

Sample Project API Documentation

1. INTRODUCTION

This Application has been designed in Spring Boot and deployed on Apache Tomcat. This application is deployed on 10.10.0.103 server.

2. BUILDING APPLICATIONS WITH THIS API

3. API FUNCTIONS

- Azure Login with JWT
- CRUD Operation
- File Upload & Download

3. API URL

Methods	Urls	Actions
POST	/api/auth/ signin	login with azure account
POST	/api/auth/signout	logout the account
GET	/api/post	retrieve Post data
GET	/api/post/{id}	retrieve post by id
POST(form-data)	/api/post/user	Create Post
POST(form-data)	/api/post/{id}	Update Post data by id
DELETE	/api/post/{id}	Delete Post data by id

1. LOGIN WITH AZURE (MICROSOFT AUTHENTICATION LIBRARY)

HTTP Method: POST

URL: <http://10.10.0.103:8080/angularAPI/api/auth/signin>

Description: The Sentiment Analysis

Request Parameters:

Parameter Name	Description
email	
sub	
fname	
lname	
mname	

JSON Response:

```
{
  "id": 1,
  "username": null,
  "email": "santosh.sarkar@telecommand.com",
  "roles": [],
  "tokenType": "Bearer",
  "accessToken":
"eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJzYW50b3NoLnNhcmthckB0ZWxIY29tbWFuZC5jb20iLCJpYXQiOiE2NjlyNzIzNjY5ImlV4cCI6MTY2MjM1OTE2Nn0.xDEft-9d_DHqn0-vv8oznSIHLL1pVLbRdr2fwuJBslRW8gs_P-i-0-f0was7jhnB93KJqz621qic1UtGUuSezg"
}
```

Example:

POST

http://10.10.0.103:8080/angularAPI/api/auth/signin

Params

Send

Save

Authorization

Headers (1)

Body

Pre-request Script

Tests

Code

form-data

x-www-form-urlencoded

raw

binary

JSON (application/json)

```

1 {
2   "email": "santosh.sarkar@telecommand.com",
3   "sub": "1234",
4   "fname": "Santosh",
5   "lname": "Sarkar",
6   "mname": "Kumar"
7 }

```

Body

Cookies

Headers (13)

Test Results

Status: 200 OK

Time: 944 ms

Pretty

Raw

Preview

JSON

Save Response

```

1 {
2   "id": 1,
3   "username": null,
4   "email": "santosh.sarkar@telecommand.com",
5   "roles": [],
6   "tokenType": "Bearer",
7   "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bm90cn5SIHLLIpVLBrd2FwuJBSlRw6gs_P-1-0-f0was7jhnB93KJqz621qic1UtGUuSezg"
8 }

```

2. FETCHING ALL POST

HTTP Method: GET

URL: <http://10.10.0.103:8080/angularAPI/api/post/>

Description: The Sentiment Analysis

JSON Response:

```
{
  "data": [
    {
      "id": 6,
      "title": "Funding Overview",
      "body": "Before You Apply: Technical Difficulties Regarding an",
      "published": false,
      "email": "aush@gmail.com",
      "project": "Located in downtown",
      "organisation": "Telcommand",
      "file": null,
      "postFiles": []
    },
    {
      "id": 12,
      "title": "Innovation Initiative",
      "body": " It helps everyone in the team of organization identify what the project ",
      "published": true,
      "email": "Tayler.45@telecommand.com",
      "project": "Performance Measure",
      "organisation": "Hq Software Consultant",
      "file": "Certfication.PNG",
      "postFiles": [
        {
          "id": 4,
          "fileName": "Certfication.PNG"
        }
      ]
    }
  ],
  "message": "Successfully retrieved data!",
  "status": 200
}
```

3. CREATING POST

HTTP Method: POST

URL: <http://10.10.0.103:8080/angularAPI/api/post/save>

Description: This request should be sent on form Data.

