- Web Interfaces API -

Platform for selling and buying used products.

API done by:

* Ignacio Ferrer Sanz
* Alejandro Tauste Botella

Content table

[**Methods for users** 3](#_Toc33738935)

[**Methods for items** 4](#_Toc33738936)

[Methods only available for logged users 4](#_Toc33738937)

[Methods for non-logged users (and also for logged users) 4](#_Toc33738938)

## **Methods for users**

#### POST /users/register

Register a new user into the system.

*Request description*

|  |  |  |
| --- | --- | --- |
| userName | string | Publicly visible username |
| name | string | First and family name |
| password | string | Password created by the user. It will have a Bcrypt encryptation. |
| city | string | User home city |
| country | string | User home country |
| email | string | Valid email address for the user |
| phoneNumber | string | The telephone number of the user. |
| isAdmin | boolean | Indicates if the user is admin or not. |
| postings | array | Contains the id’s of all the postings this user has created. By default is null |

*Request body example*

{

"userName": "Nacchofer",

"name": "Ignacio Ferrer",

"password": "pepitoElPapas",

"city": "Valencia",

"country": "Spain",

"email": [fercho@demo.com](mailto:fercho@demo.com)

"isAdmin": false

"postings": []

}

*Response*

|  |  |
| --- | --- |
| 200 | User created successfully |
| 400 | Missing or incorrect data in the request |

*Response data structure description*

|  |  |  |
| --- | --- | --- |
| id\_ | number | The id assigned to the created user |

*Response body example*

It returns all the users that are registered.

[

{

"\_id": 1,

"userName": "admin",

"name": "admin",

"password": "$2b$10$qSIgVzuGagsuirGeBE4ZzuciNeJ4n8xiNJTqXf3VfHfdaP9eNd0Ty",

"city": "Valencia",

"country": "Spain",

"email": "example@mail.com",

"phoneNumber": "123456789",

"isAdmin": true,

"postings": [

1

]

},

{

"\_id": 2,

"userName": "User1",

"name": "User1",

"password": "$2b$10$qSIgVzuGagsuirGeBE4ZzuciNeJ4n8xiNJTqXf3VfHfdaP9eNd0Ty",

"city": "Oulu",

"country": "Finland",

"email": "example@mail.com",

"phoneNumber": "132465798",

"isAdmin": false,

"postings": []

},

{

"\_id": 3,

"userName": "Pacorro",

"name": "Loco",

"city": "Lolo",

"country": "Lelo",

"email": "paco@gmail.com",

"isAdmin": true,

"password": "$2b$10$ehx/h1HAcHtZHNMl5CpogOmxGbciAz8KASPIdsYV97L.JkTPbKYx6"

}

]

|  |  |
| --- | --- |
| 200 | User created successfully |
| 400 | Missing or incorrect data in the request |

#### POST /users/login

This method lets the user to login and then it becomes a “logged-user”.

*Request description*

It is the same as in the register method, but "userName" and "password", are mandatory

*Response body example*

It returns the jwt token for the user to be able to do some methods for managing postings.

*Response*

{

"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyTmFtZSI6IlBhY29ycm8iLCJuYW1lIjoiTG9jbyIsImNpdHkiOiJMb2xvIiwiY291bnRyeSI6IkxlbG8iLCJpc0FkbWluIjp0cnVlLCJlbWFpbCI6InBhY29AZ21haWwuY29tIiwiaWF0IjoxNTgyODQwMTgyLCJleHAiOjE1ODM0NDQ5ODJ9.7V8vU6YckDiM9Bi3YJGtZ8Y1K1c2ks\_LEy3J0D95VgE"

}

*Status codes*

|  |  |
| --- | --- |
| 200 | User logged in successfully |
| 400 | Password does not match. |

