🤪

Designing a REST API

Filipovych Marharyta

I have chosen the topic **Scheduling and alerting of the deadlines of education activities**.

Functional Requirements

1. The API must support the creation, reviewing, updating, and deleting of courses
2. The API must support the creation, reading, updating, and deleting of assignments within courses
3. The API must support setting and modification of deadlines for assignments
4. The API must support the creation, reading, updating, and deleting of users (students, teachers, administrators)
5. The API must support enrolling/unenrolling students in/from courses
6. The API must support assigning teachers to courses
7. The API must support the creation and management of alerts for upcoming deadlines
8. The API must support tracking assignment submission status

Non-Functional Requirements

1. The API should adhere to security concerns by implementing JWT-based authentication and role-based authorization
2. The API response time should be under 200ms for most operations
3. The API should be scalable: support at least 10000 concurrent users
4. The API should be 24/7 available for usage
5. The API should cache frequently accessed data -> GET requests
6. All list endpoints should support pagination
7. The API should have a rate limit: max 100 requests per minute per user
8. The API should support versioning to maintain backward compatibility

Entity Description

User: represents individuals in the system with different roles (student, teacher, admin).

- id
- first name
- last name
- phone (can be null if not specified)
- email
- password hash
- password salt: salt used in password hashing
- role (student, teacher, admin) → enum
- registered_at: when the user account was created (datetime)

Relationships:

- One-to-many with Enrollment: a student can be enrolled in multiple courses
- One-to-many with Submission: a student can submit multiple assignments
- One-to-many with Alert: a student can receive multiple alerts
- Many-to-many with Course: for teachers who teach multiple courses

Course

- id
- title
- short description
- start date
- end date
- max students: max number of students
- prerequisites: courses that serve as a prerequisite to this particular course; can be null if there are no such courses.
- credits: number of credits
- hours: number of hours required to be spent on this course
- teacher ids: theachers that educate this course

Relationships:

- One-to-many with Assignment: a course can have multiple assignments
- One-to-many with Enrollment: a course can have multiple students enrolled
- Many-to-many with User: a course can be taught by multiple teachers

Assignment

- id
- title
- description
- course id: reference to the Course entity
- deadline: when the assignment is due (datetime)
- points: maximum points possible
- created at: when the assignment was created (datetime)
- obligatory: whether the assignment is required (boolean)

Relationships:

- Many-to-one with Course: an assignment belongs to a course
- One-to-many with Submission: an assignment can have multiple submissions
- One-to-many with Alert: an assignment can have multiple alerts

Enrollment

- id
- user id: reference to the User entity
- course id: reference to the Course entity
- enrolment date
- unenrollment date
- status (active, completed, dropped) → enum

Relationships:

- Many-to-one with User: many enrollments can belong to one student
- Many-to-one with Course: many enrollments can be for one course

Submission

- id
- assignment id: reference to the Assignment entity

- user id: reference to the User entity
- submission date
- status (submitted, graded, overdue) → enum
- grade date: when the submission was graded
- grade (if graded)
- feedback (optional)

Relationships:

- Many-to-one with Assignment: many submissions can be for one assignment
- Many-to-one with User: many submissions can be from one student

Alert

- id
- user id: reference to the User entity
- assignment id: reference to the Assignment entity
- alert type (upcoming deadline, grade posted, grade changed, feedback added, deadline today) → enum
- sent at
- read at

Relationships:

- Many-to-one with User: many alerts can be sent to one user
- Many-to-one with Assignment: many alerts can be about one assignment

REST API Design and Operations

General guidelines:

Base URL: **https://api.student-deadline-sceduling.com** .

Version: v1.

All endpoints are prefixed with **/api/v1**.

All protected endpoints require an Authorization header:

```
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Response Format

All successful responses, except for authorization endpoints, follow this format:

```
{

    "data": {...}, // main response data
    "pagination": {...}, //pagination info
    "links": {...} // HATEOAS links
}
```

The structure of the data field depends on a particular endpoint.

Only list endpoints (collections) support pagination. Pagination format:

```
    "pagination": {
        "total": <total number of items>, //number
        "pages": <total number of pages>,  //number
        "page": <current page>, //number
        "limit": <items per page> //number
    }
```

The links field provides links to related information to adhere to **RMM** model requirements (**HATEOAS**).

If this is a collection endpoint, links to the next (next), previous (prev), first (first) and last (last) pages are provided to preserve pagination. The **self** link is present in both types of endpoints: the collection and the singleton. The self link refers to the current endpoint. If the endpoint refers to the concrete singleton resource (not a collection), additional related fields, such as course, assignment, user, etc., will be included in the link section, but not pagination links.

```
// for collection endpoints (with pagination)
"links": {
        "self": "link to itself",
        "prev": "link to the previous page", //null if there is no previous page
        "next": "link to the next page", //null if there is no next page
        "first": "link to the first page",
        "last": "link to the last page"
    }
// for singleton endpoints (without pagination)
"links": {
        "self": "link to itself",
        ... // links to related fields
    }
```

Error Format

The **error** field contains error details: **code** and **message**.

```
"error": {
    "code": <ERROR_CODE>, // this field value is a number
    "message": "human-readable error message",
}
```

Common Query Parameters

- **page** (number): page number for pagination, default - **1**
- **limit** (number): number of items per page, default - **20**
- **sort** (string): field(s) to sort by. This field sorting options differ depending on the endpoint.

Collection endpoints support **filtering** and **sorting** with **query parameters**.

All GET requests are **cacheable**. The **Cache-Control** header is used for that purpose.

Rate Limiting

- Rate limit: **100** requests per minute per authenticated user
- Rate limit headers included in responses:
    - **X-RateLimit-Limit**: maximum requests per time window (100)
    - **X-RateLimit-Remaining**: remaining requests in current window
    - **X-RateLimit-Reset**: time until window reset (seconds)

Common status codes

Success codes: **200** (OK), **201** (Created), **204** (No content, successful deletion)

Client error: **400** (Bad request), **401** (Unauthorized), **403** (Forbidden), **404** (Not found), **409** (Conflict), **429** (Too many requests)

Server error: **500** (internal server error), **503** (service unavailable)

Caching

All GET requests in this API are cacheable, with the Cache-Control header determining the caching behavior:

- **Public resources** (non-user data): Cache-Control: public, max-age=[seconds]
  - Example: Courses list (300s), course details (600s)
- **Private resources** (user-specific data): Cache-Control: private, max-age=[seconds]
  - Example: User profiles (300s), user alerts (60s)
- **Mutable operations** (POST, PUT, PATCH, DELETE): Cache-Control: no-store
- Frequently changing data: 60-300 seconds
- Stable data: 600-3600 seconds

API Endpoints

Authentication

This API uses **JWT** for authentication. Available roles include **student**, **teacher**, **admin**. **JWT structure:**

