

All endpoints are available at `/api`

Request data is expected to be provided as JSON body.

## Endpoints

---

### User Authentication

- **Register**

`POST /api/register`

Request:

```
{
  name: required,
  email: required,
  password: required,
  role: required
}
```

`role` must be one of `admin`, `mentor` or `student`

Success response:

```
{
  message
}
```

- **Login**

`POST /api/login`

Request:

```
{
  email: required,
  password: required
}
```

Success response:

```
{
  message,
  token,
  uid
}
```

`token` refers to the JWT authentication token that should be saved and has to be included with the `Authorization` header for endpoints that require user authentication. `uid` refers to the user ID of the authenticated user.

- **Verify a token**

`GET /api/verify`

Headers:

```
Authorization: 'Bearer $token'
```

`$token` refers to the JWT token Request:

```
None
```

Success response:

```
{
  message
}
```

### Assessments

- **Fetch all assessments**

GET /api/assessment

Headers:

```
Authorization: 'Bearer $token'
```

`$token` refers to the JWT token

Request:

```
None
```

Success response:

```
{
  message,
  data
}
```

`data` is a list/array of all assessments

- **Create new assessment**

POST /api/assessment/create

Headers:

```
Authorization: 'Bearer $token'
```

`$token` refers to the JWT token

Request:

```
{
  title: required,
  description: required,
  mentorId: optional,
  deadline: optional
}
```

`deadline` refers to a valid datetime string.

`mentorId` has to be passed only if an `admin` is creating an assessment on behalf of a mentor.

Success response:

```
{
  message,
  id
}
```

`id` refers to the unique ID of the assessment created.

- **Update/edit an assessment**

*Only an `admin` can edit an assessment.*

PUT /api/assessment/:id

`:id` refers to the unique ID of the assessment to be updated.

Headers:

```
Authorization: 'Bearer $token'
```

`$token` refers to the JWT token

Request:

```
{
  updated: {
    title: optional,
    description: optional
  }
}
```

Success response:

```
{
  message
}
```

- **Delete an assessment**

*Only an `admin` can delete an assessment.*

DELETE /api/assessment/:id

:id refers to the unique ID of the assessment to be deleted.

Headers:

```
Authorization: 'Bearer $token'
```

\$token refers to the JWT token

Request:

```
None
```

Success response:

```
{
  message
}
```

## Submissions

- **Fetch submissions for an assessment**

GET /api/assessment/:aid/submissions

:aid refers to the unique ID of the assessment.

Headers:

```
Authorization: 'Bearer $token'
```

\$token refers to the JWT token

Request:

```
None
```

Success response:

```
{
  message,
  data
}
```

data is a list/array of relevant submissions. In case of a `mentor`, all submissions for an assessment are returned. In case of a `student`, only their submission for the assessment is returned.

- **Add a submission**

*Only an `admin` or a `student` can add a submission.*

POST /api/assessment/:id/submit

:id refers to the unique ID of the assessment to be submitted against.

Headers:

```
Authorization: 'Bearer $token'
```

`$token` refers to the JWT token

Request:

```
{
  files: [ required ],
  studentId: optional,
  grade: { optional }
}
```

`files` is a list/array of URL-as-strings.

`studentId` has to be passed only if an `admin` is adding a submission on behalf of a student.

`grade` is a object in the form of `{ marks: number, remarks: string }`. This can only be passed if the authenticated user is an admin.

Success response:

```
{
  message,
  id
}
```

`id` refers to the unique ID of the submission created.

- **Update/edit a submission**

*Only an `admin` can edit a submission*

PUT `/api/assessment/:aid/:sid`

`:aid` refers to the unique ID of the assessment. `:sid` refers to the unique ID of the submission to be updated.

Headers:

```
Authorization: 'Bearer $token'
```

`$token` refers to the JWT token

Request:

```
{
  updated: {
    files: [ optional ]
  }
}
```

`files` is a list/array of URL-as-strings.

Success response:

```
{
  message
}
```

- **Delete a submission**

*Only an `admin` can delete a submission*

DELETE `/api/assessment/:aid/:sid`

`:aid` refers to the unique ID of the assessment. `:sid` refers to the unique ID of the submission to be updated.

Headers:

```
Authorization: 'Bearer $token'
```

`$token` refers to the JWT token

Request:

```
None
```

Success response:

```
{
  message
}
```

## Grading

- **Add a grade to a submission**

*Only an `admin` or a `mentor` can add a grade*

POST `/api/assessment/:aid/:sid/grade`

`:aid` refers to the unique ID of the assessment. `:sid` refers to the unique ID of the submission to be updated.

Headers:

```
Authorization: 'Bearer $token'
```

`$token` refers to the JWT token

Request:

```
{
  marks: required,
  remarks: required
}
```

Success response:

```
{
  message
}
```

- **Update/edit a grade**

*Only an `admin` can edit a grade*

PUT `/api/assessment/:aid/:sid/grade`

`:aid` refers to the unique ID of the assessment. `:sid` refers to the unique ID of the submission to be updated.

Headers:

```
Authorization: 'Bearer $token'
```

`$token` refers to the JWT token

Request:

```
{
  grade: {
    marks: optional,
    remarks: optional
  }
}
```

Success response:

```
{
  message
}
```

- **Delete a grade**

*Only an `admin` can delete a grade*

DELETE `/api/assessment/:aid/:sid/grade`

`:aid` refers to the unique ID of the assessment. `:sid` refers to the unique ID of the submission to be updated.

Headers:

```
Authorization: 'Bearer $token'
```

`$token` refers to the JWT token

Request:

```
None
```

Success response:

```
{
  message
}
```

## Users

- **Fetch logged in user's data**

```
GET /api/user
```

Headers:

```
Authorization: 'Bearer $token'
```

`$token` refers to the JWT token Request:

```
None
```

Success response:

```
{
  message,
  uid,
  email,
  name,
  role,
  joined
}
```

- **Update/edit a user**

*Only an admin can edit user credentials*

```
PUT /api/user/:uid
```

`:uid` refers to the unique ID of the user to be updated.

Headers:

```
Authorization: 'Bearer $token'
```

`$token` refers to the JWT token

Request:

```
{
  updated: {
    email: optional,
    password: optional,
    role: optional,
    name: optional
  }
}
```

Success response:

```
{
  message
}
```

- **Delete a user**

*Only an `admin` can delete a user*

```
DELETE /api/user/:uid
```

`:uid` refers to the unique ID of the user to be updated. Another `admin` cannot be deleted.

Headers:

```
Authorization: 'Bearer $token'
```

`$token` refers to the JWT token

Request:

```
None
```

Success response:

```
{
  message
}
```

## Notes

Response for failed requests:

```
{
  error
}
```

`error` contains necessary error messages

Failed requests due to authentication are sent with a status code of `401`, failed requests due to providing insufficient or corrupt data are sent with a status code of `400`, failed requests due to server error are sent with a status code of `500`.