

### 1. Course Resource (courseId is a String)

GET: ../webapi/courses/allcourses	list all courses
GET: ../webapi/courses/{courseId}	list one course by courseId
GET: ../webapi/courses/professor/{professor}	list all courses taught by a professor name
GET: ../webapi/courses	using QueryParam studentTA - list courses belongs to a TA
GET: ../webapi/courses/{courseId}/professor	list the professor's information of a course
GET: ../webapi/courses/{courseId}/TA	list the TA's information of a course
DELETE: ../webapi/courses/{courseId}	delete a course
POST: ../webapi/courses	add a course
PUT: ../webapi/courses/{courseId}	update a course
PUT: ../webapi/courses/{courseId}/TA	update the TA's information of a course (request body should be a student object)
PUT: ../webapi/courses/{courseId}/professor	update the professor's information of a course (request body should be a professor object)

### 2. Course Roster Resource

GET: ../webapi/courses/{courseId}/roster	get the roster information of a course
DELETE: ../webapi/courses/{courseId}/roster	clear the roster of a course (not delete, just clear)
POST: ../webapi/courses/{courseId}/roster	add a student to the roster of a course (request body should be a student object)
PUT: ../webapi/courses/{courseId}/roster	delete a student from the roster of a course (request body should be a student object: you can just give it a studentId)

### 3. Course Board Resource

GET: ../webapi/courses/{courseId}/board	get the board information of a course
---	---------------------------------------

#### 4. Lecture Resource

GET: ../webapi/courses/{courseId}/lectures	get all lectures of a course
DELETE: ../webapi/courses/{courseId}/lectures	delete a lecture of a course  (request body should be a lecture: you can just give it a lectureId which is already exist in the memory)
POST: ../webapi/courses/{courseId}/lectures	add a lecture to a course  (request body should be a lecture)
PUT: ../webapi/courses/{courseId}/lectures	update a lecture of a course  (request body should be a lecture whose lectureId is already exist in the memory)

#### 5. Professor Resource

GET: ../webapi/professors/allprofessors	list all professors
GET: ../webapi/professors	using QueryParam department. list all professor of a particular department
GET: ../webapi/professors/{professorId}	list a professor
DELETE: ../webapi/professors/{professorId}	delete a professor
POST: ../webapi/professors/{professorId}	add a new professor
PUT: ../webapi/professors/{professorId}	update a professor

#### 6. Program Resource

GET: ../webapi/programs/allprograms	list all programs
GET: ../webapi/programs/{programId}	list a program by programId
GET: ../webapi/programs	using QueryParam program: list a program by name
GET: ../webapi/programs/{programId}/courses	list all courses of a program by programId
GET: ../webapi/programs/{programId}/students	list all students of a program by programId
DELETE: ../webapi/programs/{programId}	delete a program

DELETE: ../webapi/programs/{programId}/courses      delete a course of a program (request body should be a course object: you can just give it a courseId (it is a String such as "CYSE6225"))

POST: ../webapi/programs      add a new program

POST: ../webapi/programs/{programId}/courses      add a course to a program (request body should be a course object)

PUT: ../webapi/programs/{programId}      update a program

## 7. Student Resource

GET: ../webapi/students      using QueryParam programName: list all students of a particular program

GET: ../webapi/students/{studentId}      list a student

DELETE: ../webapi/students/{studentId}      delete a student

POST: ../webapi/students      add a new student

PUT: ../webapi/students/{studentId}      update a student