

QUICKWORK

DOCUMENTATION REST

UserController (/api/users)

Get all users

Get all users information except the password **(THIS IS ONLY USED FOR DEBUG AND WILL NEVER BE USED IN THE APP)**

/api/users (**get**)

success:

```
[
  {
    "id": 14,
    "name": "Luis Jose Maria",
    "email": "ljm@sapo.pt",
    "jobnumber": 84,
    "rating": 3
  },
  ...
]
```

Error:

No message available

Example:

```
public Iterable<User> getUsers() {
    Logger.info("Sending all units");
    return userRepository.findAll();
}
```

Add user

Adds a user when a user signs in

/api/users (**POST**)

Post example:

```
{
  "name": "name",
  "email": "email",
  "jobnumber":0,
  "password":"password"
}
```

When adding a new user on the app, (for debug purposes it's irrelevant), "jobnumber" must always be 0

Success:

```
{
  "id": id,
  "name": "name",
  "email": "email",
  "jobnumber":0,
  "rating": null
}
```

Error:

No message available

Example:

```
@PostMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public User saveUsr(@RequestBody User usr) {
    Logger.info("User named: "+usr.getName()+" saved");
    return userRepository.save(usr);
}
```

Remove a user

Removes a specific user

/api/users/:id **(DELETE)**

Parameters:

Id: Id of the user

Success:

Error:

“No value presente”

Example:

```
@DeleteMapping(path="/{usrid}", produces = MediaType.APPLICATION_JSON_VALUE)
public void delete(@PathVariable("usrid") int usrid){
    User user;
    user = userRepository.findById(usrid).get();
    String name = user.getName();

    Logger.info("Delete user "+name);
    userRepository.deleteById(usrid);
}
```

Get a user

Removes a specific user

/api/users/:id (get)

Parameters:

Id: Id of the user

Success:

```
{
  "id": 18,
  "name": "Apolonia Casimiro",
  "email": "AC@gmail.com",
  "jobnumber": 6,
  "rating": 5
}
```

Error:

null

Example:

```
@GetMapping(path = "/{usrid}", produces = MediaType.APPLICATION_JSON_VALUE)
public Optional<User> getUser(@PathVariable("usrid") int usrid){
    Logger.info("User with id: " + usrid + " given");
    return userRepository.findById(usrid);
}
```

Get a work's creator

Receives the id of a job and send back the owner

/api/users/owner/:id (**get**)

Parameters:

Id: Id of a job

Success:

```
{
  "id": 14,
  "name": "Luis Jose Maria",
  "email": "ljm@sapo.pt",
  "jobnumber": 84,
  "rating": 3
}
```

Error:

No message available

Example:

```
@GetMapping(path = "/owner/{id}", produces = MediaType.APPLICATION_JSON_VALUE)
public Optional<User> getowner(@PathVariable("id") int id) throws
NotFoundException{
    Logger.info(id+ "");
    Optional<getownerview> _ownerid = userRepository.getownerid1(id);
    if(_ownerid.isEmpty()){
        throw new NotFoundException();
    }else{
        getownerview _owner = _ownerid.get();
        return userRepository.findById(_owner.getownerid());
    }
}
```

WorkController(api/Work)

Get all AVAILABLE jobs for display on the map

This will send all jobs that have no starting date, and will only send the id, location and type

/api/work (get)

Success:

```
[
  {
    "type": "Limpeza",
    "lat": 38.7115487417554,
    "lon": -9.159549968757467,
    "id": 2
  }, ...
]
```

Error:

No message available

Example:

```
@GetMapping(path = "", produces = MediaType.APPLICATION_JSON_VALUE)
public Iterable<Workmapview> getworks(){

    Logger.info("sending all jobs");

    return workrepository.workmapshow();
}
```

Get all information of a specific work

Send all the information of a specific work

/api/work /:id (get)

Parameters:

Id: Id of a work

Success:

```
{
  "id": 2,
  "pricehr": 23.0,
  "tip": null,
  "started_time": null,
  "finished_time": null,
  "typeid": 2,
  "lat": 38.7115487417554,
  "lon": -9.159549968757467
}
```

Error:

No message available

Example:

```
@GetMapping(path =("/{id}", produces = MediaType.APPLICATION_JSON_VALUE)
public Optional<Work> getwork(@PathVariable("id") int id) {
    Logger.info("Sending "+ id);
    return workrepository.findById(id);
}
```


Get the type of a work

Gives the type of a specified work

/api/work/type/:id (**get**)

Parameters:

Id: Id of a work

Success:

```
{
  "name": "Limpeza"
}
```

Error:

No message available

Example:

```
@GetMapping(path = "/type/{id:[2-5]}", produces = MediaType.APPLICATION_JSON_VALUE)
public getView gettype(@PathVariable("id") int id){

    return workrepository.gettype(id);
}
```