```
{
  "header": {
    "alg": "HS256",
    "typ": "JWT"
  },
  "payload": {
    "id": "user_id",
    "first_name": "first_name",
    "last_name": "last_name",
    "phone": "phone",
    "email": "email",
    "role": "student|teacher|admin",
    "iat": 1516239022,
    "exp": 1516242622
  }
}
```

`POST /api/v1/auth/login` *authenticate a user and get access tokens*

Request body parameters:

Parameter	Type	Required
email	string	Yes (if no phone)
phone	string	Yes (if no email)
password	string	Yes

Status codes: 201, 400 (invalid format), **401** (invalid credentials), **429**

Caching: Cache-Control: no-store

Example request:

```
POST /api/v1/auth/login HTTP/1.1
Host: api.student-deadline-sceduling.com
Content-Type: application/json

{
  "email": "john.doe@example.com",
  "password": "securePassword123"
}
```

Example response:

```
HTTP/1.1 201 Created
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9",
  "refresh_token" : "hjghjghjgjh",
  "user": {
    "id": "123",
    "role": "student"
  }
}
```

`POST /api/v1/auth/logout:` *end the user's session*

Request body parameters: none required

Status Codes: 201, 401, 429

Caching: Cache-Control: no-store

Example request:

```
POST /api/v1/auth/logout HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Example response:

```
HTTP/1.1 201 Created
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "message": "Successfully logged out"
}
```

`POST /api/v1/auth/refresh` *get a new access token using a refresh token.*

Request body parameters:

Parameter	Type	Required
refresh_token	string	Yes

Status Codes: 201, 400 (invalid format),**401** (invalid refresh token), **429**

Caching: Cache-Control: no-store

Example request:

```
POST /api/v1/auth/refresh HTTP/1.1
Host: api.student-deadline-sceduling.com
Content-Type: application/json

{
  "refresh_token": "hjghjghjgjh"
}
```

Example response:

```
HTTP/1.1 201 Created
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "refresh_token": "jhkuhkhjkhkj"
}
```

`POST /api/v1/auth/forgot` *request a password reset code.*

Request body parameters:

Parameter	Type	Required
email	string	Yes (if no phone)
phone	string	Yes (if no email)

Status Codes: 201, 404 (user not found), **429, 400** (invalid format)

Caching: Cache-Control: no-store

Example request:

```
POST /api/v1/auth/forgot HTTP/1.1
Host: api.student-deadline-sceduling.com
Content-Type: application/json

{
  "email": "john.doe@example.com"
}
```

Example response:

```
HTTP/1.1 201 Created
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "reset_code": "jhjkhkhk",
  "exp": "2023-04-12T15:30:00Z"
}
```

`POST /api/v1/auth/reset` *reset a user's password using a reset code.*

Request body parameters:

Parameter	Type	Required
email	string	Yes (if no phone)
phone	string	Yes (if no email)

reset_code	string	Yes
new_password	string	Yes

Status Codes: 201, 401 (invalid reset code), **429, 400** (invalid format), **404** (user not found)

Caching: Cache-Control: no-store

Example request:

```
POST /api/v1/auth/reset HTTP/1.1
Host: api.student-deadline-sceduling.com
Content-Type: application/json

{
  "email": "john.doe@example.com",
  "reset_code": "jhjkhkhk",
  "new_password": "newSecurePassword123"
}
```

Example response:

```
HTTP/1.1 201 Created
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "message": "Password successfully reset"
}
```

`POST /api/v1/auth/register` *create a new user account in the system*

Request body parameters:

Parameter	Type	Required
email	string	Yes (if no phone)
phone	string	Yes (if no email)
password	string	Yes
first_name	string	Yes
last_name	string	Yes

Status codes: 201, 400 (validation error),**429**, **409** (email/phone already in use)

Caching: Cache-Control: no-store

Example request:

```
POST /api/v1/auth/register HTTP/1.1
Host: api.student-deadline-sceduling.com
Content-Type: application/json

{
  "email": "john.doe@example.com",
  "phone": "+380961234567",
  "password": "securePassword123",
  "first_name": "John",
  "last_name": "Doe"
}
```

Example response:

```
HTTP/1.1 201 Created
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "data": {
    "id": "123",
    "first_name": "John",
    "last_name": "Doe",
    "email": "john.doe@example.com",
    "phone": "+380961234567",
    "role": "student",
    "registered_at": "2023-04-12T14:30:00Z"
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/users/123"
  }
}
```

Users

`GET /api/v1/users` *retrieve a list of all users (admin only)*

Query parameters:

Parameter	Type	Required
email	string	No
phone	string	No
sort	string	No
first-name	string	No
last-name	string	No
page	number	No
limit	number	No
role	string	No

Request body parameters: none required

Status codes: 200, 401, 403 (non-admin users), **429**

Caching: Cache-Control: private, max-age=60

Example request:

```
GET /api/v1/users?page=1&limit=10&role=student HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: private, max-age=60
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
```

```
X-RateLimit-Reset: [seconds until reset]

{
  "data": [
    {
      "id": "123",
      "first_name": "John",
      "last_name": "Doe",
      "email": "john.doe@example.com",
      "phone": "+380961234567",
      "role": "student",
      "registered_at": "2023-04-12T14:30:00Z"
    },
    {
      "id": "124",
      "first_name": "Jane",
      "last_name": "Smith",
      "email": "jane.smith@example.com",
      "phone": "+380967654321",
      "role": "student",
      "registered_at": "2023-04-10T11:20:00Z"
    }
  ],
  "pagination": {
      "total": 45,
      "pages": 5,
      "page": 1,
      "limit": 10
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/users?page=1&limit=10&role=student",
    "prev": null,
    "next": "https://api.student-deadline-sceduling.com/api/v1/users?page=2&limit=10&role=student",
      "last": "https://api.student-deadline-sceduling.com/api/v1/users?page=5&limit=10&role=student",
      "first": "https://api.student-deadline-sceduling.com/api/v1/users?page=1&limit=10&role=student",
  }
}
```

`GET /api/v1/users/{id}` *retrieve the specific user's details*

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	User id

Request body parameters: none required

Status codes: 200, 401, 403 (not own profile or not admin), **404** (user not found), **429**

Caching: Cache-Control: private, max-age=300

Example request:

```
GET /api/v1/users/123 HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: private, max-age=300
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]

{
  "data": {
    "id": "123",
    "first_name": "John",
    "last_name": "Doe",
    "email": "john.doe@example.com",
    "phone": "+380961234567",
    "role": "student",
    "registered_at": "2023-04-12T14:30:00Z"
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/users/123",
    "courses": "https://api.student-deadline-sceduling.com/api/v1/users/123/courses",
    "assignments": "https://api.student-deadline-sceduling.com/api/v1/users/123/assignments",
    "alerts": "https://api.student-deadline-sceduling.com/api/v1/users/123/alerts"
  }
}
```

`PUT /api/v1/users/{id}` *update all user fields (admin only)*

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	User id

Request body parameters:

Parameter	Type	Required
first_name	string	Yes
last_name	string	Yes
email	string	Yes
phone	string	No
role	string (student, teacher, admin)	Yes

Status codes: 200, 400 (invalid data), **401, 403** (not admin), **404** (user not found), **409** (email/phone already in use), **429**

Cache: Cache-Control: no-store

Example request:

```
PUT /api/v1/users/123 HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "first_name": "Johnny",
  "last_name": "Doe",
  "phone": "+380961234567",
  "email": "example@gmail.com"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "data": {
    "id": "123",
    "first_name": "Johnny",
    "last_name": "Doe",
    "email": "example@gmail.com",
    "phone": "+380961234567",
    "role": "student",
    "registered_at": "2023-04-12T14:30:00Z"
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/users/123"
  }
}
```

`PATCH /api/v1/users/{id}` *update some concrete user fields (admin only)*

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	User id

Request body parameters:

Parameter	Type	Required
first_name	string	No
last_name	string	No
email	string	No
phone	string	No
role	string (student, teacher, admin)	No

All body parameters are marked as optional in the table, however one of them must be provided in any way.

