

## API Documentation

---

Below is a short description of the different API endpoints in the server. There is also examples of how a valid JSON object should look like when sending and what kind of objects are returned.

### Users

GET Request to get users without details

GET [/users/](#)

- JSON Response body:

```
{
  "users": [
    {
      "id": 1,
      "facebookId": null,
      "username": "sam123",
      "email": "sam.gamgee@lotr.com",
      "active": 1
    },
    {
      "id": 9,
      "facebookId": 123456789123456,
      "username": "Frodo.baggins",
      "email": "frodo.baggins@lotr.org",
      "active": 1
    }
  ]
}
```

GET Request to get user by id

GET [/users/1](#)

- JSON Response body:

```
{
  "user": {
    "id": 1,
    "facebookId": null,
    "username": "sam123",
    "email": "sam.gamgee@lotr.com",
    "active": 1
  }
}
```

## GET Request to get users details with user id

GET /users/1/details

- JSON Response body:

```
{
  "details": {
    "userId": 1,
    "detailsId": 8,
    "firstName": "Sam",
    "lastName": "Gamgee",
    "suburb": "Kazadum",
    "zipcode": 66611
  }
}
```

## GET Request to get user with details by user id

GET /users/1/all

- JSON Response body:

```
{
  "user": {
    "id": 1,
    "facebookId": 123456789123456,
    "username": "sam123",
    "email": "sam.gamgee@lotr.com",
    "active": 1,
    "details": {
      "userId": 1,
      "detailsId": 9,
      "firstName": "Sam",
      "lastName": "Gamgee",
      "suburb": "Kazadum",
      "zipcode": 66611
    }
  }
}
```

## POST Request to create user

POST /users/

- Mandatory data: username, email, password
- JSON Request body:

```
{
  "username": "Gimli",
  "email": "gimli.gloin@lotr.com",
  "password": "test123"
}
```

- JSON Response body on success:

```
{
  "success": "Successfully registered! Verify your email to activate your account and log in."
}
```

PUT Request to update user by id

PUT /users/1

- Requires auth
- Sensitive data cannot be updated through this: email, password, activationToken, active and all ID:s
- Optional data: username, firstname, lastname, hood, zip
- JSON Request body:

```
{
  "username": "GimliDwarf",
  "firstname": "Gimli",
  "lastname": "Gloin",
  "hood": "Moria",
  "zip": 12345
}
```

- JSON Response body:

```
{
  "user": {
    "id": 1,
    "facebookId": null,
    "username": "GimliDwarf",
    "email": "gimli.gloin@lotr.com",
    "active": 0
  },
  "success": "Successfully updated!"
}
```

DELETE Request to delete user by id

DELETE /users/1

- Requires auth
- JSON Response body:

```
{
  "success": "Successfully deleted your account!"
}
```

GET Request to verify users email, user gets token in email message

GET /users/1/verify?activationToken=<activationToken>

- On success sets users field active = 1, activationToken = "". Then redirects to login page (URL hardcoded in UsersController activateUser() -method)
- JSON Response body on error:

```
{
  "error": "Activation tokens did not match!"
}
```

## Reviews

GET request to fetch all reviews without the ratings (ratings are only displayed when opening a specific review)

GET /reviews/

- JSON Response body:

```
{
  "reviews": [
    {
      "reviewId": 2,
      "hoodId": 666,
      "userId": 7,
      "title": "Test Review From Mordor",
      "text": "One Ring to rule them all, One Ring to find them, One Ring to bring them all and in the darkness bind them.",
      "verified": 0
    }
  ]
}
```

GET request to fetch a review with id of 2

GET /reviews/2

- JSON Response body:

```
{
  "review": {
    "reviewId": 2,
    "hoodId": 666,
    "userId": 7,
    "title": "Test Review From Mordor",
    "text": "One Ring to rule them all, One Ring to find them, One Ring to bring them all and in the darkness bind them.",
    "verified": 0,
    "ratings": [