## **Methods for items**

### Methods only available for logged users

#### POST /users/items/create

Register a new posting in the system

|  |  |  |
| --- | --- | --- |
| title | string | Name of the item to be sold |
| description | string | Short description of the item |
| category | string | Name of the category in which this item can be found. |
| ***location(\*)*** | object | This object contains different values of the location of posting such as city, country, etc.. |
| images | array | Contains the images of the item (Max. 4 pictures) |
| askingPrice | number | Price for the posting |
| currency | string | Type of currency. It can be “euro” or “dollar”. |
| dateOfPosting | string | Date of posting of the item |
| deliveryType | string | Type of delivery in case that the item is sold. It can be *Shipping* or  *Pickup.* |
| contactInfo | object | Information for contacting the seller. It contains phone number and e-mail |

In the case of the object ***location(\*)***, it contains the following values (inherited from the user who created the posting):

|  |  |  |
| --- | --- | --- |
| city | string | Name of the city where the posting is. |
| country | string | Name of the country of the posting |

In the case of the array “*images*” it contains a list of maximum 4 objects, (image) which has these values:

*To be continued….*

In the case of the object “*contactInfo”* it contains the following values (inherited from the user who created the posting:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| sellerName | string | | Name of the seller and family name. | |
| phoneNumber | | number | | Number of the seller |
| sellerEmail | | string | | E-mail of the seller |

The date of the posting will be added after successfully creating the item by putting the actual date.

*Request body example*

{

"title": "Teddy bear",

"description": "Teddy bear",

"category": "Rasberry PI",

"images": [],

"askingPrice": 36,

"currency": "euro",

"deliveryType": "Shipping",

}

*Response*

Status codes

|  |  |
| --- | --- |
| 200 | Item created successfully |
| 400 | Missing or incorrect data in the request |

*Response data structure description*

|  |  |  |
| --- | --- | --- |
| \_id | number | The id assigned to the created posting |

*Response body example*

{

"\_id": 1111111

}

This id will be automatically stored in the array ***postings***, of the user who created it.

#### GET /users /items/search/:\_id

#### PUT /users /ítems/modify/:\_id

#### DELETE /users /items/:\_id

A posting will be deleted from the database, and the reference to this posting (found in the posting array of each user) will also be deleted.

There is no body for this method.

Status codes

|  |  |
| --- | --- |
| 200 | Item has been deleted |
| 400 | Item is not from the user who wants to delete it |

The response will be the rest of the ítems in the database.

*Response body example:*

[

{

"\_id": 1,

"title": "Teddy Bear",

"description": "A small teddy bear",

"category": "Toy",

"location": {

"city": "Valencia",

"country": "Spain"

},

"images": [],

"askingPrice": "10",

"currency": "euro",

"dateOfPosting": "13.01.2020",

"deliveryType": "Shipping",

"sellerName": "admin",

"phoneNumber": "123456789",

"sellerEmail": "example@mail.com"

},

{

"\_id": 2,

"title": "HOIGRHAORAGAGA",

"description": "YES",

"category": "NOTHING",

"location": {

"city": "Lolo",

"country": "Lelo"

},

"images": [],

"askingPrice": "5",

"currency": "euro",

"dateOfPosting": "Thu Feb 27 2020",

"deliveryType": "Shipping",

"sellerName": "Loco",

"sellerEmail": "paco@gmail.com"

}

]

### Methods for non-logged users (and also for logged users)

#### GET /users/search\_category/:category

This operation will let any user search for postings that match the same category that is specified in the URL, so therefore there is no request body.

#### GET /users/search\_location

This operation will let any user to search for postings that matches any of the atributes of the location specified in the request body.

*Request body example*

location: {

city: "Madrid",

country: "Spain"

}

#### GET /users/search\_date

This operation will let any user search for postings that match the same date in which the posting was created. The date will be specified in the request body.

{

"dateOfPosting": "Thu Feb 27 2020"

}

#### Status codes for the search methods

Status codes

|  |  |
| --- | --- |
| 200 | Search was done successfully |
| 404 | No results have been found during the search |