Controller API Documentation

Overview

This document serves as the comprehensive API documentation for the Cegep Management System. Designed to facilitate the integration of external systems with the Cegep management platform, this API offers a wide range of functionalities to interact with various entities within the system. These entities include academic years, addresses, courses, departments, employees, evaluations, mail, personal information, programs, schedules, sessions, students, teachers, and types.

The API endpoints are structured to provide seamless access to the system's features, enabling clients to perform operations such as retrieving, creating, updating, and deleting data entities. Whether it's managing academic calendars, accessing student records, or scheduling courses, the Cegep Management System API offers robust capabilities to meet diverse needs.

Developers can utilize the provided endpoints to build custom applications, integrate with third-party services, or automate administrative tasks within the Cegep environment. With clear documentation and standardized request/response formats, developers can quickly understand and leverage the API's capabilities to enhance the functionality and efficiency of their systems.

Current Version : 1.2.5-Snapshot -- The Url, Will only update when Version 2.0.0-Alpha is released.

Base URL

All endpoints described in this document are relative to the base URL:

```
https://cms-api-1crm.onrender.com/api/v1
```

Each controller's documentation will detail the available endpoints, their functionalities, request methods, and expected responses.

Endpoints

Introduction

This section provides an overview of the available controllers and their corresponding endpoints.

- `/academicYears`
- `/addresses`
- `/courses`
- `/departments`
- `/employees`
- `/evaluations`
- `/mail`
- `/person`
- `/programs`
- `/schedules`
- `/sessions`
- `/students`
- `/teachers`
- `/types`

AcademicYearController

Description

The `AcademicYearController` is responsible for managing academic years in the Cegep Management System. This controller provides a set of APIs for creating, retrieving, updating, and deleting academic years. It interacts with the `AcademicYearService` to perform these operations.

Request Methods

GET /academicYears

Gets a list of all academic years.

Example URL: `https://example.com/api/academicYears`

Example Request Body: None

Response: A list of `AcademicYear` objects in JSON format

Example Response:

```
[
  {
    "id": 1,
    "startDate": "2022-09-01",
    "endDate": "2023-08-31",
  },
  {
    "id": 2,
    "startDate": "2021-09-01",
    "endDate": "2022-08-31",
  }
]
```

GET /academicYears/

Gets a specific academic year by ID.

Example URL: `https://example.com/api/academicYears/123` Example Request Body: None Response: A single `AcademicYear` object in JSON format

Example Response:

```
{
  "id": 123,
  "startDate": "2022-09-01",
  "endDate": "2023-08-31",
}
```

POST /academicYears

Creates a new academic year.

Example URL: `https://example.com/api/academicYears`

Example Request Body:

```
{
  "startDate": "2022-09-01",
  "endDate": "2023-08-31",
}
```

Response: The newly created `AcademicYear` object in JSON format

Example Response:

```
{
  "id": 456,
  "startDate": "2022-09-01",
  "endDate": "2023-08-31",
  "name": "Academic Year 2022-2023"
}
```

POST /academicYears/default

Creates or look for an already created AcademicYear object, if none find it will create a new academic year with the currentDate and default to the following dates :

- StartDate = year/08/22
- EndDate = year/05/26

Example URL: `https://example.com/api/academicYears/default`

Example Request Body:

```
{
  "currentDate": "2022-09-01"
}
```

Response: The newly created `AcademicYear` object in JSON format

Example Response:

```
{
  "id": 789,
  "startDate": "2022-08-22",
  "endDate": "2023-05-26",
}
```

PUT /academicYears/

Updates an existing academic year. Altought this should **never** be done.

Example URL: `https://example.com/api/academicYears/123`

Example Request Body:

```
{
  "startDate": "2022-09-01",
  "endDate": "2023-08-31",
}
```

Response: The updated `AcademicYear` object in JSON format

Example Response:

```
{
  "id": 123,
  "startDate": "2022-09-01",
  "endDate": "2023-08-31",
}
```

DELETE /academicYears/

Deletes an existing academic year.

Example URL: `https://example.com/api/academicYears/123`

Example Request Body: None

Response: A 204 No Content status code

AddressController

Description

The `AddressController` is responsible for managing addresses in the CEgep Management System. This controller provides a set of APIs for creating, retrieving, updating, and deleting addresses. It interacts with the `AddressService` and `PersonService` to perform these operations.