```

```
{
  "ratingId": 9,
  "reviewId": 2,
  "value": 5,
  "category": {
    "categoryId": 1,
    "name": "safety"
  }
},
{
  "ratingId": 10,
  "reviewId": 2,
  "value": 1,
  "category": {
    "categoryId": 6,
    "name": "services"
  }
},
{
  "ratingId": 11,
  "reviewId": 2,
  "value": 1,
  "category": {
    "categoryId": 7,
    "name": "activities"
  }
},
{
  "ratingId": 12,
  "reviewId": 2,
  "value": 2,
  "category": {
    "categoryId": 4,
    "name": "school"
  }
},
{
  "ratingId": 13,
  "reviewId": 2,
  "value": 1,
  "category": {
    "categoryId": 8,
    "name": "family friendly"
  }
},
{
  "ratingId": 14,
  "reviewId": 2,
  "value": 1,
  "category": {
    "categoryId": 5,
    "name": "transportation"
  }
},
{
  "ratingId": 15,
  "reviewId": 2,
  "value": 4,
  "category": {
```

```

        "categoryId": 2,
        "name": "cleanliness"
      }
    },
    {
      "ratingId": 16,
      "reviewId": 2,
      "value": 3,
      "category": {
        "categoryId": 3,
        "name": "nature"
      }
    }
  ]
}

```

GET request to fetch reviews from specific hood

**GET** /reviews/hood/1

- JSON Response body:

```

{
  "reviews": [
    {
      "reviewId": 2,
      "hoodId": 1,
      "userId": 7,
      "title": "Review From Mordor",
      "text": "One Ring to rule them all, One Ring to find them, One Ring to
bring them all and in the darkness bind them!",
      "verified": 0,
      "ratings": [
        {
          "ratingId": 9,
          "reviewId": 2,
          "value": 5,
          "category": {
            "categoryId": 1,
            "name": "safety"
          }
        },
        {
          "ratingId": 10,
          "reviewId": 2,
          "value": 1,
          "category": {
            "categoryId": 6,
            "name": "services"
          }
        }
      ],
      {
        "ratingId": 11,
        "reviewId": 2,
        "value": 1,
        "category": {

```

```

        "categoryId": 7,
        "name": "activities"
    },
    {
        "ratingId": 12,
        "reviewId": 2,
        "value": 2,
        "category": {
            "categoryId": 4,
            "name": "school"
        }
    },
    {
        "ratingId": 13,
        "reviewId": 2,
        "value": 1,
        "category": {
            "categoryId": 8,
            "name": "family friendly"
        }
    },
    {
        "ratingId": 14,
        "reviewId": 2,
        "value": 1,
        "category": {
            "categoryId": 5,
            "name": "transportation"
        }
    },
    {
        "ratingId": 15,
        "reviewId": 2,
        "value": 4,
        "category": {
            "categoryId": 2,
            "name": "cleanliness"
        }
    },
    {
        "ratingId": 16,
        "reviewId": 2,
        "value": 3,
        "category": {
            "categoryId": 3,
            "name": "nature"
        }
    }
]
}

```

POST request to create a review entry to database

POST /reviews/

- Mandatory request data: hoodId, userId, title, text, ratings (from all categories)
- JSON Request body:

```
{
  "hoodId": 666,
  "userId": 7,
  "title": "Test Review From Mordor",
  "text": "One Ring to rule them all, One Ring to find them, One Ring to bring
them all and in the darkness bind them.",
  "ratings": [
    {
      "value": 5,
      "category": {
        "categoryId": 1,
        "name": "safety"
      }
    },
    {
      "value": 4,
      "category": {
        "categoryId": 2,
        "name": "cleanliness"
      }
    },
    {
      "value": 3,
      "category": {
        "categoryId": 3,
        "name": "nature"
      }
    },
    {
      "value": 2,
      "category": {
        "categoryId": 4,
        "name": "school"
      }
    },
    {
      "value": 1,
      "category": {
        "categoryId": 5,
        "name": "transportation"
      }
    },
    {
      "value": 1,
      "category": {
        "categoryId": 6,
        "name": "services"
      }
    },
    {
      "value": 1,
```



```

        "category": {
            "categoryId": 7,
            "name": "activities"
        }
    },
    {
        "value": 1,
        "category": {
            "categoryId": 8,
            "name": "family friendly"
        }
    }
]
}

