

# RESTFul API Design

---

- **School**
  - 1. GET api/school
  - 2. GET api/school/{school\_id}
  - 3. POST /api/school + json
  - 4. PUT /api/school/{school\_id} + json
  - 5. DELETE /api/school/{school\_id}
  - 6. GET api/school/{school\_id}/courses
  - 7. GET api/school/{school\_id}/tree
  - 8. POST /api/school/{school\_id}/tree + json
  - 9. PUT /api/school/{school\_id}/tree + json
  - 10. DELETE /api/school/{school\_id}/tree + json
- **Course**
  - 1. GET api/course
  - 2. GET api/course/{course\_id}
  - 3. POST api/course + json + file
  - 4. PUT api/course/{course\_id} + json
  - 5. DELETE api/course/{course\_id}
  - 6. GET api/course/{course\_id}/syllabus
  - 7. POST api/course/{course\_id}/syllabus + PDF
  - 8. PUT api/course/{course\_id}/syllabus + PDF
  - 9. DELETE api/course/{course\_id}/syllabus
- **User**
  - 1. GET api/user
  - 2. GET api/user/{user\_id}
  - 3. POST api/user + json
  - 4. PUT api/user + json
  - 5. DELETE api/user

## School

---

### 1. GET api/school

a. Description

**Get** schools list. b. Return Format: **json**

```
{  
  "school_ids": [1, 2, 3, 4]  
}
```

### 2. GET api/school/{school\_id}

a. Description

**Get** a specific school with id `school_id` b. Return Format: **json**

```
{
  "school_id": 1,
  "school_name": "SUSTech",
  "website": "sustc.edu.cn",
  "description": "Best university of the world."
}
```

### 3. POST /api/school + json

a. Description

**Add** a school to database.

### 4. PUT /api/school/{school\_id} + json

a. Description **Update** a school in database. b. Request Body: **json**

```
{
  "school_name": "SUSTech",
  "website": "sustc.edu.cn",
  "description": "Best university of the world."
}
```

### 5. DELETE /api/school/{school\_id}

a. Description

**Delete** a school from database.

### 6. GET api/school/{school\_id}/courses

a. Description

**Get** all courses id of the school with id `school_id` . b. Return Format: **json**

```
{
  "course_ids": [1, 2, 3, 4]
}
```

### 7. GET api/school/{school\_id}/tree

a. Description

**Get** a json that represents the relationship of all courses in that school.

b. Return Format: **json**

```
{
  "nodes": [
    { "name": "A" },
    { "name": "B" }
  ],
  "links": [
    { "source": "A", "target": "B" }
  ]
}
```

```
}
```

## 8. POST /api/school/{school\_id}/tree + json

a. Description

**Add** courses relation tree for a school. POST body is a json same as above.

## 9. PUT /api/school/{school\_id}/tree + json

a. Description

**Modify** courses relation tree for a school. PUT body is a json same as above.

## 10. DELETE /api/school/{school\_id}/tree

a. Description

**Delete** courses relation tree for a school.

---

# Course

## 1. GET api/course

a. Description

**Get** courses list. b. Return Format: **json**

```
{
  "course_ids": [1, 2, 3, 4]
}
```

**Note:** this command return all courses id stored in the databases, not only the courses in one school!

## 2. GET api/course/{course\_id}

a. Description

**Get** a specific course with id `course_id`. b. Return Format: **json**

```
{
  "course_id": 1,
  "course_code": "CS303",
  "course_name": "Database Principle",
  "description": "this is a course",
  "professor": "John",
  "author": 12,
  "school": 1
}
```

**Note:** The Response does not contain the syllabus(Which is PDF format). So you should send another request to server in order to get it.

### 3. POST api/course + json

a. Description

**Add** a specific course.

b. Procedures:

1. First we need to post a json which contains the basic information of the course except the syllabus file to the server.

Request Body: **json**

```
{
  "course_code": "CS303",
  "course_name": "Database Principle",
  "description": "this is a course",
  "professor": "John",
  "author": 12,
  "school": 1
}
```

2. If the json is accepted by the server, the server will return the id of the course. Then we post syllabus with this id in order to associate it with the course we created previously.

Request Body: **multipart form data**

|        |              |
|--------|--------------|
| [file] | syllabus.pdf |
|--------|--------------|

### 4. PUT api/course/{course\_id} + json

a. Description **Update** a specific course (when you need to update the syllabus, use **8**).

### 5. DELETE api/course/{course\_id}

a. Description

**Delete** a specific course with id `course_id`.

### 6. GET api/course/{course\_id}/syllabus

a. Description

**Get** a syllabus of a course with id `course_id` b. Return Format: **PDF**

### 7. POST api/course/{course\_id}/syllabus + PDF

a. Description

**Add** a syllabus of a course with id `course_id`

### 8. PUT api/course/{course\_id}/syllabus + PDF

a. Description

**Update** a syllabus of a course with id `course_id`

## 9. DELETE api/course/{course\_id}/syllabus

a. Description

**Delete** a syllabus of a course with id `course_id`

---

## User

### 1. GET api/user

a. Description

Return all users in the databases b. Return Format: **json**

```
{
  "user_ids": [1, 2, 3, 4]
}
```

### 2. GET api/user/{user\_id}

a. Description

Get a specific user with id `user_id` b. Return Format: **json**

```
{
  "name": "John",
  "email": "John@gmail.com",
  "password": "123456",
  "school": "SUSTech"
}
```

**Note:** When logged in, the `user_id` of current user will be stored in cookie.

### 3. POST api/user + json

a. Description

**Add** a user to database.

### 4. PUT api/user + json

a. Description

**Update** a user to database.

### 5. DELETE api/user

a. Description

**Delete** a user to database.