# Justification for technology

I have used **SPRING BOOT** which is most popular and cutting edge technology in the current market scenario. Below are few pointers given .

1. Production readiness is about looking beyond functional requirements and ensuring your application can be properly managed and monitored in production
2. Health checks : provides actuator and metrics
3. Viewing application configuration - application properties, Environment variables etc.
4. Viewing and altering log configuration
5. Viewing application metrics - JVM, classloader, threading and garbage collection.
6. Audibility of key application events

These are only few advantages , apart from these are tons of advantages which we can utilize.

# Install & Run

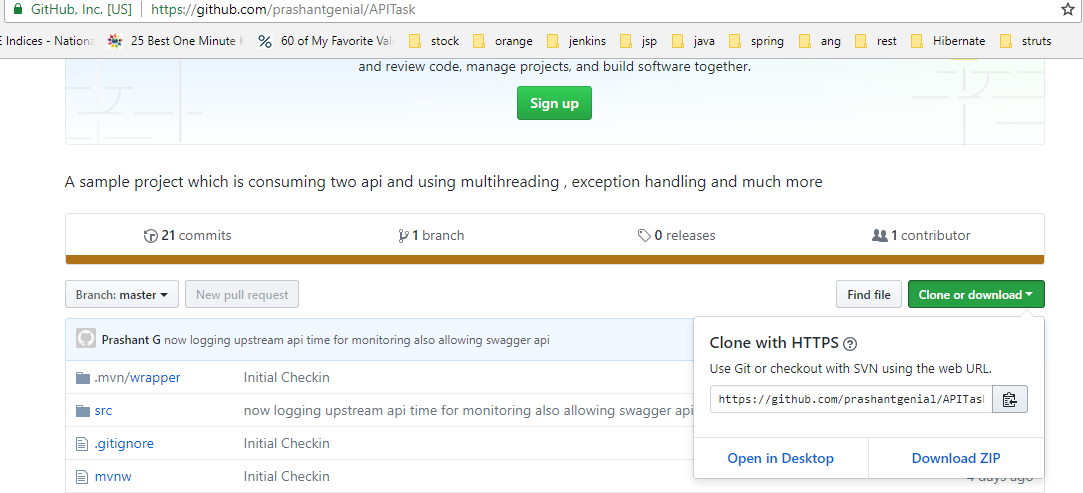
##### Clone the project :

either download the project by clicking clone/download button

OR

open git bash (if git is installed on your machine)

**git clone** [**https://github.com/prashantgenial/APITask**](https://github.com/prashantgenial/APITask)

****

2. open a command prompt and go to the root directory project to compile run below command  
**mvn clean package**

4. Once project compiled a jar file will be generated in the target folder. Lets run it  
**java -jar target/APITask-0.0.1-SNAPSHOT.jar**

OR

java -jar target/APITask-0.0.1-SNAPSHOT.jar --spring.profiles.active=dev

6. open postman and run below URL

http://localhost:9191/media?input=road

Response recieved :

{

"timestamp": "2018-08-16T09:40:51.013+0000",

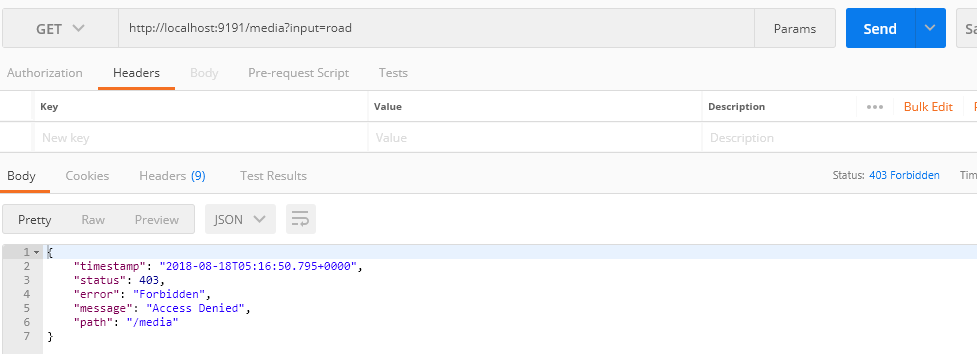
"status": 403,

"error": "Forbidden",

"message": "Access Denied",

"path": "/media"

