

Adding new Schedule Requests – Faculty/Staff

The screenshot displays the Postman application interface. At the top, the 'Overview' tab is selected, and the 'POST Trainings' collection is active. The request URL is set to 'http://localhost:3000/api/trainings'. The 'Body' tab is selected, and the request is formatted as 'raw' JSON. The JSON body contains the following data:

```
1 {
2   "Title": "Full Stack Web Development",
3   "Resource": "Online Platform",
4   "Domain": "Software Engineering",
5   "Technical": "JavaScript, React, Node.js",
6   "Year": "2024",
7   "StartDate": "2024-03-10 09:00:00",
8   "EndDate": "2024-03-20 17:00:00",
9   "Description": "Comprehensive training on full-stack web development.",
10  "TrainingStatus": "Completed",
11  "TrainerID": "TR1234",
12  "Duration": 10,
13  "Assessments": "Final Project, MCQ Test",
14 }
```

The 'Send' button is visible, and the response is displayed in the 'Body' tab. The response status is '201 Created', with a response time of 131 ms and a size of 246 B. The response body is formatted as 'Pretty' JSON and shows the following data:

```
1 {
2   "id": 1
3 }
```

The bottom of the interface shows a navigation bar with icons for Postbot, Runner, Start Proxy, Cookies, Vault, Trash, and a help icon.

Fetching Schedule Requests – Faculty/Staff

The screenshot displays the Postman interface for a GET request to `http://localhost:3000/api/trainings`. The request is configured with the following details:

- Method:** GET
- URL:** `http://localhost:3000/api/trainings`
- Body:** The request body is empty, as indicated by the `{}` in the raw view.
- Response:** The response is a 200 OK status, received in 22 ms with a body size of 876 B. The response is displayed in the "Body" tab, showing a JSON array with one object.

The JSON response is as follows:

```
[
  {
    "ID": 1,
    "Title": "Full Stack Web Development",
    "Resource": "Online Platform",
    "Domain": "Software Engineering",
    "Technical": "JavaScript, React, Node.js",
    "Year": "2024",
    "StartDate": "2024-03-10T03:30:00.000Z",
    "EndDate": "2024-03-20T11:30:00.000Z",
    "Description": "Comprehensive training on full-stack web development."
  }
]
```

Getting Requests By Id to view Submitted Requests – Faculty/Staff

ork

Search Postman

Invite

Upgrade

Overview

GET Trainings

No environment

Placement Training / Trainings

Save

Share

GET

http://localhost:3000/api/trainings/1

Send

Params

Authorization

Headers (7)

Body

Scripts

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

1

Body

Cookies

Headers (7)

Test Results

200 OK

11 ms

874 B

Pretty

Raw

Preview

Visualize

JSON

1 {

2 "ID": 1,

3 "Title": "Full Stack Web Development",

4 "Resource": "Online Platform",

5 "Domain": "Software Engineering",

6 "Technical": "JavaScript, React, Node.js",

7 "Year": "2024",

8 "StartDate": "2024-03-10T03:30:00.000Z",

9 "EndDate": "2024-03-20T11:30:00.000Z",

10 "Description": "Comprehensive training on full-stack web development.",

11 "TrainingStatus": "Completed",

Postbot

Runner

Start Proxy

Cookies

Vault

Trash

Updating Training Status - Admin

Overview

PUT Trainings

No environment

Placement Training / Trainings

SaveShare

PUThttp://localhost:3000/api/trainings/1/statusSend

ParamsAuthorizationHeaders (9)BodyScriptsTestsSettingsCookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```
1  {}
2  |  "status": "Ongoing"
3  |  }
```

BodyCookiesHeaders (7)Test Results200 OK • 13 ms • 285 B

PrettyRawPreviewVisualizeJSON

```
1  {}
2  |  "message": "Training status updated successfully"
3  |  }
```

PostbotRunnerStart ProxyCookiesVaultTrash

Updating Venue Details - Admin

Search Postman

Invite

Upgrade

Overview

PUT Trainings

Placement Training

PUT New Request

No environment

HTTP

Placement Training / New Request

Save

Share

PUT

http://localhost:3000/api/trainings/1/venue

Send

Params

Authorization

Headers (9)

Body

Scripts

Tests

Settings

Cookies

Beautify

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1 {
2   |   "venueDetails": "Room 204, Building A"
3 }
```

Body

Cookies

Headers (7)

Test Results

200 OK

107 ms

283 B

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   |   "message": "Venue details updated successfully"
3 }
```

Postbot

Runner

Start Proxy

Cookies

Vault

Trash

Updating Request Status - Admin

The screenshot shows the Postman interface for a PUT request. The URL is `http://localhost:3000/api/trainings/1/request-status`. The request body is a JSON object: `{ "status": "Rejected", "remarks": "Requirements not met" }`. The response is a 200 OK status with a response time of 20 ms and a body size of 284 B. The response body is a JSON object: `{ "message": "Request status updated successfully" }`.

PUT `http://localhost:3000/api/trainings/1/request-status` Send

Params Authorization Headers (9) Body Scripts Tests Settings Cookies Beautify

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   "status": "Rejected",
3   "remarks": "Requirements not met"
4 }
```

Body Cookies Headers (7) Test Results 200 OK • 20 ms • 284 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Request status updated successfully"
3 }
```

Postbot Runner Start Proxy Cookies Vault Trash

Email Notifications from Nodemailer

The screenshot shows an email inbox with a test email from Nodemailer. The email is from `dharanielangovan26@gmail.com` to `dharani.cb21, ragul.se21, me`. The subject is "Test Email from Nodemailer". The email is dated 5 Feb 2025, 15:57. The email is signed by `gmail.com` and is encrypted using Standard encryption (TLS). The email is marked as important because previous messages in the conversation were important. The email body contains the text: "Hello! This is a test email sent using Nodemailer."

from: dharanielangovan26@gmail.com
to: priyaelango285@gmail.com, dharani.cb21@bitsathy.ac.in, ragul.se21@bitsathy.ac.in
date: 5 Feb 2025, 15:57
subject: Test Email from Nodemailer
mailed-by: gmail.com
Signed by: gmail.com
security: Standard encryption (TLS) [Learn more](#)
Important because previous messages in the conversation were important.

Hello! This is a test email sent using Nodemailer.

Reply Reply to all Forward