Request Methods

GET /addresses

Gets a list of all addresses.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/addresses`

Example Request Body: None

Response: A list of `Address` objects in JSON format

Example Response:

```
[
  {
    "id": 1,
    "person": {** Refer to the Person Controller **},
    "address": "123 Main St",
    "city": "Anytown",
    "province": "CA",
    "postalCode": "12345"
  },
  {
    "id": 2,
    "person": {** Refer to the Person Controller **},
    "address": "456 Elm St",
    "city": "Othertown",
    "province": "NY",
    "postalCode": "67890"
  }
]
```

GET /addresses/

Gets a specific address by ID.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/addresses/123`

Example Request Body: None

Response: A single `Address` object in JSON format

Example Response:

```
{
  "id": 123,
  "person": {** Refer to the Person Controller **},
  "address": "123 Main St",
  "city": "Anytown",
  "province": "CA",
  "postalCode": "12345"
}
```

GET /addresses/byUser/

Gets a list of addresses associated with a specific user.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/addresses/byUser/123`

Example Request Body: None

Response: A list of `Address` objects in JSON format

Example Response:

```
[
  {
    "id": 1,
        "person": {** Refer to the Person Controller **},
    "address": "123 Main St",
    "city": "Anytown",
    "province": "CA",
    "postalCode": "12345"
  },
  {
    "id": 2,
        "person": {** Refer to the Person Controller **},
    "address": "456 Elm St",
    "city": "Othertown",
    "province": "NY",
    "postalCode": "67890"
  }
]
```

POST /addresses

Creates a new address.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/addresses`

Example Request Body:

```
{
  "userId": 1
  "address": "789 Oak St",
  "city": "Thistown",
  "province": "Qc",
  "postalCode": "Z3X 3W2"
}
```

Response: The newly created `Address` object in JSON format

Example Response:

```
{
  "id": 456,
  "person": {** Refer to the Person Controller **}, --> Refer to a Person with id: 1.
  "address": "789 Oak St",
  "city": "Thistown",
  "province": "Qc",
  "postalCode": "Z3X 3W2"
}
```

PUT /addresses/

Updates an existing address.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/addresses/123`

Example Request Body:

```
{
  "id": 456,
  "person": {** Refer to the Person Controller **},
  "address": "789 Oak St",
  "city": "NewTown", --> Changes right here.
  "province": "Qc",
  "postalCode": "Z3X 3W2"
}
```

Response: The updated `Address` object in JSON format

Example Response:

```
{
  "id": 456,
  "person": {** Refer to the Person Controller **},
  "address": "789 Oak St",
  "city": "NewTown", --> Changes are reflected here.
  "province": "Qc",
  "postalCode": "Z3X 3W2"
}
```

DELETE /api/v1/addresses/

Deletes an existing address.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/addresses/123`

Example Request Body: None

Response: A 204 No Content status code

CourseController

Description

The `CourseController` is responsible for managing courses in the CEgep Management System. This controller provides a set of APIs for creating, retrieving, updating, and deleting courses. It interacts with the `CourseService` to perform these operations.

Request Methods

GET /courses

Gets a list of all courses.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/courses`

Example Request Body: None

Response: A list of `Course` objects in JSON format

Example Response:

```
[
  {
    "id": 1,
    "name": "Introduction to Computer Science",
    "sigle": "CS101",
    "department": {
      "id": 1,
      "name": "Computer Science"
    },
    "teacher": {** Refer to the Teacher Controller**}
  },
  {
    "id": 2,
    "name": "Data Structures and Algorithms",
    "sigle": "CS202",
    "department": {
      "id": 1,
      "name": "Computer Science"
    },
    "teacher": {** Refer to the Teacher Controller**}
  }
]
```

GET /courses/

Gets a specific course by ID.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/courses/123`

Example Request Body: None

Response: A single `Course` object in JSON format

Example Response:

```
{
  "id": 123,
  "name": "Introduction to Computer Science",
  "sigle": "CS101",
  "department": {
    "id": 1,
    "name": "Computer Science"
  },
  "teacher": {** Refer to the Teacher Controller**}
}
```

POST /courses

Creates a new course.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/courses`

Example Request Body:

```
{
  "name": "Introduction to Computer Science",
  "sigle": "CS101",
  "departmentId": 1,
  "teacherId": 1
}
```

Response: The newly created `Course` object in JSON format

Example Response:

```
{
  "name": "Introduction to Computer Science",
  "sigle": "CS101",
  "department": {** Refer to the Department Controller**}
  "teacher": {** Refer to the Teacher Controller**}
}
```

PUT /courses/

Updates an existing course.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/courses/123` Example Request Body:

```
{
  "name": "Introduction to Computer Science Next Level", --> Changes right here
  "sigle": "CS101",
  "department": {** Refer to the Department Controller**} --> Same goes for here. And all other Update
                                                                                                    method.
  "teacher": {** Refer to the Teacher Controller**} --> You could also update a new Teacher,
                                                                                                In that case, for no
                                                                                                create a new Teacher
                                                                                                the API and pass it
}
```

