

For building and running the application below is needed:

- [JDK 1.8](#)
- [Maven 3](#)
- [Docker](#)

## Technology Stack Server - Backend

- [JDK](#)
- [Spring Boot](#)
- [Maven](#)
- [JSON Web Token](#)

## Docker Container Initialiser.

To create a docker container.

1. Create docker file
2. Create starter file to run the associated jar  
`java -jar /opt/abhinav/app/VMW/assignment-0.0.1-SNAPSHOT.jar`
3. Write **build.sh** in the VMWareAssignment folder.
4. Create file **run.sh** in the target folder and write the following  
`docker run --restart=always --name vmw-app -p 8080:8080 -d vmw:1.0.0`

Load the docker image **load <vmw.tar.gz**

## Testing API

## Testing with Postman Runner

### [PostmanCollection](#)

```
{"info":  
{"_postman_id":"772326e8-7f32-47a1-828f-2c1a16e16504","name":"VMWAssignment_PostmanCollection","schema":"https://  
schema.getpostman.com/json/collection/v2.0.0/  
collection.json"},"item":[{"name":"http://localhost:8080/
```

```
uploadFile","id":"e6f5772d-12ce-4bf0-8a0e-0800349c9d8b","request":
{"method":"POST","header":[],"body":{"mode":"formdata","formdata":
[{"key":"file","type":"file","src":"\\Users\\abhinav\\uploads\\
user.txt"}]},"url":"http:\\\\localhost:8080\\
uploadFile"},"response":[{}],{"name":"http:\\\\localhost:8080\\
getTaskStatus\\/2","id":"573c8d1a-beb5-46e9-
a2f9-0ddbf989dc5b","protocolProfileBehavior":
{"disableBodyPruning":true},"request":{"method":"GET","header":
[],"body":{"mode":"formdata","formdata":
[{"key":"","type":"file","src":[]]}],"url":"http:\\\\localhost:
8080\\getTaskStatus\\/2"},"response":[{}],{"name":"http:\\\\
localhost:8080\\api\\timer\\runbatchprocessor","id":"5cc47e9b-
fd91-4e82-9a16-bc0e644d124f","request":{"method":"POST","header":
[],"url":"http:\\\\localhost:8080\\api\\timer\\
runbatchprocessor"},"response":[[]]}}
```

## What is does ?

This service allows to upload sample file which can be processed at the server to upload the data in the database. It returns task id associated with the process. User later can query the same task id and get the status of the task.

### FileStorageService:

This service is used to upload the files on the server.

### TaskStatusService:

This service is used to create an entry in the task table with all the features related with it.

Quartz Scheduler:

**BatchSchedulerJob** is responsible for running the job at regular interval and query the task table to check if there is any new task id with running status “submitted”. It process the pending task and process those files.

Spring Batch

**SpringBatchConfig** class is responsible for setting the spring container with necessary information to execute the batch process to parse the file and upload the data in database.

Spring Batch Working

