Singleton

User

- id: int
- name:str
- surname:str
- email:str

+ getter/setter

Current User

- token: str
- role: int
- user: Current User

+ login(password)
+ logout()

Singleton

Static

Classroom

- id: str
- campus: Campus

+ getter/setter

Professor

- id: int
- rating: int
- rating_count: int

+ getter/setter
- calculateRating(new_rating): int
+ addRating(new_rating): bool

Department

- id: int
- name: str
- faculty_name: str

+ getter/setter

Course

- id: str
- name: str
- department: Department
- type: Type
- rating_count: int
- rating: int

+ getter/setter
+ addRating(new_rating): bool
- calculateRating(new_rating): int

Section

- id: int
- course: Course
- professor: Professor
- start_time: DateTime(?)
- end_time: DateTime(?)
- week_day: int
- classroom: ClassRoom

+ getter/setter

<<Enumeration>>
Type

+ Face to Face
+ Online
+ Asynchronous

<<Enumeration>>
Campus

a

<<Enumeration>>
Role

a

API

- instance: API

+ getClassroom(id: str): Classroom
+ getCourse(id: str): Course
+ getDepartment(id: int): Department
+ getProfessor(id: int) Professor
+ getSection(id: int, course_id: str): Section
+ getUser(iid: str): User
+ getClassrooms(count: int): Classroom[]
+ getCourses(count: int): Course[]
+ getUsers(count: int): User[]
+ getSections(icourse_id: str): Section[]
+ getProfessors(count: int) Professor[]
+ getDepartments(): Department[]
+ getCredentials(email: str): String[]