

Postman and Swagger Demo

SWEN 343: Engineering of Enterprise Software Systems

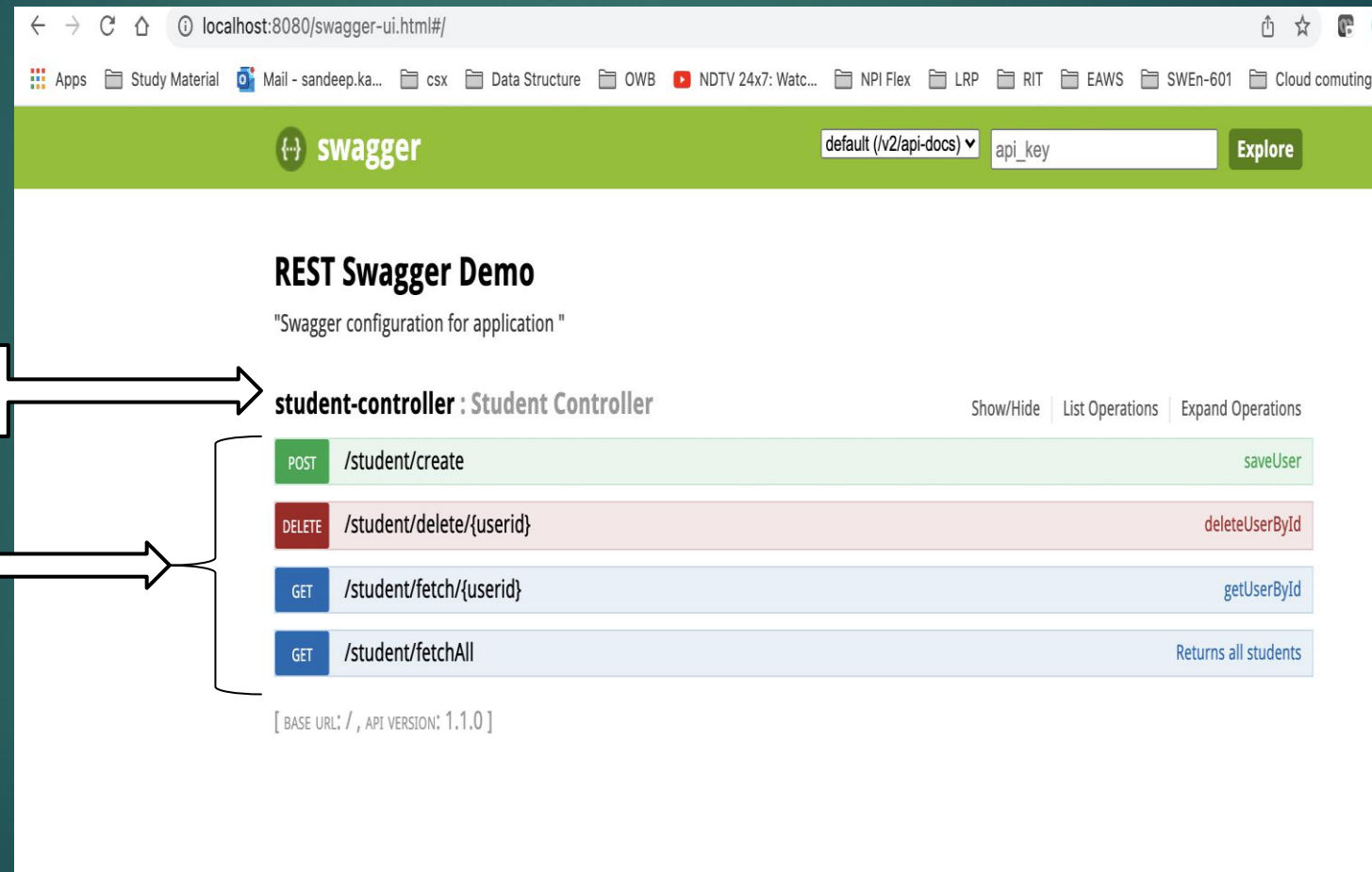
Department of Software Engineering
Rochester Institute of Technology