Response: The updated `Course` object in JSON format

Example Response:

```
{
  "name": "Introduction to Computer Science Next Level", --> Change are reflected here.
  "sigle": "CS101",
  "department": {** Refer to the Department Controller**}
  "teacher": {** Refer to the Teacher Controller**}
}
```

DELETE /courses/

Deletes an existing course.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/courses/123`

Example Request Body: None

Response: A 204 No Content status code

POST /courses/

Enrolls a student in a course.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/courses/123/enroll/456` Example Request Body: None Response: The student object in JSON format

Example Response:

```
{
        "student": {** Refer to the Student Controller**}
}
```

DELETE /courses/

Unenrolls a student from a course.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/courses/123/enroll/456` Example Request Body: None Response: A 204 No Content status code

DepartmentController

Description

The `DepartmentController` is responsible for managing departments in the CEgep Management System. This controller provides a set of APIs for creating, retrieving, updating, and deleting departments. It interacts with the `DepartmentService` to perform these operations.

Request Methods

GET /departments

Gets a list of all departments.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/departments`

Example Request Body: None

Response: A list of `Department` objects in JSON format

Example Response:

```
[
  {
    "id": 1,
    "name": "Computer Science"
  },
  {
    "id": 2,
    "name": "Mathematics"
  }
]
```

GET /departments/

Gets a specific department by ID.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/departments/123`

Example Request Body: None

Response: A single `Department` object in JSON format

Example Response:

```
{
  "id": 123,
  "name": "Computer Science"
}
```

GET /departments/name/

Gets a department by name.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/departments/name/Computer Science`

Example Request Body: None

Response: A single `Department` object in JSON format

Example Response:

```
{
  "id": 123,
  "name": "Computer Science"
}
```

POST /api/v1/departments

Creates a new department.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/departments`

Example Request Body:

```
{
  "name": "Electrical Engineering"
}
```

Response: The newly created `Department` object in JSON format

Example Response:

```
{
  "id": 456,
  "name": "Electrical Engineering"
}
```

PUT /api/v1/departments/

Updates an existing department.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/departments/123` Example Request Body:

```
{
  "name": "New Name"
}
```

Response: The updated `Department` object in JSON format

Example Response:

```
{
  "id": 123,
  "name": "New Name"
}
```

DELETE /departments/

Deletes an existing department.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/departments/123`

Example Request Body: None

Response: A 204 No Content status code**# DepartmentController**

EmployeeController

Description

The `EmployeeController` is responsible for managing employees in the Cegep Management System. This controller provides a set of APIs for creating, retrieving, updating, and deleting employees. It interacts with the `EmployeeService` to perform these operations.

Request Methods

GET /employees

Gets a list of all employees.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/employees` Example Request Body: None Response: A list of `Employee` objects in JSON format

Example Response:

```
[
  {
    "id": 1,
    "person": {** Refer to the Person Controller**},
    "seniority": "New Employee",
    "type": {** Refer to the Type Controller**}
  },
  {
    "id": 2,
    "person": {** Refer to the Person Controller**},
    "seniority": "Experienced Employee",
    "type": {** Refer to the Type Controller**}
  },
]
```

GET /employees/

Gets a specific employee by ID.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/employees/1`

Example Request Body: None

Response: A single `Employee` object in JSON format

Example Response:

```
  {
    "id": 1,
    "person": {** Refer to the Person Controller**},
    "seniority": "New Employee",
    "type": {** Refer to the Type Controller**}
  }
```

POST /employees

Creates a new employee.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/employees`

Example Request Body:

```
  "firstName": "John",
  "lastName": "Doe",
  "email": "johndoe@example.com",
  "phone": "1234567890",
  "password": "password",
  "dateOfBirth": "1990-01-01",
  "seniority": "New Employee",
  "typeId": 1,
  "personId": 1, --> Note: This ID can be NULL, simply don't specify it. If you do that, it will
                                              default to the value you've specifiy and create a new "Person" Object
                                              On the other hand, if you do specify the Id, you don't have to give the
                                              "Person" informations, such as the first 6 field.
  "departmentId": 1
}
```

Response: The newly created `Employee` object in JSON format

Example Response:

```
  {
    "id": 1,
    "person": {** Refer to the Person Controller**},
    "seniority": "New Employee",
    "type": {** Refer to the Type Controller**}
  }
```

PUT /employees/

Updates an existing employee.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/employees/123`

Example Request Body:

```
{
  "Pass the same as if you were creating an employee."
}
```

Response: The updated `Employee` object in JSON format

Example Response:

```
{
  "Same response as if you were creating an employee."
}
```

DELETE /employees/

Deletes an existing employee.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/employees/123`

Example Request Body: None

Response: A 204 No Content status code

Notes

- When creating a new employee, you can specify the ID of an existing person, or the controller will create a new person object directly.

