

Service	In	Out
Authentication service	POST /login { password: string, netid: string }	JWT token containing authorities, this JWT is passed in the Bearer header in every following request
	POST /register { password: string, netid: string }	
	DELETE /unregister { netid?: string }	
	POST /modifyUser { password: string, netid?: string }	
Courses service: Port: 8085	POST /createCourse { name: string, code: string, year: int }	Course created in the database, get JSON back with status
	POST /modifyCourse { courseId: int, name?: string, code?: string, year?: int }	Returns the edited course
	POST /createLecture/{courseId} { name: string, teacherNetId: String, date: yyyy-mm-dd, nSlots: int }	Lecture created in the database, get HTTP status back and the JSON contains the created lecture
	GET /getMyCourses	[{ name: string, code: string, year: int, id: int }, { name: string, code: string, year: int, id: int }, { name: string, code: string, year: int, id: int }, etc]
	POST /moveLectureOnline { lectureId: int }	
Room Schedule service Port : 8081	POST /scheduleLecturesUntil { date: string }	A JSON with all the scheduled lectures to rooms, including the time

	POST /createRule { name: string, value: string }	"Saved Rule"
	GET /allrules	Return all rules
	PUT /modifyRule { id: int, name: string, value: string }	"Changed Rule" or "Rule not found for the id ..."
	DELETE /deleteRule/{id}	"Deleted Rule" or "Rule not found for the id ..."
	GET /countRooms	Return the number of rooms in the database
	PUT /generateRoomSlots/{numDays}/{firstSlotDateTime}	"Room Slots Created"
	POST /scheduleLecture { lectureId: int }	
	DELETE /cancelLecture { lectureId: int }	
	GET /getSchedule	[{ roomSlotId: int, lectureId: int }, ...]
Room service Port: 8080	POST /createRoom { name: string, capacity: int }	"Saved Room"
	GET /room/{id}	Return the room with the given id
	GET /allrooms	Return all rooms
	DELETE /deleteRoom/{id}	"Deleted Room" or "Room not found for the id ..."
	PUT /modifyRoom { id: int, name: string, capacity: int }	"Changed Room" or "Room not found for the id ..."
	GET /countRooms	Return the number of rooms in the database
	PUT /generateRooms	"Rooms Created"
Student service Port: 8087	POST /assignStudentsUntil { date: string }	
	GET /allMyLectures	[{ lectureId: int, courseName: String, roomName: String, ... }]

		onCampus: boolean },]
	GET /allMyCourses	[{ courseName: string, yearOfStudy: integer }]
	POST /declineOnCampusOffer { lectureId: int }	
	POST /setPreferences { wantsToGoToCampus: boolean }	