Status codes: 200, 400 (invalid data), **401, 403** (not admin), **404** (user not found), **409** (email/phone already in use), **429**

Cache: Cache-Control: no-store

Example request:

```
PATCH /api/v1/users/123 HTTP/1.1
Host: api.eduscheduler.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "first_name": "Johnny",
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "data": {
   "id": "123",
    "first_name": "Johnny",
    "last_name": "Doe",
    "email": "john.doe@example.com",
    "phone": "+380961234567",
    "role": "student",
    "registered_at": "2023-04-12T14:30:00Z"
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/users/123"
  }
}
```

Courses

`GET /api/v1/courses` *list all courses*

Query parameters:

Parameter	Type	Required	Description
title	string	No	Filter by title (partial match)
sort	string	No	**Fields** you can sort by: start-desc, start-asc, end-desc, end-asc, credits-asc, credits-desc, hours-asc, hours-desc, title-asc, title-desc (choose one of these options)
page	number	No	page number
limit	number	No	how many items to show
start--after	date	No	Filter by start date after
start-before	date	No	Filter by start date before
end-after	date	No	Filter by end date after
end-before	date	No	Filter by end date before
credits-min	number	No	Filter by minimum credits
credits-max	number	No	Filter by maximum credits
hours-min	number	No	Filter by minimum hours
hours-max	number	No	Filter by maximum hours
credits	number	No	Amount of credits
hours	number	No	Amount of hours
start	date	No	Filter by start date
end	date	No	Filter by end date

Request body parameters: none required

Status codes: 200, 429

Caching: Cache-Control: public, max-age=300

Example request:

```
GET /api/v1/courses?page=1&limit=10&credits-min=3 HTTP/1.1
Host: api.student-deadline-sceduling.com
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: public, max-age=300
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]

{
  "data": [
    {
      "id": "101",
      "title": "Introduction to Mathematics",
      "short_description": "Basic mathematical concepts",
      "start_date": "2023-09-01T00:00:00Z",
      "end_date": "2023-12-15T00:00:00Z",
      "max_students": 30,
      "credits": 3,
      "hours": 45,
    },
    {
      "id": "102",
      "title": "Advanced Physics",
      "short_description": "In-depth study of mechanics and thermodynamics",
      "start_date": "2023-09-01T00:00:00Z",
      "end_date": "2023-12-15T00:00:00Z",
      "max_students": 25,
      "credits": 4,
      "hours": 60,
    }
  ],
  "pagination": {
    "total": 15,
    "pages": 2,
    "page": 1,
    "limit": 10
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/courses?page=1&limit=10&credits-min=3",
    "next": "https://api.student-deadline-sceduling.com/api/v1/courses?page=2&limit=10&credits-min=3",
    "prev": null,
    "first": "https://api.student-deadline-sceduling.com/api/v1/courses?page=1&limit=10&credits-min=3",
    "last": "https://api.student-deadline-sceduling.com/api/v1/courses?page=2&limit=10&credits-min=3",
  }
}
```

`GET /api/v1/courses/{id}` *retrieve details of the specific course*

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	Course id

Request body parameters: none required

Status codes: 200, 404 (course not found), **429**

**Caching: **Cache-Control: public, max-age=600

Example request:

```
GET /api/v1/courses/101 HTTP/1.1
Host: api.student-deadline-sceduling.com
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: public, max-age=600
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]

{
  "data": {
    "id": "101",
    "title": "Introduction to Mathematics",
    "short_description": "Basic mathematical concepts",
    "start_date": "2023-09-01T00:00:00Z",
    "end_date": "2023-12-15T00:00:00Z",
    "max_students": 30,
    "credits": 3,
    "hours": 45,
    "current_enrollment": 25
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/courses/101",
    "assignments": "https://api.student-deadline-sceduling.com/api/v1/courses/101/assignments",
    "enrollments": "https://api.student-deadline-sceduling.com/api/v1/courses/101/enrollments",
    "teachers": "https://api.student-deadline-sceduling.com/api/v1/courses/101/teachers",
    "prerequisites": "https://api.student-deadline-sceduling.com/api/v1/courses/101/prerequisites"
  }
}
```

`POST /api/v1/courses` *create a new course (teachers and admins only)*

Request body parameters:

Parameter	Type	Required
title	string	Yes
short_description	string	Yes
start_date	date	Yes
end_date	date	Yes
max_students	number	Yes
credits	number	Yes
hours	number	Yes

Status codes: 201, 400 (invalid data), **401, 403** (not teacher or admin), **429**

**Caching: **Cache-Control: no-store

Example request:

```
POST /api/v1/courses HTTP/1.1
Host: api.student-deadline-sceduling.com
```

```
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "title": "Introduction to Computer Science",
  "short_description": "Fundamentals of programming and computer systems",
  "start_date": "2023-09-15",
  "end_date": "2024-01-15",
  "max_students": 40,
  "credits": 4,
  "hours": 60
}
```

Example Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "data": {
    "id": "103",
    "title": "Introduction to Computer Science",
    "short_description": "Fundamentals of programming and computer systems",
    "start_date": "2023-09-15T00:00:00Z",
    "end_date": "2024-01-15T00:00:00Z",
    "max_students": 40,
    "credits": 4,
    "hours": 60,
    "current_enrollment": 0
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/courses/103",
    "assignments": "https://api.student-deadline-sceduling.com/api/v1/courses/103/assignments",
    "enrollments": "https://api.student-deadline-sceduling.com/api/v1/courses/103/enrollments",
    "teachers": "https://api.student-deadline-sceduling.com/api/v1/courses/103/teachers",
    "prerequisites": "https://api.student-deadline-sceduling.com/api/v1/courses/103/prerequisites"

  }
}
```

`PUT /api/v1/courses/{id}` *update the information about the specific existing course (teachers and admins only)*

Request body parameters:

Parameter	Type	Required
title	string	Yes
short_description	string	Yes
start_date	date	Yes
end_date	date	Yes
max_students	number	Yes
credits	number	Yes
hours	number	Yes

Status codes: 200, 400, 401, 403 (not teacher or admin), **404, 429** (update conflicts with the resource's current state or this updates creates a conflict with some other courses))

Caching: Cache-Control: no-store

Example request:

```
PUT /api/v1/courses/101 HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "title": "Advanced Mathematics",
  "short_description": "Comprehensive study of advanced mathematical concepts",
  "start_date": "2023-09-01",
  "end_date": "2023-12-15",
  "max_students": 35,
  "credits": 4,
  "hours": 60
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "data": {
    "id": "101",
    "title": "Advanced Mathematics",
    "short_description": "Comprehensive study of advanced mathematical concepts",
    "start_date": "2023-09-01T00:00:00Z",
    "end_date": "2023-12-15T00:00:00Z",
    "max_students": 35,
    "credits": 4,
    "hours": 60,
    "current_enrollment": 25
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/courses/101",
    "assignments": "https://api.student-deadline-sceduling.com/api/v1/courses/101/assignments",
    "enrollments": "https://api.student-deadline-sceduling.com/api/v1/courses/101/enrollments",
    "teachers": "https://api.student-deadline-sceduling.com/api/v1/courses/101/teachers",
    "prerequisites": "https://api.student-deadline-sceduling.com/api/v1/courses/101/prerequisites"
  }
}
```

`PATCH /api/v1/courses/{id}` *update the some specific information about the specific existing course (teachers and admins only)*

Request body parameters:

Parameter	Type	Required
title	string	No

short_description	string	No
start_date	date	No
end_date	date	No
max_students	number	No
credits	number	No
hours	number	No

One of the parameters must be present.

Status codes: 200, 400, 401, 403 (not teacher or admin)**, 404, 429, 409** (update conflicts with the resource's current state or this updates creates a conflict with some other courses)

Caching: Cache-Control: no-store

Example request:

```
PATCH /api/v1/courses/101 HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "title": "Mathematics"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "data": {
    "id": "101",
    "title": "Mathematics",
    "short_description": "Comprehensive study of advanced mathematical concepts",
    "start_date": "2023-09-01T00:00:00Z",
    "end_date": "2023-12-15T00:00:00Z",
    "max_students": 35,
    "credits": 4,
    "hours": 60,
    "current_enrollment": 25
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/courses/101",
    "assignments": "https://api.student-deadline-sceduling.com/api/v1/courses/101/assignments",
    "enrollments": "https://api.student-deadline-sceduling.com/api/v1/courses/101/enrollments",
    "teachers": "https://api.student-deadline-sceduling.com/api/v1/courses/101/teachers",
    "prerequisites": "https://api.student-deadline-sceduling.com/api/v1/courses/101/prerequisites"
  }
}
```

`DELETE /api/v1/courses/{id}` *delete a specific course from the system (teachers and admins only)*

Request body parameters: none required

Status codes: **204**, **401**, **403** (not teacher or admin; course cannot be deleted due to existing dependencies), **404** (course not found), **429**

Caching: Cache-Control: no-store

Example request:

```
DELETE /api/v1/courses/101 HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Example response:

```
HTTP/1.1 204 No Content
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store
```

Assignment

`GET /api/v1/courses/{course-id}/assignments` *retrieve assignments for the specific course*

Path parameters:

Parameter	Type	Required	Description
course-id	string	Yes	Course id

Query parameters:

Parameter	Type	Required	Description
page	number	No	page number
limit	number	No	how many items to show
deadline-after	datetime	No	Filter by deadline after
deadline-before	datetime	No	Filter by deadline before
deadline	datetime	No	assignment deadline
obligatory	string	No	Filter by obligatory status
sort	string	No	Fields you can sort by: deadline-asc, deadline-desc, title-asc, title-desc, points-asc, points-desc. You may choose one of these options.
title	string	No	assignment title
points	number	No	how many points you can get

Request body parameters: none required

Status codes: 200, 404 (course not found), **429**

Caching: Cache-Control: public, max-age=300

Example request:

```
GET /api/v1/courses/101/assignments HTTP/1.1
Host: api.student-deadline-sceduling.com
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: public, max-age=300
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]

{
  "data": [
    {
      "id": "201",
      "title": "Calculus Homework 1",
      "description": "Problems 1-10 from Chapter 3",
      "course_id": "101",
      "deadline": "2023-09-15T23:59:59Z",
      "points": 100,
      "obligatory": true,
      "created_at": "2023-08-01T10:00:00Z"
    },
    {
      "id": "202",
      "title": "Linear Algebra Quiz",
      "description": "Online quiz covering vectors and matrices",
      "course_id": "101",
      "deadline": "2023-10-10T23:59:59Z",
      "points": 75,
      "obligatory": true,
      "created_at": "2023-08-15T14:30:00Z"
    }
  ],
  "meta": {
    "pagination": {
      "total": 8,
      "pages": 4,
      "page": 1,
      "limit": 2
    }
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/courses/101/assignments?page=1&limit=2",
    "prev": null,
    "next": "https://api.student-deadline-sceduling.com/api/v1/courses/101/assignments?page=2&limit=2",
    "last": "https://api.student-deadline-sceduling.com/api/v1/courses/101/assignments?page=4&limit=2",
    "first": "https://api.student-deadline-sceduling.com/api/v1/courses/101/assignments?page=1&limit=2",
    "student": "https://api.student-deadline-sceduling.com/api/v1/users/101"
  }
}
```

`POST /api/v1/courses/{course-id}/assignments` *create a new assignment for a course (teachers and admins only).*

Path parameters:

Parameter	Type	Required	Description
course-id	string	Yes	Course id

Request body parameters:

Parameter	Type	Required
title	string	Yes
description	string	Yes
deadline	datetime	Yes
points	number	Yes
obligatory	boolean	Yes

Status codes: 201, 400 (invalid data), **401, 403** (not course teacher or admin), **404** (course not found), **429**

Caching: Cache-Control: no-store

Example request:

```
POST /api/v1/courses/101/assignments HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "title": "Final Exam",
  "description": "Comprehensive exam covering all course material",
  "deadline": "2023-12-10T23:59:59Z",
  "points": 200,
  "obligatory": true
}
```

Example response:

```
HTTP/1.1 201 Created
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "data": {
    "id": "205",
    "title": "Final Exam",
    "description": "Comprehensive exam covering all course material",
    "course_id": "101",
    "deadline": "2023-12-10T23:59:59Z",
    "points": 200,
    "obligatory": true,
    "created_at": "2023-09-01T09:15:00Z"
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/assignments/205",
    "course": "https://api.student-deadline-sceduling.com/api/v1/courses/101",
    "submissions": "https://api.student-deadline-sceduling.com/api/v1/assignments/205/submissions"
  }
}
```

`GET /api/v1/assignments/{id}` *retrieve details of the specific assignment*

Path parameters:

Parameter	Type	Required	Description
course-id	string	Yes	Course id

Request body parameters: none required

Status codes: 200, 404 (assignment not found), **429**

Caching: Cache-Control: public, max-age=600

Example request:

```
GET /api/v1/assignments/201 HTTP/1.1
Host: api.student-deadline-sceduling.com
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: public, max-age=600
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]

{
  "data": {
    "id": "201",
    "title": "Calculus Homework 1",
    "description": "Problems 1-10 from Chapter 3",
    "course_id": "101",
    "deadline": "2023-09-15T23:59:59Z",
    "points": 100,
    "obligatory": true,
    "created_at": "2023-08-01T10:00:00Z"
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/assignments/201",
    "course": "https://api.student-deadline-sceduling.com/api/v1/courses/101",
    "submissions": "https://api.student-deadline-sceduling.com/api/v1/assignments/201/submissions"
  }
}
```