EvaluationController

Description

The `EvaluationController` is responsible for managing evaluations in the CEgep Management System. This controller provides a set of APIs for creating, retrieving, updating, and deleting evaluations. It interacts with the `EvaluationService` to perform these operations.

Request Methods

GET /evaluations

Gets a list of all evaluations.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/evaluations`

Example Request Body: None

Response: A list of `Evaluation` objects in JSON format

Example Response:

```
[
  {
    "id": 1,
    "name": "Midterm Exam",
    "courseId": 123,
    "ponderation": 30,
    "denominator": 100
  },
  {
    "id": 2,
    "name": "Final Project",
    "courseId": 456,
    "ponderation": 20,
    "denominator": 50
  }
]
```

GET /evaluations/

Gets a specific evaluation by ID.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/evaluations/123`

Example Request Body: None

Response: A single `Evaluation` object in JSON format

Example Response:

```
{
  "id": 123,
  "name": "Midterm Exam",
  "courseId": 123,
  "ponderation": 30,
  "denominator": 100
}
```

POST /evaluations

Creates a new evaluation.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/evaluations`

Example Request Body:

```
{
  "name": "New Evaluation",
  "courseId": 789,
  "ponderation": 40,
  "denominator": 80
}
```

Response: The newly created `Evaluation` object in JSON format

Example Response:

```
{
  "id": 789,
  ...
}
```

PUT /evaluations/

Updates an existing evaluation.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/evaluations/123`

Example Request Body:

```
{
  "Pass the same as if you were creating an evaluation."
}
```

Response: The updated `Evaluation` object in JSON format

Example Response:

```
{
  "Same response as if you were creating an evaluation."
}
```

DELETE /evaluations/

Deletes an existing evaluation.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/evaluations/123`

Example Request Body: None

Response: A 204 No Content status code

Additional Endpoints

GET /evaluations/students/

Gets a list of evaluations for a specific student.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/evaluations/students/123`

Example Request Body: None

Response: A list of `Evaluation` objects in JSON format

Example Response:

```
[ { "id": 123, ... }, { "id": 456, ... } ]
```

POST /evaluations/students/

Adds an evaluation to a student.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/evaluations/students/123/evaluations/456`

Example Request Body: None

Response: The updated `Student` object in JSON format

Example Response:

```
{ ... }
```

DELETE /evaluations/students/

Removes an evaluation from a student.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/evaluations/students/123/evaluations/456`

Example Request Body: None

Response: A 204 No Content status code

MailController

Description

The `MailController` is responsible for managing emails in the CEgep Management System. This controller provides a set of APIs for creating, retrieving, updating, and deleting emails. It interacts with the `MailService` to perform these operations.

Request Methods

GET /mail

Gets a list of all emails.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/mail` Example Request Body: None Response: A list of `Mail` objects in JSON format

Example Response:

```
[
  {
    "id": 1,
    "receiver": {** Refer to the Person Controller **},
    "sender": {** Refer to the Person Controller **},
    "subject": "Hello",
    "content": "This is a test email",
    "date": "2022-01-01T12:00:00"
  },
  {
    "id": 2,
    "receiver": {** Refer to the Person Controller **},
    "sender": {** Refer to the Person Controller **},
    "subject": "Test Email",
    "content": "This is another test email",
    "date": "2022-01-02T14:00:00"
  }
]
```

GET /mail/

Gets a specific email by ID.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/mail/123`

Example Request Body: None

Response: A single `Mail` object in JSON format

Example Response:

```
{
    "id": 123,
    "receiver": {** Refer to the Person Controller **},
    "sender": {** Refer to the Person Controller **},
    "subject": "Test Email",
    "content": "This is another test email",
    "date": "2022-01-02T14:00:00"
}
```

GET /mail/bysender/

Gets a list of emails sent by a specific sender.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/mail/bysender/123`