Swagger Overview

- ▶ It is an Open API description for REST API's
- ▶ Swagger is an Interface for describing RESTful APIs expressed using JSON.
- ▶ It can be used by various tools to automate API-related processes.
- ▶ It can be used to share documentation among product managers, testers and developers

Swagger Homepage

<http://localhost:8080/swagger-ui.html#/>



Name of the Controller

All the REST API's handled by this controller

There is Just one controller in this example. In real applications there are many controllers

POST request

5

POST /student/create

Response Class (Status 200)
OK

Model | Model Schema

```
{
  "country": "string",
  "firstname": "string",
  "lastname": "string",
  "studentId": "string",
  "zipCode": "string"
}
```

Response Content Type: */* ▼

Parameters

Parameter	Value	Description
user	<pre>{ "country": "United States", "firstname": "Kenn", "lastname": "Martinez", "studentId": "01", "zipCode": "14623" }</pre>	user

Parameter content type: application/json ▼

Response
after
hitting
the API

Request URL

http://localhost:8080/student/create

Request Headers

```
{
  "Accept": "*/*"
}
```

Response Body

```
{
  "firstname": "Kenn",
  "lastname": "Martinez",
  "country": "United States",
  "zipCode": "14623",
  "studentId": "01"
}
```

Response Code

200

GET Request

6

Response Model Structure

GET /student/fetch/{userid}

Response Class (Status 200)
OK

Model | Model Schema

```
{  
  "country": "string",  
  "firstname": "string",  
  "lastname": "string",  
  "studentId": "string",  
  "zipCode": "string"  
}
```

Response Content Type

Parameters

Parameter	Value	Description	Parameter Type	Data Type
userid	<input type="text" value="01"/>	userid	path	string

Response Messages

HTTP Status Code	Reason	Response Model
401	Unauthorized	
403	Forbidden	
404	Not Found	

[Hide Response](#)

Request Param

Click on this button

Delete Request

7

DELETE /student/delete/{studentid}

Response Class (Status 200)

Response Content Type

Parameters

Parameter	Value	Description
studentid	<input type="text" value="(required)"/>	studentid

Response Messages

HTTP Status Code	Reason	Response Model
204	No Content	
401	Unauthorized	
403	Forbidden	

Required
Request
Param



Response
after
hitting
the API



Response Body

Deleted Successfully

Response Code

200

Postman Overview

8

- ▶ Postman is an API client which makes easier for developers to create, share, test and document APIs.
- ▶ It allows users to create and save simple and complex HTTP/s requests, as well as read their responses

GET Request

9

Request Type

Request URL

Click to add request params

GET Params Send

Make sure it's checked

GET Params Send

Key	Value	Description
<input checked="" type="checkbox"/> studentId	01	
New key	Value	Description

Authorization Headers (1) Body Pre-request Script Tests

Type No Auth

Body Cookies Headers (5) Test Results Status: 200 OK

Pretty Raw Preview JSON

```
1 {
2   "firstname": "Kenn",
3   "lastname": "Martinez",
4   "country": "US",
5   "zipCode": "14623",
6   "studentId": "01"
7 }
```

Response In JSON format

Different Formats to view output

JSON

JSON

XML

HTML

Text

Auto

Post Request

10

Select body to pass an Object/file/form data or other information

Request Body in raw JSON format

The screenshot shows a REST client interface for a POST request to `http://localhost:8080/student/create`. The interface includes a 'Send' button and a 'Params' tab. Below the URL bar, there is a table with columns 'Key', 'Value', and 'Description'. The 'Body' tab is selected, and the 'raw' radio button is chosen. The request body is displayed as raw JSON:

```
1 {  
2   "country": "US",  
3   "firstname": "Kenn",  
4   "lastname": "Martinez",  
5   "studentId": "01",  
6   "zipCode": "14623"  
7 }
```