`PUT /api/v1/assignments/{id}` ***update all assignment fields (teachers and admins only)***

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	Assignment id

Request body parameters:

Parameter	Type	Required
title	string	Yes
description	string	Yes
deadline	datetime	Yes
points	number	Yes
obligatory	boolean	Yes

Status codes: 200, **400** (invalid data), **401**, **403** (not course teacher or admin), **404** (assignment not found), **409** (conflict with existing data), **429**

Caching: Cache-Control: no-store

Example request:

```
PUT /api/v1/assignments/201 HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "title": "Calculus Homework 1 - Revised",
  "description": "Problems 1-15 from Chapter 3",
  "deadline": "2023-09-20T23:59:59Z",
  "points": 120,
  "obligatory": true
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "data": {
    "id": "201",
    "title": "Calculus Homework 1 - Revised",
    "description": "Problems 1-15 from Chapter 3",
    "course_id": "101",
    "deadline": "2023-09-20T23:59:59Z",
    "points": 120,
    "obligatory": true,
    "created_at": "2023-08-01T10:00:00Z",
    "updated_at": "2023-09-05T14:25:00Z"
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/assignments/201",
    "course": "https://api.student-deadline-sceduling.com/api/v1/courses/101",
    "submissions": "https://api.student-deadline-sceduling.com/api/v1/assignments/201/submissions"
  }
}
```

`PATCH /api/v1/assignments/{id}` *partially update assignment fields (teachers and admins only)*

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	Assignment id

Request body parameters:

Parameter	Type	Required
title	string	No
description	string	No
deadline	datetime	No
points	number	No
obligatory	boolean	No

Status codes: 200, 400 (invalid data), **401, 403** (not course teacher or admin), **404** (assignment not found), **409** (conflict with existing data), **429**

Caching: Cache-Control: no-store

Example request:

```
PATCH /api/v1/assignments/201 HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "deadline": "2023-09-22T23:59:59Z",
  "points": 110
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "data": {
    "id": "201",
    "title": "Calculus Homework 1",
    "description": "Problems 1-10 from Chapter 3",
    "course_id": "101",
    "deadline": "2023-09-22T23:59:59Z",
    "points": 110,
    "obligatory": true,
    "created_at": "2023-08-01T10:00:00Z",
    "updated_at": "2023-09-05T15:10:00Z"
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/assignments/201",
    "course": "https://api.student-deadline-sceduling.com/api/v1/courses/101",
    "submissions": "https://api.student-deadline-sceduling.com/api/v1/assignments/201/submissions"
  }
}
```

`DELETE /api/v1/assignments/{id}` *delete an assignment (teachers and admins only)*

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	Assignment id

Request body parameters: none required

Status codes: 204, **401**, **403** (not course teacher or admin; cannot be deleted due to existing dependencies), **404** (assignment doesn't exist), **429**

Caching: Cache-Control: no-store

Example request:

```
DELETE /api/v1/assignments/201 HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Example response:

```
HTTP/1.1 204 No Content
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store
```

Submission

`GET /api/v1/assignments/{assignment-id}/submissions` ***retrieve submissions for the specific assignment (teachers, admins, or own submissions only).***

Path parameters:

Parameter	Type	Required	Description
assignment-id	string	Yes	Assignment id

Query parameters:

Parameter	Type	Required	Description
limit	number	No	how many items to show
page	number	No	page number
status	string	No	submitted, graded, overdue
submission-after	datetime	No	Filter by submission date after
submission-before	datetime	No	Filter by submission date before
submission-date	datetime	No	date of submission
grade-min	number	No	Filter by minimum grade
grade-max	number	No	Filter by maximum grade
grade	number	No	grade (null if not yet graded)
sort		No	Fields you can sort by: submission-asc, submission-desc, grade-asc, grade-desc. You may choose one of these options.

Request body parameters: none required

Status codes: 200, 401, 403 (not course teacher, admin, or own submissions), **404** (assignment not found), **429**

Caching: Cache-Control: private, max-age=300

Example request:

```
GET /api/v1/assignments/201/submissions HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Example Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "data": [
    {
      "id": "401",
      "assignment_id": "201",
      "user_id": "123",
      "submission_date": "2023-09-14T20:45:00Z",
      "status": "graded",
      "grade": 90,
      "feedback": "Excellent work, but missed one key concept."
    },
    {
      "id": "402",
      "assignment_id": "201",
      "user_id": "124",
      "submission_date": "2023-09-14T22:30:00Z",
      "status": "graded",
      "grade": 85,
      "feedback": "Good work, but missed a few concepts."
    }
  ],
  "meta": {
    "pagination": {
      "total": 25,
      "pages": 13,
      "page": 1,
      "limit": 2
    }
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/assignments/201/submissions?page=1&limit=2",
    "next": "https://api.student-deadline-sceduling.com/api/v1/assignments/201/submissions?page=2&limit=2",
    "prev": null,
    "last": "https://api.student-deadline-sceduling.com/api/v1/assignments/201/submissions?page=13&limit=2",
      "first": "https://api.student-deadline-sceduling.com/api/v1/assignments/201/submissions?page=1&limit=2"
  }
}
```

`POST /api/v1/assignments/{assignment-id}/submissions` *submit an assignment (students only)*

Path parameters:

Parameter	Type	Required	Description
assignment-id	string	Yes	Assignment id

Request body parameters:

Parameter	Type	Required	Description
content	string	Yes	Submission content or file reference

Status codes: 201, 400 (invalid data), **401, 403** (not a student or not enrolled in course), **404** (assignment not found), **409** (already submitted), **429**

Caching: Cache-Control: no-store

Example request:

```
POST /api/v1/assignments/201/submissions HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "content": "My solution to the calculus problems can be found at this link..."
}
```

Example response:

```
HTTP/1.1 201 Created
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "data": {
    "id": "405",
    "assignment_id": "201",
    "user_id": "123",
    "submission_date": "2023-09-15T10:20:00Z",
    "status": "submitted",
    "content": "My solution to the calculus problems can be found at this link..."
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/submissions/405",
    "assignment": "https://api.student-deadline-sceduling.com/api/v1/assignments/201",
    "user": "https://api.student-deadline-sceduling.com/api/v1/users/123"
  }
}
```

`GET /api/v1/submissions/{id}` ***retrieve details of the specific submission.***

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	Submission id

Request body parameters: none required

Status codes: 200, 401, 403 (not submission owner, course teacher, or admin), **404** (submission not found), **429**

Caching: Cache-Control: private, max-age=600

Example request:

```
GET /api/v1/submissions/401 HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: private, max-age=600
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]