Example Request Body: None

Response: A list of `Mail` objects in JSON format

Example Response:

```
[
  {
    "id": 1,
    "receiver": {** Refer to the Person Controller **},
    "sender": {** Refer to the Person Controller **}, --> The person with the Id you passed
    "subject": "Hello",
    "content": "This is a test email",
    "date": "2022-01-01T12:00:00"
  },
  {
    "id": 2,
    "receiver": {** Refer to the Person Controller **},
    "sender": {** Refer to the Person Controller **},
    "subject": "Test Email",
    "content": "This is another test email",
    "date": "2022-01-02T14:00:00"
  }
]
```

GET /mail/byreceiver/

Gets a list of emails received by a specific receiver.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/mail/byreceiver/456`

Example Request Body: None

Response: A list of `Mail` objects in JSON format

Example Response:

```
[
  {
    "... Same as with the bySender"
  },
  {
    "..."
  }
]
```

POST /mail

Creates a new email.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/mail`

Example Request Body:

```
{
    "receiver": {** Refer to the Person Controller **},
    "sender": {** Refer to the Person Controller **},
    "subject": "Test Email",
    "content": "This is another test email",
    "date": "2022-01-02T14:00:00"
}
```

Response: The newly created `Mail` object in JSON format

Example Response:

```
{
  "id": 789,
  ...
}
```

PUT /mail/

Updates an existing email.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/mail/123`

Example Request Body:

```
{
  "Pass the same as if you were creating a Mail."
}
```

Response: The updated `Mail` object in JSON format

Example Response:

```
{
  "Same response as if you were creating a Mail."
}
```

DELETE /mail/

Deletes an existing email.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/mail/123`

Example Request Body: None

Response: A 204 No Content status code

Notes

- The `MailDTO` object is used to create and update mail objects. The properties of the `MailDTO` object are used to populate the corresponding fields of the `Mail` object. This is how you can directly use the ID of the sender/receiver.

PersonController

Description

The `PersonController` is responsible for managing users in the CEgep Management System. This controller provides a set of APIs for creating, retrieving, updating, and deleting users. It interacts with the `PersonService` to perform these operations.

Request Methods

GET /person

Gets a list of all users.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/person`

Example Request Body: None

Response: A list of `Person` objects in JSON format

Example Response:

```
[
  {
    "id": 1,
    "firstName": "John",
    "lastName": "Doe",
    "email": "johndoe@example.com",
    "phone": "1234567890",
    "password": (hashed)"passowrd"
    "dateOfBirth": "1990-01-01"
  },
  {
    "id": 2,
    "firstName": "Jane",
    "lastName": "Doe",
    "email": "janedoe@example.com",
    "phone": "9876543210",
    "password":  (hashed)"passowrd"
    "dateOfBirth": "1992-02-02"
  }
]
```

GET /person/

Gets a specific user by ID.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/person/123`

Example Request Body: None

Response: A single `Person` object in JSON format

Example Response:

```
{
  "id": 123,
  "firstName": "John",
  ...
}
```

GET /person/email/

Gets a specific user by email.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/person/email/johndoe@example.com`

Example Request Body: None

Response: A single `Person` object in JSON format

Example Response:

```
{
  "id": 123,
  "firstName": "John",
  ...
}
```

POST /person

Creates a new user.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/person`

Example Request Body:

```
{
  "firstName": "John",
  "lastName": "Doe",
  "email": "johndoe@example.com",
  "phone": "1234567890",
  "password": (not hashed)"password"
  "dateOfBirth": "1990-01-01"
}
```

Response: The newly created `Person` object in JSON format

Example Response:

```
{
  "id": 789,
  "firstName": "John",
  ...
}
```

PUT /person/

Updates an existing user.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/person/123`

Example Request Body:

```
{ ... }
```

Response: The updated `Person` object in JSON format

Example Response:

```
{ ... }
```

DELETE /api/v1/person/

Deletes an existing user.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/person/123`

Example Request Body: None

Response: A 204 No Content status code

Notes

- The `Person` object is used to represent a user. The properties of the `Person` object are used to populate the corresponding fields of the user.
- The person object is the root object of all three : Student/Employee/Teacher

ProgramController

Description

The `ProgramController` is responsible for managing programs in the CEgep Management System. This controller provides a set of APIs for creating, retrieving, updating, and deleting programs. It interacts with the `ProgramService` to perform these operations.

