

# Tori restAPI documentation

## Requirements:

- Mysql database called 'toriapi'
- Cd to server && npm install
- After npm install, npm run migrate-up && npm run seed
- Api can be run locally with vscode debugger or node index.js in src folder.

## URLs:

- **POST** Localhost:8080/api/v1/auth/register
  - Register a new user
  - Input: (JSON string)
  - API checks validations:
  - Username doesn't exist, password length and password confirmation matches, email form is valid.

Username	String	Publicly visible username
Password	String	User password
Password confirmation	String	User password confirmation
Name	String	First and family name
Address	String	User address
City	String	User home city
Country	String	User home country, two letter code
Email	String	Valid email address for the user
Phone number	String	User phone number

## Body form

```
{
  "userName": "example_user",
  "password": "123456",
  "passwordConfirmation": "123456",
  "name": "Test name",
  "address": "kotkantie 1",
  "city": "Oulu",
  "country": "Fi",
  "email": "bob@test.com",
  "phoneNumber": "123213123",
}
```

## Response

- Status codes

201	"OK"
400	"Bad request" (missing data or validation had error)
403	"Forbidden" User "xxx" already exists. (Checks username and email to be unique)

- Body response JSON object

data	User registered.	
Error message	Validation error message	Contains error code, error message what went wrong

- **POST** Localhost:8080/api/v1/auth/token
  - Login existing user
  - Input: (JSON String)
  - API checks that username or email matches with hashed password

Username or Email	password
----------------------	----------

## Body form

```
{
  "username": "example_user",
  "password": "123456"
}
```

Or

```
{
  "username": "example@user.email ",
  "password": "123456"
}
```

## Response

- Status codes

200	"OK"
400	"Bad request" (missing data or validation had error)

- Response data structure description

JWT token	JWT token that can be used to fetch user data
-----------	---

- Body response JSON object

JWT token	Saved to local storage	Stays valid for one year -> requires login again to get new token
Error message	Unauthorized	Username didn't match password

- **POST** Localhost:8080/api/v1/posts/create-post
  - User can list item
  - Input: (JSON)

Title	Title of listing	String
Description	Description about the listing	String
Category	Which category the item belongs to	String
City	City	String
Country	Country code represented with 2 letters	String
Price	Price of the item	Number
Delivery	What type of delivery is available	String
Contact info	Sellers name and contact info	String

Body form

```
{
  "title": "Dishwasher",
  "description": "Working old dishwasher.",
  "category": "Electronics",
  "city": "Oulu",
  "country": "Fi",
  "price": 200,
  "delivery": "Pickup"
}
```

Response

- Status codes

201	"OK" Creating listing was successful.
400	"Bad request" (missing data or validation had error)

- Body response JSON object

Title		Title of the item
Description		Description of the item
Category	Category of the item	car, house, electronics... etc.
City	City	Where the item is at
Country	Country code	2 letter country code
Price		Price of the item
Delivery	Delivery options	Pickup, delivery, shipping etc.
Error message	Validation error message	Contains error code, error message what went wrong.

- **GET** Localhost:8080/api/v1/posts/all-posts
  - Fetch all posts

Response

- Status codes

200	"OK"
-----	------

- Response data structure description

All posts data		Title, image, location.
User data	Name, email, phone number	Returns data from user who has created the post

- Body response JSON object

Error message	Validation error message	Contains error code, error message what went wrong
Data about posts		Json response will have all values
Owner of post data	Name, email, phone number	Name of poster, email and phone number

- **GET** Localhost:8080/api/v1/posts/sort-by/category/:category
  - Fetch all posts by category

Response

- Status codes

200	"OK"
-----	------

- Response data structure description

All posts data where category matches		Id, itemownerId, title, description, category, city, country, images, price, delivery, created_at
User data	Name, email, phone number	Returns data from user who has created the post

- Body response JSON object

Error message	Validation error message	Contains error code, error message what went wrong
Data about posts		Json response will have all values, filtered by category
Owner of post data	Name, email, phone number	Name of poster, email and phone number

- **GET** Localhost:8080/api/v1/posts/sort-by/category/:city
  - Fetch all posts by city

Response

- Status codes

200	"OK"
-----	------

- Response data structure description

All posts data where category matches		Id, itemownerId, title, description, category, city, country, images, price, delivery, created_at
User data	Name, email, phone number	Returns data from user who has created the post

- Body response JSON object

Error message	Validation error message	Contains error code, error message what went wrong
Data about posts		Json response will have all values, filtered by city
Owner of post data	Name, email, phone number	Name of poster, email and phone number

- **GET** Localhost:8080/api/v1/posts/sort-by/category/date

- Fetch all posts by date newest first (descending order)

## Response

- Status codes

200	"OK"
-----	------

- Response data structure description

All posts data where category matches		Id, itemownerId, title, description, category, city, country, images, price, delivery, created_at
User data	Name, email, phone number	Returns data from user who has created the post

- Body response JSON object

Error message	Validation error message	Contains error code, error message what went wrong
Data about posts		Json response will have all values, filtered by date, newest first (descending order).
Owner of post data	Name, email, phone number	Name of poster, email and phone number

- **PUT** Localhost:8080/api/v1/posts/edit-post/:id

- Edit specific sensor data values

Post	Data values for title, description, category, city, country, price, delivery
------	--

## Body form

```
{
  "title": "old / new title",
  "description": " old / new description",
  "category": " old / new category",
  "city": " old / new city",
  "country": " old / new country",
  "price": old / new price as numbers,
  "delivery": " old / new delivery type"
}
```

## Response

- Status codes

20	"OK" Editing was successful
401	"Unauthorized" Person editing is not the owner of the post.

- Response data structure description

Post updated data	All data that belong to this post	All body from values + id and itemOwnerId
-------------------	-----------------------------------	---

- Body response JSON object

Error message	Validation error message	Contains error code, error message what went wrong
Post data	Separated data for post type, description, location,	This data will have the values for everything that, then can be displayed to the user

- **PUT** Localhost:8080/api/v1/posts/add-image/:id
  - Edit specific sensor data values

Post	Files images max 4
------	--------------------

## Response

- Status codes

200	"OK" Added images successfully.
401	"Unauthorized" Person editing is not the owner of the post.

- Body response JSON object

Error message	Validation error message	Contains error code, error message what went wrong
Post data		Added images successfully

- **DELETE** Localhost:8080/api/v1/posts/delete-post/:id
  - Delete post by id

#### Response

- Status codes

200	"OK" "Successfully deleted post"
401	"Unauthorized" Person editing is not the owner of the post.

- Body response JSON object

Error message	Validation error message	Contains error code, error message what went wrong
Data		"Successfully deleted post"