{
  "data": {
    "id": "401",
    "assignment_id": "201",
    "user_id": "123",
    "submission_date": "2023-09-14T20:45:00Z",
    "status": "graded",
    "grade": 90,
    "grade_date": "2023-09-15T10:30:00Z",
    "feedback": "Excellent work, but missed one key concept.",
    "content": "My solution to the calculus problems..."
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/submissions/401",
    "assignment": "https://api.student-deadline-sceduling.com/api/v1/assignments/201",
    "user": "https://api.student-deadline-sceduling.com/api/v1/users/123"
  }
}
```

`PUT /api/v1/submissions/{id}/grade` ***grade a submission (teachers within the specific course only)***

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	Submission id

Request body parameters:

Parameter	Type	Required
grade	number	Yes
feedback	string	No

Status codes: 200, 401, 400 (invalid data), **403** (not course teacher), **404** (submission not found), **429**

Caching: Cache-Control: no-store

Example request:

```
PUT /api/v1/submissions/405/grade HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "grade": 95,
  "feedback": "Excellent work! Very thorough solutions."
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "data": {
    "id": "405",
    "assignment_id": "201",
    "user_id": "123",
    "submission_date": "2023-09-15T10:20:00Z",
    "status": "graded",
    "grade": 95,
    "grade_date": "2023-09-16T14:30:00Z",
    "feedback": "Excellent work! Very thorough solutions.",
    "content": "My solution to the calculus problems can be found at this link..."
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/submissions/405",
    "assignment": "https://api.student-deadline-sceduling.com/api/v1/assignments/201",
    "user": "https://api.student-deadline-sceduling.com/api/v1/users/123"
  }
}
```

`PATCH /api/v1/submissions/{id}/` *modify the already graded submission (teachers within the specific course only): change grade or add feedback*

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	Submission id

Request body parameters:

Parameter	Type	Required
grade	number	No
feedback	string	No

One of the request body parameters must be provided.

Status codes: 200, 401, 400 (invalid data), **403** (not course teacher), **404** (submission not found), **429**

Caching: Cache-Control: no-store

Example request:

```
PATCH /api/v1/submissions/405/grade HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "grade": 100
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "data": {
    "id": "405",
    "assignment_id": "201",
    "user_id": "123",
    "submission_date": "2023-09-15T10:20:00Z",
    "status": "graded",
    "grade": 100,
    "grade_date": "2023-09-16T14:30:00Z",
    "feedback": "Excellent work! Very thorough solutions.",
    "content": "My solution to the calculus problems can be found at this link..."
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/submissions/405",
    "assignment": "https://api.student-deadline-sceduling.com/api/v1/assignments/201",
    "user": "https://api.student-deadline-sceduling.com/api/v1/users/123"
  }
}
```

Enrollment

`GET /api/v1/courses/{course-id}/enrollments` *retrieve enrollments for the specific course (teachers and admins only)*

Path parameters:

Parameter	Type	Required	Description
course-id	String	Yes	Course id

Query parameters:

Parameter	Type	Required	Description
page	number	No	page number
limit	number	No	how many items to show
status		No	active, completed, dropped
enrollment-after	date	No	Filter by enrollment date after
enrollment-before	date	No	Filter by enrollment date before
enrollment-date	date	No	when the enrollment ocurred
sort	string	No	Fields you can sort by: enrollment-asc, enrollment-desc, unenrollment-asc, unenrollment-desc. You may choose one of these options.
unenrollment-after	date	No	Filter by unenrollment date after
unenrollment-before	date	No	Filter by unenrollment date before
unenrollment-date	date	No	when the enrollment ocurred (this field will have the value of null if the student is still enrolled)

Request body parameters: none required

Status codes: 200, 401, 400 (invalid data), **403** (not course teacher or admin), **404** (course not found), **429**

Caching: Cache-Control: private, max-age=300

Example request:

```
GET /api/v1/courses/101/enrollments HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: private, max-age=300
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]

{
  "data": [
    {
      "id": "301",
      "user_id": "124",
      "course_id": "101",
      "enrollment_date": "2023-09-01T10:15:00Z",
      "status": "active",
      "user": {
        "id": "124",
        "first_name": "Jane",
        "last_name": "Smith"
      }
    },
    {
      "id": "302",
      "user_id": "125",
      "course_id": "101",
      "enrollment_date": "2023-08-28T14:30:00Z",
      "status": "active",
      "user": {
        "id": "125",
        "first_name": "Robert",
        "last_name": "Johnson"
      }
    }
  ],
  "meta": {
    "pagination": {
      "total": 28,
      "pages": 14,
      "page": 1,
      "limit": 2
    }
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/courses/101/enrollments?page=1&limit=2",
    "next": "https://api.student-deadline-sceduling.com/api/v1/courses/101/enrollments?page=2&limit=2",
      "prev": null,
    "first":  "https://api.student-deadline-sceduling.com/api/v1/courses/101/enrollments?page=1&limit=2",
    "last": "https://api.student-deadline-sceduling.com/api/v1/courses/101/enrollments?page=14&limit=2"
  }
}
```

`POST /api/v1/courses/{course-id}/enrollments` ***enroll a student in a course (self-enrollment, teachers, or admins).***

Path parameters:

Parameter	Type	Required	Description
course-id	string	Yes	Course id

Request body parameters:

Parameter	Type	Required	Description
user_id	string	Yes	User id of student to enroll

Status codes: 201, 401, 400 (invalid data), **403** (not allowed to enroll this user), **404** (course or user not found), **409** (already enrolled or course full), **429**

Caching: Cache-Control: no-store

Example request:

```
POST /api/v1/courses/101/enrollments HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "user_id": "124"
}
```

Example response:

```
HTTP/1.1 201 Created
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
 "data": {
  "id": "301",
  "user_id": "124",
  "course_id": "101",
  "enrollment_date": "2023-09-01T10:15:00Z",
  "status": "active"
 },
 "links": {
  "self": "https://api.student-deadline-sceduling.com/api/v1/enrollments/301",
  "user": "https://api.student-deadline-sceduling.com/api/v1/users/124",
  "course": "https://api.student-deadline-sceduling.com/api/v1/courses/101"
 }
}
```

`GET /api/v1/enrollments/{id}` ***retrieve details of the specific enrollment.***

Path parameters:

Parameter	type	Required	Description
id	string	Yes	Enrollment id

Request body parameters: none required

Status codes: 200, 401, 403 (not course teacher, admin, or enrolled student), **404** (enrollment not found), **429**

Caching: Cache-Control: private, max-age=600

Example request:

```
GET /api/v1/enrollments/301 HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: private, max-age=600
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]

{
 "data": {
  "id": "301",
  "user_id": "124",
  "course_id": "101",
  "enrollment_date": "2023-09-01T10:15:00Z",
  "status": "active",
  "user": {
   "id": "124",
   "first_name": "Jane",
      "last_name": "Smith",
      "email": "jane.smith@example.com"
    },
    "course": {
      "id": "101",
      "title": "Introduction to Mathematics"
    }
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/enrollments/301",
    "user": "https://api.student-deadline-sceduling.com/api/v1/users/124",
    "course": "https://api.student-deadline-sceduling.com/api/v1/courses/101"
  }
}
```

`PATCH /api/v1/enrollments/{id}` *update the status of the enrollment (teachers and admins only)*

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	Enrollment id

Request body parameters:

Parameter	Type	Required	Description
status	string	Yes	New status (active, completed, dropped)

Status codes: 200, 401, 403 (not course teacher or admin), **404** (enrollment not found), **429, 400**

Caching: Cache-Control: no-store

Example request:

```
PATCH /api/v1/enrollments/301 HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "status": "completed"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "data": {
    "id": "301",
    "user_id": "124",
    "course_id": "101",
    "enrollment_date": "2023-09-01T10:15:00Z",
    "status": "completed"
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/enrollments/301",
    "user": "https://api.student-deadline-sceduling.com/api/v1/users/124",
    "course": "https://api.student-deadline-sceduling.com/api/v1/courses/101"
  }
}
```

`DELETE /api/v1/enrollments/{id}` *unenroll a student from the course (self-unenrollment, teachers, or admins)*

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	Enrollment id

Request body parameters: none required

Status codes: 204, 401, 403 (not course teacher, admin, or enrolled student), **404** (enrollment not found), **429**

Caching: Cache-Control: no-store

Example request:

```
DELETE /api/v1/enrollments/301 HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Example response:

```
HTTP/1.1 204 No Content
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store
```

Alert

`POST /api/v1/alerts` *create a new alert (teachers and admins only)*

Request body parameters:

Parameter	Type	Required
user_id	string	Yes
alert_type	string	Yes

Status codes: 201, 401, 403 (not admin or teacher), **404** (user or assignment not found), **429, 409, 400**

Caching: Cache-Control: no-store

Example request:

```
POST /api/v1/alerts HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "user_id": "123",
  "assignment_id": "201",
  "alert_type": "upcoming_deadline",
  "message": "Reminder: Calculus Homework 1 is due in 24 hours"
}
```

Example response:

```
HTTP/1.1 201 Created
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "data": {
    "id": "501",
    "user_id": "123",
    "assignment_id": "201",
    "alert_type": "upcoming_deadline",
    "message": "Reminder: Calculus Homework 1 is due in 24 hours",
    "sent_at": null,
    "read_at": null
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/alerts/501",
    "user": "https://api.student-deadline-sceduling.com/api/v1/users/123",
    "assignment": "https://api.student-deadline-sceduling.com/api/v1/assignments/201"
  }
}
```

`POST /api/v1/alerts/{id}/send` *send an existing alert (teachers and admins only)*

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	Alert id

Request body parameters: none required

Status codes: 201, 401, 403 (not admin or teacher; already sent), **404** (alert not found), **429, 409**

Caching: Cache-Control: no-store

Example request:

```
POST /api/v1/alerts/501/send HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "data": {
    "id": "501",
    "user_id": "123",
    "assignment_id": "201",
    "alert_type": "upcoming_deadline",
    "message": "Reminder: Calculus Homework 1 is due in 24 hours",
    "sent_at": "2023-09-14T08:30:00Z",
  "read_at": null
 },
 "links": {
  "self": "https://api.student-deadline-sceduling.com/api/v1/alerts/501",
  "user": "https://api.student-deadline-sceduling.com/api/v1/users/123",
  "assignment": "https://api.student-deadline-sceduling.com/api/v1/assignments/201"
 }
}
```

`GET /api/v1/users/{user-id}/alerts` *retrieve alerts for the specific user*

Path parameters:

Parameter	Type	Required	Description
user-id	string	Yes	User id

Query parameters:

Parameter	Type	Required	Description
sort	string	No	You can sort by one of these fields: **send-desc, sent-asc, read-asc, read-desc**.
assignment-id	string	No	Filter by assignment id
page	number	No	page number
limit	number	No	how many items to show
sent-after	datetime	No	Filter by sent date after
sent-before	datetime	No	Filter by sent date before
sent	datetime	No	when the alert was sent
read at	datetime	No	when the alert was read (this field has the value of null if the alert is untouched)

Request body parameters: none required

Status codes: 200, 401, 403 (not own alerts or not admin/teacher), **404** (user not found), **429**

Caching: Cache-Control: private, max-age=60

Example request:

```
GET /api/v1/users/123/alerts?page=1&limit=10 HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: private, max-age=60
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]

{
 "data": [
  {
   "id": "501",
   "user_id": "123",
   "assignment_id": "201",
   "alert_type": "upcoming_deadline",
   "message": "Reminder: Calculus Homework 1 is due in 24 hours",
   "sent_at": "2023-09-14T08:30:00Z",
      "read_at": null
    },
    {
      "id": "502",
      "user_id": "123",
      "assignment_id": "202",
      "alert_type": "grade_posted",
      "message": "Your grade for Linear Algebra Quiz has been posted",
      "sent_at": "2023-10-12T15:45:00Z",
      "read_at": "2023-10-12T16:20:00Z"
    }
  ],
  "meta": {
    "pagination": {
      "total": 2,
      "pages": 1,
      "page": 1,
      "limit": 10
    }
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/users/123/alerts?page=1&limit=10",
    "next": null,
    "prev": null,
    "first": "https://api.student-deadline-sceduling.com/api/v1/users/123/alerts?page=1&limit=10",
    "last": "https://api.student-deadline-sceduling.com/api/v1/users/123/alerts?page=1&limit=10",

  }
}
```

`GET /api/v1/alerts/{id}` *retrieve details of a specific alert.*

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	Alert id

Request body parameters: none required

Status codes: 200, 401, 403 (not alert recipient, admin, or teacher), **404** (alert not found), **429**

Caching: Cache-Control: private, max-age=60

Example request:

```
GET /api/v1/alerts/501 HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: private, max-age=60
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]

{
  "data": {
    "id": "501",
    "user_id": "123",
    "assignment_id": "201",
    "alert_type": "upcoming_deadline",
    "message": "Reminder: Calculus Homework 1 is due in 24 hours",
    "sent_at": "2023-09-14T08:30:00Z",
    "read_at": null
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/alerts/501",
    "user": "https://api.student-deadline-sceduling.com/api/v1/users/123",
    "assignment": "https://api.student-deadline-sceduling.com/api/v1/assignments/201"
  }
}
```

`POST /api/v1/alerts/{id}/read` ***mark an alert as read (alert recipient only).***

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	Alert id

Request body parameters: none required

Status codes: 200, 401, 403 (not alert recipient), **404** (alert not found), **429**

Caching: Cache-Control: no-store

Example request:

```
POST /api/v1/alerts/501/read HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "data": {
    "id": "501",
    "user_id": "123",
    "assignment_id": "201",
    "alert_type": "upcoming_deadline",
    "message": "Reminder: Calculus Homework 1 is due in 24 hours",
    "sent_at": "2023-09-14T08:30:00Z",
    "read_at": "2023-09-14T09:15:00Z"
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/alerts/501",
    "user": "https://api.student-deadline-sceduling.com/api/v1/users/123",
    "assignment": "https://api.student-deadline-sceduling.com/api/v1/assignments/201"
  }
}
```

`PATCH /api/v1/alerts/{id}` *update an alert (teachers and admins only, before sending)*

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	Alert id

Request body parameters:

Parameter	Type	Required	Description
alert_type	string	No	Type of alert (upcoming deadline, grade posted, grade changed, feedback added, deadline today)
message	string	No	Alert content

One of the parameters must be provided.