Request Methods

GET /programs

Gets a list of all programs.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/programs`

Example Request Body: None

Response: A list of `Program` objects in JSON format

Example Response:

```
[
  {
    "id": 1,
    "name": "Computer Science",
    "department": {** Refer to the Department Controller**}
  },
  {
    "id": 2,
    "name": "Electrical Engineering",
    "department": {** Refer to the Department Controller**}
  }
]
```

GET /programs/

Gets a specific program by ID.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/programs/123`

Example Request Body: None

Response: A single `Program` object in JSON format

Example Response:

```
{
        "id": 123,
    "name": "Electrical Engineering",
    "department": {** Refer to the Department Controller**}
}
```

POST /programs

Creates a new program.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/programs`

Example Request Body:

```
{
  "name": "New Program",
  "departmentId": 789
}
```

Response: The newly created `Program` object in JSON format

Example Response:

```
{
  "id": 987
  "name": "New Program",
  "department": {** Refer to the Department Controller**}
}
```

PUT /programs/

Updates an existing program.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/programs/123`

Example Request Body:

```
{ ... }
```

Response: The updated `Program` object in JSON format

Example Response:

```
{ ... }
```

DELETE /programs/

Deletes an existing program.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/programs/123`

Example Request Body: None

Response: A 204 No Content status code

Notes

- The `ProgramDTO` object is used to create and update programs. The properties of the `ProgramDTO` object are used to populate the corresponding fields of the program.
- The `Program` object is used to represent a program. The properties of the `Program` object are used to populate the corresponding fields of the program.
- The `ProgramService` is used to interact with the database and perform operations on programs.

ScheduleController

Description

The `ScheduleController` is responsible for managing schedules in the CEgep Management System. This controller provides a set of APIs for creating, retrieving, updating, and deleting schedules. It interacts with the `ScheduleService` to perform these operations.

Request Methods

GET /schedules

Gets a list of all schedules.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/schedules`

Example Request Body: None

Response: A list of `Schedule` objects in JSON format

Example Response:

```
[
  {
    "id": 1,
    "studentId": 123,
    "courseId": 456,
    "hourStart": "09:00",
    "hourEnd": "10:00"
  },
  {
    "id": 2,
    "student": {** Refer to the Student Controller},
    "course": {** Refer to the Course Controller},
    "hourStart": "11:00",
    "hourEnd": "12:00"
  }
]
```

GET /schedules/

Gets a specific schedule by ID.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/schedules/123`

Example Request Body: None

Response: A single `Schedule` object in JSON format

Example Response:

```
{
    "id": 123,
    "student": {** Refer to the Student Controller},
    "course": {** Refer to the Course Controller},
    "hourStart": "11:00",
    "hourEnd": "12:00"
}
```

GET /schedules/students/

Gets a list of schedules for a specific student by ID.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/schedules/students/123`

Example Request Body: None

Response: A list of `Schedule` objects in JSON format

Example Response:

```
[
  {
    ...
  },
  {
    ...
  }
]
```

POST /schedules

Creates a new schedule.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/schedules`

Example Request Body:

```
{
  "studentId": 123,
  "courseId": 456,
  "hourStart": "09:00",
  "hourEnd": "10:00"
}
```

Response: The newly created `Schedule` object in JSON format

Example Response:

```
{ "id": 789, ... }
```

PUT /schedules/

Updates an existing schedule.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/schedules/123`

Example Request Body:

```
{
  "studentId": 123,
  "courseId": 456,
  "hourStart": "08:00", --> change here
  "hourEnd": "10:00"
}
```

Response: The updated `Schedule` object in JSON format

Example Response:

```
{ ... }
```

DELETE /schedules/

Deletes an existing schedule.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/schedules/123`

Example Request Body: None

Response: A 204 No Content status code

Notes

- The `ScheduleDTO` object is used to create and update schedules. The properties of the `ScheduleDTO` object are used to populate the corresponding fields of the schedule.
- The `Schedule` object is used to represent a schedule. The properties of the `Schedule` object are used to populate the corresponding fields of the schedule.
- The `ScheduleService` is used to interact with the database and perform operations on schedules.

SessionController

Description

The `SessionController` is responsible for managing sessions in the CEgep Management System. This controller provides a set of APIs for creating, retrieving, updating, and deleting sessions. It interacts with the `SessionService` to perform these operations.

Request Methods

GET /sessions

Gets a list of all sessions.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/sessions`

Example Request Body: None

Response: A list of `Session` objects in JSON format

Example Response:

```
[
  {
    "id": 1,
    "name": "Fall 2022",
    "startDate": "2022-09-01",
    "endDate": "2022-12-31"
  },
  {
    "id": 2,
    "name": "Winter 2023",
    "startDate": "2023-01-01",
    "endDate": "2023-03-31"
  }
]
```

GET /sessions/

Gets a specific session by ID.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/sessions/123`

Example Request Body: None

Response: A single `Session` object in JSON format

Example Response:

```
{
  "id": 2,
  "name": "Winter 2023",
  "startDate": "2023-01-01",
  "endDate": "2023-03-31"
}
```

