**QUICKWORK**

**DOCUMENTATION REST**

UsersController (/api/users)

|  |
| --- |
| **Get Location**  Get a user location |
| /api/users/{id}/location (**get**) |
| Parameters:  {id} = User id |
| success:  {      "lat": null,      "lon": null  } |
| Error:  {      "timestamp": "2023-01-05T10:52:33.114+00:00",      "status": 404,      "error": "Not Found",      "message": "User with id 50 not found.",      "path": "/api/users/50/location"  } |
| Example:       @GetMapping(path = "/{usrid}/location", produces =  MediaType.APPLICATION\_JSON\_VALUE)      public UsrLocview getLocation(@PathVariable("usrid") int id) throws NotFoundException{          Optional<User> \_user = userRepository.findById(id);          if(!\_user.isEmpty()){              return userRepository.getlocation(id);          }else throw new NotFoundException(""+id, "User", "id");      } |

|  |
| --- |
| **Update User Location**  Updates user location |
| /api/users/{id}/location **(PATCH)** |
| Parameters:  {id} = user id |
| Patch example:  {      "lat": 3.603,      "lon": 2.14  } |
| Success: |
| Error:  No message available |
| Example:  @PatchMapping(path = "/{usrid}/location", produces =  MediaType.APPLICATION\_JSON\_VALUE)      public void setLocation(@PathVariable("usrid") int id, @RequestBody usrloc loc){              userRepository.setlocation(id, loc.getLat(), loc.getLon());      } |

|  |
| --- |
| **Add +1 job to a user** |
| /api/users/{id}/addjob **(PATCH)** |
| Parameters:  {id} = user id |
| Patch example: |
| Success: |
| Error:  No message available |
| Example:  @PatchMapping(path = "/{usrid}/addjob", produces =  MediaType.APPLICATION\_JSON\_VALUE)      public void addnjob(@PathVariable("usrid") int id){          User user = userRepository.findById(id).get();          int njob = user.getJobnumber();          userRepository.setnjob(id, njob+1);      } |

|  |
| --- |
| **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) {    User \_user = usr;          \_user.setJobnumber(0);          Logger.info("User named: "+usr.getName()+" saved");          return userRepository.save(usr);      } |

|  |
| --- |
| **Owner of a work** |
| /api/users/owner/{id} **(Get)** |
| Parameters:  {Id} = Id of the user |
| Success:  {      "id": 5,      "name": "Apolonia Casimiro",      "email": "AC@gmail.com",      "jobnumber": 6,      "rating": 5  } |
| Error:  {      "timestamp": "2023-01-05T11:22:05.212+00:00",      "status": 404,      "error": "Not Found",      "message": "work with id 50 not found.",      "path": "/api/users/owner/50"  } |
| 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(""+id, "work", "id");          }else{              getownerview \_owner = \_ownerid.get();               return userRepository.findById(\_owner.getownerid());          }      } |

|  |
| --- |
| **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);      } |

|  |
| --- |
| **Is User Occupied** |
| /api/users/{id}/occupied **(get)** |
| Parameters:  {id} = id |
| Success:  {      "haswork": false,      "isowner": false  } |
| Error:  {      "timestamp": "2023-01-05T11:29:45.592+00:00",      "status": 404,      "error": "Not Found",      "message": "User with Id 54 not found.",      "path": "/api/users/jobs/54"  } |
| Example:   @GetMapping(path ="/{usrid}/occupied", produces = MediaType.APPLICATION\_JSON\_VALUE)      public usrworkinfo getBoolean(@PathVariable("usrid") int usrid){          usrworkinfo response = null;          if( userRepository.isUserOccupied(usrid)){              response = new usrworkinfo(true, userRepository.isUserowner(usrid));          }else{ response = new usrworkinfo(false);}          return response;      } |

|  |
| --- |
| **Users history** |
| /api/users/job/{id} |
| Parameters:  {id} = id |
| Success:  {      "haswork": false,      "isowner": false  } |
| Error:  No message available |
| Example:   @GetMapping(path = "/jobs/{usrid}", produces = MediaType.APPLICATION\_JSON\_VALUE)        public Iterable<UsrJobsView> getJobs(@PathVariable("usrid") int usrid) throws NotFoundException{              Optional<User> \_user = userRepository.findById(usrid);              if(!\_user.isEmpty()){              Iterable<UsrJobsView> jobs = userRepository.findusrjobs(usrid);              Logger.info("Sending All users "+ usrid+" jobs");              return jobs;              }else throw new NotFoundException(""+usrid, "User", "Id");   } |

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 ArrayList<Workmapview> getworks(){          ArrayList<Workmapview> \_workmap = workrepository.workmapshow();          ArrayList<Workmapview> workmap = new ArrayList<>();          for(int i = 0; i< \_workmap.size(); i++){                  Workmapview temp = \_workmap.get(i);                  String state = workrepository.getState(temp.getid());                  if(state.equals("Em espera")){                      workmap.add(temp);                  }              }          return workmap;      } |

|  |
| --- |
| **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:  {      "timestamp": "2023-01-05T11:32:35.741+00:00",      "status": 404,      "error": "Not Found",      "message": "id with Work 37 not found.",      "path": "/api/work/37"  } |
| Example:  @GetMapping(path = "/{workid}", produces = MediaType.APPLICATION\_JSON\_VALUE)      public workview getwork(@PathVariable("workid") int id)throws NotFoundException{          workview \_work = workrepository.getworkandtype(id);          if( \_work != null) {              Logger.info("Sending "+ id);              return \_work;          }else throw new NotFoundException(""+id, "id", "Work" );      } |

|  |
| --- |
| **Get all users from work**  Give all users that are working on a work |
| /api/work/users/{id} **(get)** |
| Parameters:  {id} = Id of a work |
| Success:  [      {          "id": 9,          "rating": 4,          "jobnumber": 2,          "email": "UT@outlook.com",          "name": "Urbano Tao"      },…  ] |
| Error:  {      "timestamp": "2023-01-05T11:35:55.060+00:00",      "status": 404,      "error": "Not Found",      "message": "id with work 90 not found.",      "path": "/api/work/users/90"  } |
| Example:      @GetMapping(path = "/users/{workid}", produces = MediaType.APPLICATION\_JSON\_VALUE)      public ArrayList<customusrview> getusers(@PathVariable("workid") int id)throws NotFoundException{          Optional<Work> \_work= workrepository.findById(id);          if (\_work.isPresent()){              ArrayList<customusrview> test = workrepository.getusers(id);          return test;          } else throw new NotFoundException(""+id, "id", "work" );      } |