```

- JSON response body:

```

{
  "review": {
    "reviewId": 2,
    "hoodId": 666,
    "userId": 7,
    "title": "Test Review From Mordor",
    "text": "One Ring to rule them all, One Ring to find them, One Ring to bring
them all and in the darkness bind them.",
    "verified": 0,
    "ratings": [
      {
        "ratingId": 9,
        "value": 5,
        "category": {
          "categoryId": 1,
          "name": "safety"
        }
      },
      {
        "ratingId": 10,
        "value": 1,
        "category": {
          "categoryId": 6,
          "name": "services"
        }
      },
      {
        "ratingId": 11,
        "value": 1,
        "category": {
          "categoryId": 7,
          "name": "activities"
        }
      },
      {
        "ratingId": 12,
        "value": 2,
        "category": {
          "categoryId": 4,
          "name": "school"
        }
      }
    ]
  }
}

```

```

    {
      "ratingId": 13,
      "value": 1,
      "category": {
        "categoryId": 8,
        "name": "family friendly"
      }
    },
    {
      "ratingId": 14,
      "value": 1,
      "category": {
        "categoryId": 5,
        "name": "transportation"
      }
    },
    {
      "ratingId": 15,
      "value": 4,
      "category": {
        "categoryId": 2,
        "name": "cleanliness"
      }
    },
    {
      "ratingId": 16,
      "value": 3,
      "category": {
        "categoryId": 3,
        "name": "nature"
      }
    }
  ]
}

```

PUT Request to update review with id of 2

**PUT** `/reviews/2`

- Mandatory Request data: ratings
- Optional Request data: title, text
- JSON Request body:

```

{
  "title": "Review From Mordor",
  "text": "One Ring to rule them all, One Ring to find them, One Ring to bring them all and in the darkness bind them!",
  "ratings": [
    {
      "ratingId": 9,
      "reviewId": 2,
      "value": 5,
      "category": {
        "categoryId": 1,
        "name": "safety"
      }
    }
  ]
}

```

```
    }  
  },  
  {  
    "ratingId": 10,  
    "reviewId": 2,  
    "value": 1,  
    "category": {  
      "categoryId": 6,  
      "name": "services"  
    }  
  },  
  {  
    "ratingId": 11,  
    "reviewId": 2,  
    "value": 1,  
    "category": {  
      "categoryId": 7,  
      "name": "activities"  
    }  
  },  
  {  
    "ratingId": 12,  
    "reviewId": 2,  
    "value": 2,  
    "category": {  
      "categoryId": 4,  
      "name": "school"  
    }  
  },  
  {  
    "ratingId": 13,  
    "reviewId": 2,  
    "value": 1,  
    "category": {  
      "categoryId": 8,  
      "name": "family friendly"  
    }  
  },  
  {  
    "ratingId": 14,  
    "reviewId": 2,  
    "value": 1,  
    "category": {  
      "categoryId": 5,  
      "name": "transportation"  
    }  
  },  
  {  
    "ratingId": 15,  
    "reviewId": 2,  
    "value": 4,  
    "category": {  
      "categoryId": 2,  
      "name": "cleanliness"  
    }  
  },  
  {  
    "ratingId": 16,  
    "reviewId": 2,
```

```

        "value": 3,
        "category": {
            "categoryId": 3,
            "name": "nature"
        }
    }
]
}