PUT /sessions/

Updates an existing session.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/sessions/123`

Example Request Body:

```
{
  "id": 2,
  "name": "Winter 2023 2", --> Changes are right here
  "startDate": "2023-01-01",
  "endDate": "2023-03-31"
}
```

Response: The updated `Session` object in JSON format

Example Response:

```
{ ... }
```

DELETE /sessions/

Deletes an existing session.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/sessions/123`

Example Request Body: None

Response: A 204 No Content status code

GET /sessions/current

Gets the current session for the current date.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/sessions/current`

Example Request Body: None

Response: A single `Session` object in JSON format

Example Response:

```
{ "id": 789, ... }
```

Notes

- The `Session` object is used to represent a session. The properties of the `Session` object are used to populate the corresponding fields of the session.
- The `SessionService` is used to interact with the database and perform operations on sessions.
- You cannot create 'Session' Object, they are automatically created when you insert a new Student in the database, if there isnt already A session in the database. Otherwise, you "can" create a session using the /sessions/default which will create a session to the ccurrentDate if there's no session currently in the database for the currentDate.

StudentController

Description

The `StudentController` is responsible for managing students in the CEgep Management System. This controller provides a set of APIs for creating, retrieving, updating, and deleting students. It interacts with the `StudentService` to perform these operations.

Request Methods

GET /students

Gets a list of all students.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/students`

Example Request Body: None

Response: A list of `Student` objects in JSON format

Example Response:

```
[
  {
    "id": 1,
    "person": {** Refer to the Person Controller},
    "program": {** Refer to the Program Controller},
    "session": {** Refer to the session Controller},
    "field": "Computer Science"
  },
  {
    "id": 2,
    "person": {** Refer to the Person Controller},
    "program": {** Refer to the Program Controller},
    "session": {** Refer to the session Controller},
    "field": "Computer Science"
  }
]
```

GET /students/

Gets a specific student by ID.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/students/123`

Example Request Body: None

Response: A single `Student` object in JSON format

Example Response:

```
{
        "id": 2,
    "person": {** Refer to the Person Controller},
    "program": {** Refer to the Program Controller},
    "session": {** Refer to the session Controller},
    "field": "Computer Science"
}
```

POST /students

Creates a new student.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/students`

Example Request Body:

```
{
        "firstName":  "John",
        "lastName":  "Doe",
        "email":  "johndoe@example.com",
        "phone":  "1234567890",
        "password":  (not hashed)"password",
        "dateOfBirth":  "1990-01-01",
        "programId": 1,
        "sessionId": 1, --> You will mostly let this empty, since it will automatically retrieve
                                            or create the correct Session and even AcademicYear if necessary.
        "field": "Computer Science"
}
```

Response: The newly created `Student` object in JSON format

Example Response:

```
{
        "id": 2,
    "person": {** Refer to the Person Controller},
    "program": {** Refer to the Program Controller},
    "session": {** Refer to the session Controller},
    "field": "Computer Science"
}
```

PUT /students/

Updates an existing student.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/students/123` Example Request Body:

```
{
        "firstName":  "John",
        "lastName":  "Doe",
        "email":  "johndoe@example.com",
        "phone":  "1234567890",
        "password":  (not hashed)"password",
        "dateOfBirth":  "1990-01-01",
        "programId": 2, --> Change the program, you can do this below in an easier way ;)
        -- Remove Session, not needed even when updating.
        "field": "Computer Science"
}
```

Response: The updated `Student` object in JSON format

Example Response:

```
{
        "id": 2,
    "person": {** Refer to the Person Controller},
    "program": {** Refer to the Program Controller},
    "session": {** Refer to the session Controller},
    "field": "Computer Science"
}
```

PUT /students/

Updates an existing student's program.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/students/123/program/456`

Example Request Body: None

Response: The updated `Student` object in JSON format

Example Response:

```
{
        "id": 2,
    "person": {** Refer to the Person Controller},
    "program": {** Refer to the Program Controller},
    "session": {** Refer to the session Controller},
    "field": "Computer Science"
}
```

DELETE /students/

Deletes an existing student.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/students/123`

Example Request Body: None

Response: A 204 No Content status code

Notes

- The `StudentDTO` object is used to create and update students. The properties of the `StudentDTO` object are used to populate the corresponding fields of the student.
- The `Student` object is used to represent a student. The properties of the `Student` object are used to populate the corresponding fields of the student.
- The `StudentService` is used to interact with the database and perform operations on students.
- When creating & deleting a student, it will also impact the 'Person' table.

