

Installation Instructions

Required Softwares:

- Eclipse IDE
- mySQL
- Active Internet connection
- Postman (If you want to test the API)

Instructions:

- Download zipfile from github titled ActivityTrackerFinal.zip ([Link](#))
- Extract this file and save it to your preferred folder
- Download activitytrackercreate.sql file ([Link](#)) and save it to your preferred folder
- Download storyline.json file ([Link](#)) and save it to your preferred folder
- Use activitytrackercreate.sql file and open it in mySQL and click on run (This will create your schema)
- Open Eclipse IDE and open this extracted file in the IDE
- Inside Resources>application.properties file - update your mysql username, password
- Inside src>main>java>ActivityTracker>DAO>ConnManager, update your username and password.
- Inside src>main>java>ActivityTracker>Inserter update the file location to where storyline.json file is located
- Right click on the pom.xml file and click on maven>reload project
- Right click on inserter.java and click on Run 'Inserter.main()', this will insert all the data into your database (This will take a few minutes to finish)
- Right click on ActivityTrackerApplication and click on Run. This will start your server, now you can go to localhost:8080/ and use the webpages we created or use the api we have made.

POSTMAN TESTING INSTRUCTIONS

- All the collection frameworks for GET, POST, DELETE are present in postmantest.zip
- The JSON files for the POST requests are present in postrequest.zip
- Extract the JSON files of the collection frameworks and POST requests and save it in your preferred location/folder.

- Import the collection frameworks on the POSTMAN Console to begin testing the API endpoints. All the pre-request scripts, test scripts are included in the collection framework.
- Create a global variable named “url” and set its value to the URL of your machine along with port number in the format “machine_name:port_number” which is used in all the requests. Create global variables named as “date”, “year”, “month”, “day”, “value”, “value2”, “jsonbody”. Initialize “value” to “{{year}}-{{month}}-{{day}}” and “value2” to “{{{value}}}".
- The GET request for Activity Summary by specific date collection framework consumes the storyline.json file to show activity summary for all the dates. Run the collection by selecting the storyline.json file to get the test results.
- The GET request for all other endpoints is present in a separate collection framework. This collection does not consume any request body. Run the collection by setting the iteration value to get the test results.
- The POST request for all endpoints collection framework consumes a JSON file for each kind of API. Run the collection by selecting the desired endpoint and select the corresponding JSON file for the API. You can add more objects to the JSON file retaining the format so as to test the modularity of the APIs.
- The DELETE request is implemented for /records API in a collection framework. You can modify the {date} in the request to delete the specific record and associated data from the database.
- All the requests return a PASS if the API returns a success code in the response body otherwise the console will throw the respective error.