```

- JSON Response body:

```

{
  "review": {
    "reviewId": 2,
    "hoodId": 666,
    "userId": 7,
    "title": "Review From Mordor",
    "text": "One Ring to rule them all, One Ring to find them, One Ring to bring
them all and in the darkness bind them!",
    "verified": 0,
    "ratings": [
      {
        "ratingId": 9,
        "reviewId": 2,
        "value": 5,
        "category": {
          "categoryId": 1,
          "name": "safety"
        }
      },
      {
        "ratingId": 10,
        "reviewId": 2,
        "value": 1,
        "category": {
          "categoryId": 6,
          "name": "services"
        }
      },
      {
        "ratingId": 11,
        "reviewId": 2,
        "value": 1,
        "category": {
          "categoryId": 7,
          "name": "activities"
        }
      },
      {
        "ratingId": 12,
        "reviewId": 2,
        "value": 2,
        "category": {
          "categoryId": 4,
          "name": "school"
        }
      },
      {
        "ratingId": 13,

```

```

        "reviewId": 2,
        "value": 1,
        "category": {
            "categoryId": 8,
            "name": "family friendly"
        }
    },
    {
        "ratingId": 14,
        "reviewId": 2,
        "value": 1,
        "category": {
            "categoryId": 5,
            "name": "transportation"
        }
    },
    {
        "ratingId": 15,
        "reviewId": 2,
        "value": 4,
        "category": {
            "categoryId": 2,
            "name": "cleanliness"
        }
    },
    {
        "ratingId": 16,
        "reviewId": 2,
        "value": 3,
        "category": {
            "categoryId": 3,
            "name": "nature"
        }
    }
]
},
"success": "Successfully updated!"
}

```

DELETE Request to delete review with id 1

**DELETE** /reviews/1

- JSON Response body:
- On Success

```

{
  "success": "Successfully deleted your rating!"
}

```

- When user not found

```

{
  "error": "Not Found"
}

```

## Authentication

POST request to login using local strategy

**POST** /auth/login

- Mandatory data: email, password
- JSON Request body:

```
{
  "email": "frodo.baggins@lotr.com",
  "password": "test123!"
}
```

- JSON Response body:

```
{
  "user": {
    "id": 1,
    "username": "frodo123",
    "email": "frodo.baggins@lort.com",
    "active": 1
  },
  "accessToken":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6NywiYWV0IjoxNTg3MDYzNzA5LCJleHAiOi0jE1ODcwNjQ2MDl9.mPKcXqaV893oyOTQZnb9nd7Lf1ScKmoC_GVGI_wakps",
  "refreshToken":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjo3LCJpYXQiOi0jE1ODcwNjM3MDksImV4cCI6MTU4NzE1MDEwOX0.YDDceIjX1_-iJ51Q_c17T7cLNUZjjI-DtpgTZE8zs6A"
}
```

GET request that redirects to facebook logging in page

**GET** /auth/facebook

GET request will be called as callback when user provides facebook credentials in logging page

**GET** /auth/facebook/callback

- Registers user with a facebook id
- If user already exists with the facebook id, user will just be authenticated
- Redirects a successful login to client login page and passes tokens and user data through query params:  
'localhost:4200/login?jwt=&refreshToken=&id=&username=&email=&active='
- If user with provided email exists, but does not have a facebook id (account linked with facebook), it returns JSON Response (user can not be linked with facebook account through Account settings yet):

```
{
  "message": "There is already an account using this email address. Sign in to that account and link it with Facebook manually from Account Settings."
}
```

POST request to generate a new access- and refresh-token

POST /auth/refresh-token

- JSON Response body:

```
{
  "accessToken":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6NywiYW0IjoxNTg3MDYzNzA5LCJleHAiOi0jE1ODcwNjQ2MDI9.mPKcXqaV893oyOTQZnb9nd7Lf1ScKmoC_GVGI_wakps",
  "refreshToken":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyIjo3LCJpYXQiOi0jE1ODcwNjM3MDksImV4cCI6MTU4NzE1MDEwOX0.YDDceIjX1_-iJ51Q_c17T7cLNuZjjI-DtpgTZE8zs6A"
}
```