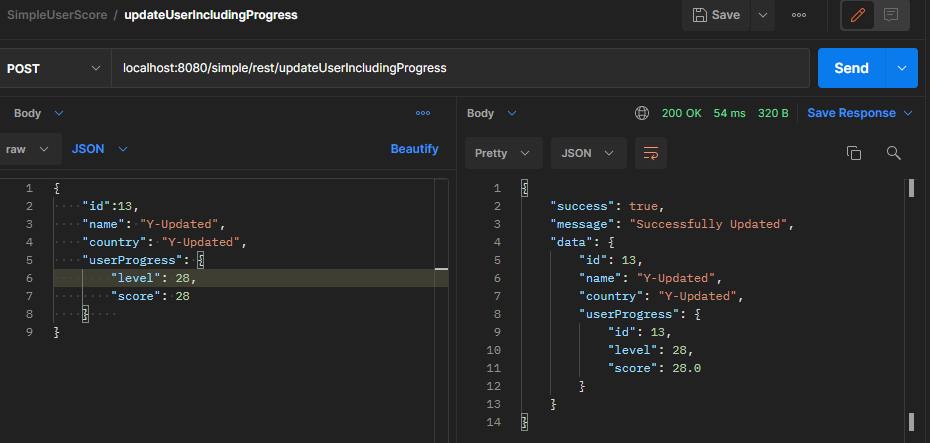
**Endpoint:** localhost:8080/simple/rest/updateUserIncludingProgress

**Request Method:** POST

**Path Variables:** N/A

**Request Body:** raw JSON request body

**Sample Request and Response:**



**Response Description:** This response is designed to be wrapped with below fields to achieve meaningful insights:

* **success:** Represents if the operations is successful or not
* **message:** Represents the error message for this request.
* **data:** This field will contain the data resulted from the API request. In this case it is the updated user in JSON format.