Request Parameters:

Parameter Name	type	Description
title	text	
body	text	
project	text	
organisation	text	
email	text	
files	file	
published	boolean	

JSON Response:

```
{
  "data": {
    "id": 13,
    "title": "Title test 2",
    "body": "Body Test 2",
    "published": true,
    "email": "test3@gmail.com",
    "project": "Project 3",
    "organisation": "Organisation 3",
    "file": "LMS.pdf",
    "postFiles": null
  },
  "message": "Successfully saved data!",
  "status": 200
}
```

Example:

The screenshot displays a REST client interface for a POST request. The request is configured with the following details:

- Method:** POST
- URL:** <http://10.10.0.103:8080/angularAPI/api/post/save>
- Authorization:** form-data
- Headers:** 1 header is set.
- Body:** 7 parameters are defined:
 - title: Title test 2
 - body: Body Test 2
 - published: true
 - files: LMS.pdf (selected via 'Choose Files')
 - project: Project 3
 - organisation: Organisation 3
 - email: test3@gmail.com

The response is shown in JSON format, indicating a successful save:

```
{
  "data": {
    "id": 13,
    "title": "Title test 2",
    "body": "Body Test 2",
    "published": true,
    "email": "test3@gmail.com",
    "project": "Project 3",
    "organisation": "Organisation 3",
    "file": "LMS.pdf",
    "postFiles": null
  },
  "message": "Successfully saved data!",
  "status": 200
}
```

The status bar at the bottom shows a status of 200 OK and a response time of 2692 ms.

4. UPDATING POST

HTTP Method: POST

URL: <http://10.10.0.103:8080/angularAPI/api/post/update/3>

Description: This request should be sent on form Data.

Request Parameters:

Parameter Name	type	Description
title	text	
body	text	
project	text	
organisation	text	
email	text	
files	file	
published	boolean	

JSON Response:

```
{
  "data": {
    "id": 3,
    "title": "Update title",
    "body": "Body Update",
    "published": false,
    "email": "santosh@gmail.com",
    "project": "Project Update ",
    "organisation": "Org Update",
    "file": "Doc1.docx",
    "postFiles": null
  },
  "message": "Successfully Update data!",
  "status": 200
}
```

Example:

The screenshot displays a REST client interface with a POST request to `http://localhost:8080/angularAPI/api/post/update/3`. The request body is configured as form-data with the following fields:

Key	Value	Description
<input checked="" type="checkbox"/> title	Update title	
<input checked="" type="checkbox"/> body	Body Update	
<input checked="" type="checkbox"/> project	Project Update	
<input checked="" type="checkbox"/> organisation	Org Update	
<input checked="" type="checkbox"/> email	santosh@gmail.com	
<input checked="" type="checkbox"/> published	false	
<input checked="" type="checkbox"/> files	<input type="button" value="Choose Files"/> Doc1.docx	
New key	Value	Description

The response is shown in JSON format:

```
1 {
2   "data": {
3     "id": 3,
4     "title": "Update title",
5     "body": "Body Update",
6     "published": false,
7     "email": "santosh@gmail.com",
8     "project": "Project Update ",
9     "organisation": "Org Update",
10    "file": "Doc1.docx",
11    "postFiles": null
12  },
13  "message": "Successfully Update data!",
14  "status": 200
15 }
```

4. DELETE POST

HTTP Method: DELETE

URL: <http://10.10.0.103:8080/angularAPI/api/post/3>

Description: This request will delete the Data.

JSON Response:

```
{
  "data": 3,
  "message": "Data 3 successfully deleted!",
  "status": 200
}
```

Example:

The screenshot displays a REST client interface with the following components:

- Method and URL:** A dropdown menu shows 'DELETE' and the URL is 'http://localhost:8080/angularAPI/api/post/3'. There are 'Send' and 'Save' buttons.
- Headers Tab:** The 'Headers' tab is active, showing a table with columns 'Key', 'Value', and 'Description'. It contains one entry: 'Authorization' with a Bearer token. There are also 'Bulk Edit' and 'Presets' options.
- Body Tab:** The 'Body' tab is active, showing the JSON response in 'Pretty' format. The response is:

```
{
  "data": 3,
  "message": "3 suceessfully deleted!",
  "status": 200
}
```

 The status is '200 OK' and the time is '8966 ms'.

5. FILE DOWNLOAD

HTTP Method: GET

URL: <http://10.10.0.103:8080/angularAPI/api/post/file/3/nippon.pdf>

Description: This request should be Get.

Request Parameters:

Parameter Name	type	Description
id	number	On URL parameters
filename	text	On URL parameters

Example:

DownloadFile

Examples (0)

GET

http://localhost:8080/angularAPI/api/post/file/3/nippon.pdf

Params

Send

Save

Authorization

Headers (1)

Body

Pre-request Script

Tests

Code

Key	Value	Description	...	Bulk Edit	Presets
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzUxMiJ9.eyJzdWliOiJzYW50b3NoLnNhcmth...				
New key	Value	Description			

Body

Cookies

Headers (13)

Test Results

Status: 200 OKTime: 28 ms