Status codes: 200, 401, 400 (invalid data), **403** (not admin or teacher; already sent), **404** (enrollment not found), **429**

Caching: Cache-Control: no-store

Example request:

```
PATCH /api/v1/alerts/501 HTTP/1.1
Host: api.student-deadline-sceduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "message": "URGENT: Calculus Homework 1 is due in 24 hours"
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "data": {
    "id": "501",
    "user_id": "123",
    "assignment_id": "201",
    "alert_type": "upcoming_deadline",
    "message": "URGENT: Calculus Homework 1 is due in 24 hours",
    "sent_at": null,
    "read_at": null
  },
  "links": {
    "self": "https://api.student-deadline-sceduling.com/api/v1/alerts/501",
    "user": "https://api.student-deadline-sceduling.com/api/v1/users/123",
    "assignment": "https://api.student-deadline-sceduling.com/api/v1/assignments/201"
  }
}
```

`DELETE /api/v1/alerts/{id}` *delete an alert (teachers and admins only)*

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	Alert id

Request body parameters: none required

Status codes: 204, 401, 403 (not admin or teacher; the alert already is sent), **404** (alert not found), **429**

Caching: Cache-Control: no-store

Example request:

```
DELETE /api/v1/alerts/501 HTTP/1.1
Host: api.student-deadline-scheduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Example response:

```
HTTP/1.1 204 No Content
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store
```

Teachers

`GET /api/v1/courses/{id}/teachers` *list all teachers id for a course*

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	Course id

Request body parameters: none required

Status codes: 200, 401, 404 (course not found), **429**

Caching: Cache-Control: public, max-age=300

Example request:

```
GET /api/v1/courses/101/teachers HTTP/1.1
Host: api.student-deadline-scheduling.com
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: public, max-age=300
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]

{
   "data":
    {
       "teachers": ["125", "126", "55"]
   },
  "links": {
   "self": "https://api.student-deadline-scheduling.com/api/v1/courses/101/teachers",
   "course": "https://api.student-deadline-scheduling.com/api/v1/courses/101",
   "126": "https://api.student-deadline-scheduling.com/api/v1/users/126",
   "125": "https://api.student-deadline-scheduling.com/api/v1/users/125",
   "55": "https://api.student-deadline-scheduling.com/api/v1/users/55"
  }
}
```

`DELETE /api/v1/courses/{id}/teachers` *delete a teacher from a course (admin only)*

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	Course id

Request body parameters:

Parameter	Type	Required	Description
teachers	array of strings	Yes	ids of teachers to delete

Status codes: 204, **400** (invalid data)**, 401**, **403** (not admin), **404** (course or teacher not found), **409** (teacher is not in this course), **429**

Caching: Cache-Control: no-store

Example request:

```
DELETE /api/v1/courses/101/teachers HTTP/1.1
Host: api.student-deadline-scheduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "teachers": ["127"]
}
```

Example response:

```
HTTP/1.1 204 No Content
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store
```

`PATCH /api/v1/courses/{id}/teachers` *add a teacher to a course (admin only)*

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	Course id

Request body parameters:

Parameter	Type	Required	Description
teachers	array of strings	Yes	ids of teachers to add

Status codes: 200, **400** (invalid data)**, 401**, **403** (not admin), **404** (course or teacher not found), **409** (teacher already assigned to course), **429**

Caching: Cache-Control: no-store

Example request:

```
PATCH /api/v1/courses/101/teachers HTTP/1.1
Host: api.student-deadline-scheduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "teachers": ["127"]
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: public, max-age=300
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]

{
  "data":
   {
    "teachers": ["125", "126", "55", "127"]
  },
 "links": {
   "self": "https://api.student-deadline-scheduling.com/api/v1/courses/101/teachers",
   "course": "https://api.student-deadline-scheduling.com/api/v1/courses/101",
  "126": "https://api.student-deadline-scheduling.com/api/v1/users/126",
  "125": "https://api.student-deadline-scheduling.com/api/v1/users/125",
  "55": "https://api.student-deadline-scheduling.com/api/v1/users/55",
  "127": "https://api.student-deadline-scheduling.com/api/v1/users/127"
 }
}
```

Prerequisites

`GET /api/v1/courses/{id}/prerequisites` ***list all prerequisites for a course***

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	Course id

Request body parameters: none required

Status codes: 200, **401**, **404** (course not found), **429**

Caching: Cache-Control: public, max-age=300

Example request:

```
GET /api/v1/courses/101/prerequisites HTTP/1.1
Host: api.student-deadline-scheduling.com
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: public, max-age=300
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]

{
 "data": [
  {
   "prerequisites": ["95", "96"]
  },
 "links": {
  "self": "https://api.student-deadline-scheduling.com/api/v1/courses/101/prerequisites",
  "course": "https://api.student-deadline-scheduling.com/api/v1/courses/101",
  "95": "https://api.student-deadline-scheduling.com/api/v1/courses/95",
      "96": "https://api.student-deadline-scheduling.com/api/v1/courses/96"
    }
  }
```

`PATCH /api/v1/courses/{id}/prerequisites` *add a prerequisite(s) to a course (teachers and admins only)*

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	Course id

Request body parameters:

Parameter	Type	Required	Description
prerequisites	array of strings	Yes	ids of prerequisite courses to add

Status codes: 200, 400 (invalid data), **401, 403** (not teacher or admin), **404** (course or prerequisite course not found), **409** (prerequisite already exists or would create circular reference), **429**

Caching: Cache-Control: no-store

Example request:

```
PATCH /api/v1/courses/101/prerequisites HTTP/1.1
Host: api.student-deadline-scheduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

{
  "prerequisites": ["97"]
}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store

{
  "data": {
    "prerequisites": ["95", "96", "97"],
  },
  "links": {
    "self": "https://api.student-deadline-scheduling.com/api/v1/courses/101/prerequisites/97",
    "course": "https://api.student-deadline-scheduling.com/api/v1/courses/101",
    "95": "https://api.student-deadline-scheduling.com/api/v1/courses/95",
    "96": "https://api.student-deadline-scheduling.com/api/v1/courses/96",
    "97": "https://api.student-deadline-scheduling.com/api/v1/courses/97",
  }
}
```

`DELETE /api/v1/courses/{id}/prerequisites` *remove a prerequisite(s) from a course (teachers and admins only)*

Path parameters:

Parameter	Type	Required	Description
id	string	Yes	Course id

Request body parameters:

Parameter	Type	Required	Description
prerequisites	array of strings	Yes	ids of prerequisite courses to delete

Status codes: 204, 401, 403 (not teacher or admin), **404** (course, prerequisite, or relationship not found), **429**

Caching: Cache-Control: no-store

Example request:

```
DELETE /api/v1/courses/101/prerequisites HTTP/1.1
Host: api.student-deadline-scheduling.com
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
{
   "prerequisites": ["97"]
}
```

Example response:

```
HTTP/1.1 204 No Content
X-RateLimit-Limit: 100
X-RateLimit-Remaining: [calculated value]
X-RateLimit-Reset: [seconds until reset]
Cache-Control: no-store
```