TeacherController

Description

The `TeacherController` is responsible for managing teachers in the CEgep Management System. This controller provides a set of APIs for creating, retrieving, updating, and deleting teachers. It interacts with the `TeacherService` to perform these operations.

This Controller is highly likely to change in the future, bear in mind that the current EndPoint are for testing purposes only.

Request Methods

GET /teachers

Gets a list of all teachers.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/teachers`

Example Request Body: None

Response: A list of `Teacher` objects in JSON format

Example Response:

```
[
  {
    "id": 1,
    "employee": {** Refer to the Employee Controller},
    "department": {** Refer to the Department Controller},
  },
  {
    "id": 2,
    "employee": {** Refer to the Employee Controller},
    "department": {** Refer to the Department Controller},
  }
]
```

GET /teachers/

Gets a specific teacher by ID.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/teachers/123`

Example Request Body: None

Response: A single `Teacher` object in JSON format

Example Response:

```
{
  "id": 2,
  "employee": {** Refer to the Employee Controller},
  "department": {** Refer to the Department Controller},
}
```

GET /teachers/

Gets a list of all courses for a specific teacher.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/teachers/123/courses`

Example Request Body: None

Response: A list of `Course` objects in JSON format

Example Response:

```
[
  {
    "id": 1,
    "employee": {** Refer to the Employee Controller},
    "department": {** Refer to the Department Controller},
  },
  {
    "id": 2,
    "employee": {** Refer to the Employee Controller},
    "department": {** Refer to the Department Controller},
  }
]
```

GET /teachers/

Gets a list of all students for a specific teacher.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/teachers/123/students`

Example Request Body: None

Response: A list of `Student` objects in JSON format

Example Response:

```
[
  {
    {** Refer to the Student Controller}
  },
  {
    {** Refer to the Student Controller}
  }
]
```

POST /teachers

Creates a new teacher.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/teachers`

Example Request Body:

```
  "firstName": "John",
  "lastName": "Doe",
  "email": "johndoe@example.com",
  "phone": "1234567890",
  "password": "password",
  "dateOfBirth": "1990-01-01",
  "seniority": "New Employee",
  "typeId": 1,
  "personId": 1, --> Note: This ID can be NULL, simply don't specify it. If you do that, it will
                                                 default to the value you've specifiy and create a new "Person" Object
                                                 On the other hand, if you do specify the Id, you don't have to give the
                                                 "Person" informations, such as the first 6 field.
  "departmentId": 1
}
```

Response: The newly created `Teacher` object in JSON format

Example Response:

```
{
  "id": 2,
  "employee": {** Refer to the Employee Controller},
  "department": {** Refer to the Department Controller},
}
```

As you can see, there is no difference between this and the employee POST request, the only difference is where they are.

PUT /teachers/

Updates an existing teacher.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/teachers/123`

Example Request Body:

```
{ ...like when creating a teacher object but with modified data. }
```

Response: The updated `Teacher` object in JSON format

Example Response:

```
{ ... }
```

DELETE /api/v1/teachers/

Deletes an existing teacher.

Example URL: `https://cms-api-1crm.onrender.com/api/v1/teachers/123`

Example Request Body: None

Response: A 204 No Content status code

Notes

- The `TeacherDTO` object is used to create and update teachers. The properties of the `TeacherDTO` object are used to populate the corresponding fields of the teacher.
- The `Teacher` object is used to represent a teacher. The properties of the `Teacher` object are used to populate the corresponding fields of the teacher.
- The `TeacherService` is used to interact with the database and perform operations on teachers.

Type Controller

Introduction:

This controller EndPoint aren't exposed, first, it's very unlikely you will ever need to interact with it. Secondly, if you really need to interact with it, here is the EndPoint themselves. It's up to you to understand them...

- Get /types
- Get /types/
- Get /types/name/
- Post /types
- Put /types/
- Delete /types/

Error Handling

- If a requested resource is not found, the API returns a 404 Not Found status.
- If an invalid request is made, the API returns a 400 Bad Request status.
- If an internal server error occurs, the API returns a 500 Internal Server Error status.

Technology Used

The Cegep Management System API is built using the following technologies:

- Spring Boot: a Java-based framework for building web applications
- Lombok: a library that simplifies the process of creating getters and setters for Java objects

- Hibernate: a persistence library for interacting with the database

Development

The Cegep Management System API was developed, overviewed and documented by William Beaudin.

Conclusion

The Cegep Management System API is a powerful tool for managing. With its robust set of APIs and flexible architecture, it provides a solid foundation for building a comprehensive management system. By using Spring Boot, Lombok, and Hibernate, the API takes advantage of the latest technologies and best practices in software development.