}



**this service is accessible by a secure token only, so first lets generate a token.**

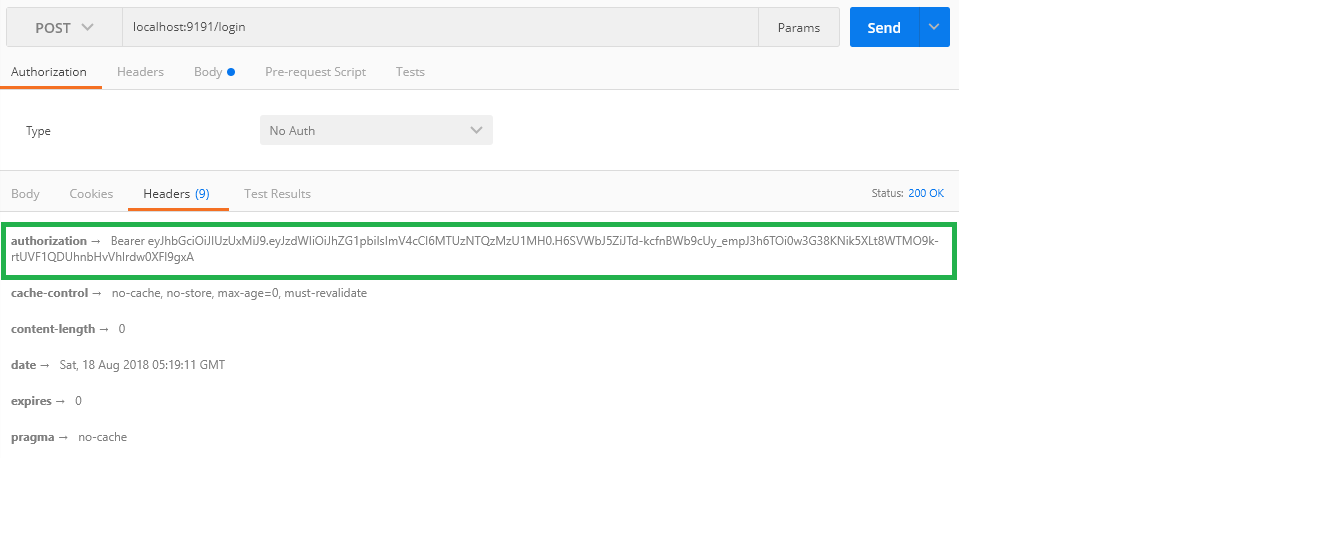
**I have used JWT Token security and token generated for 10 days as of now, this is configurable**

**Lets Generate a secure token.**

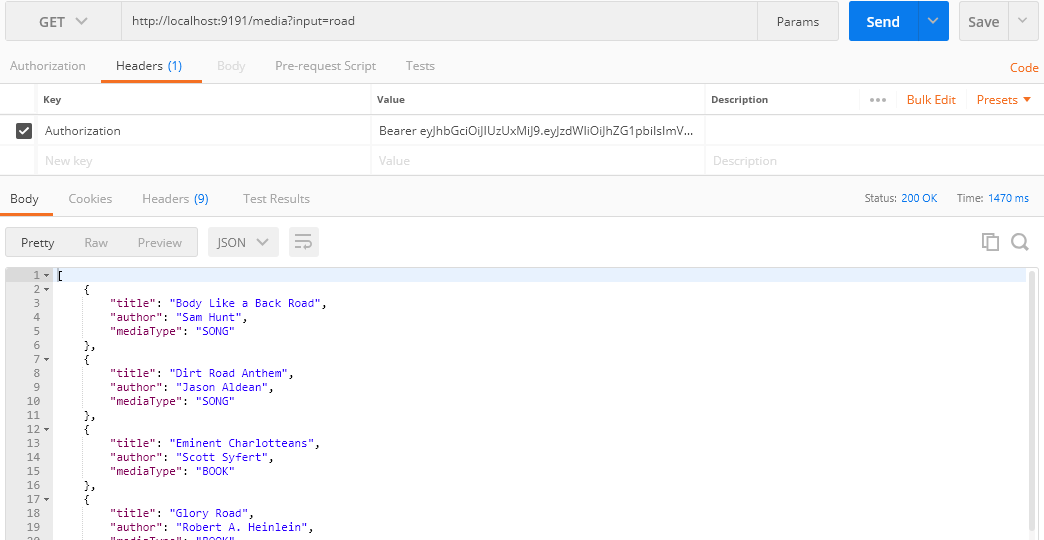
oepn postman and create a new POST Request . URL "localhost:9191/login" and Body

{"username":"admin","password":"password"}

this will generate a authorization token in header section



Now used this token in the request as a header . See below we are now getting media records.



As of now we are getting total 10 records. In which 5 records are of Books and 5 are of Tracks.

Both the records are getting fetched concurrently from below API’s.

***For albums use the iTunes API:***

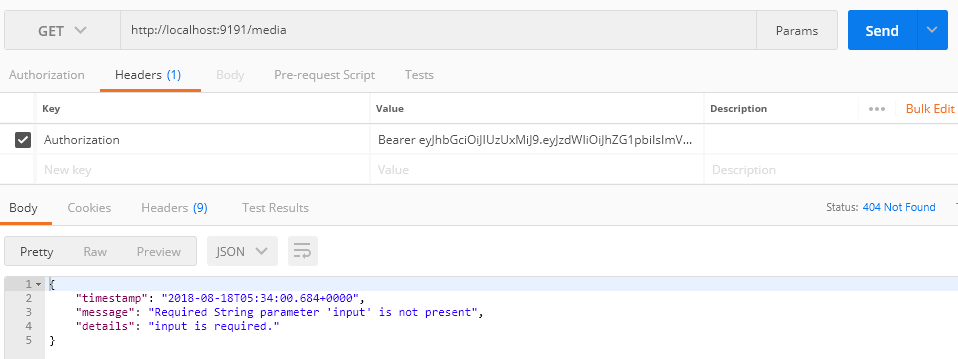
<https://affiliate.itunes.apple.com/resources/documentation/itunes-store-web-service-search-api/#searching>

***For books: use Google Books API:*** <https://developers.google.com/books/docs/v1/reference/volumes/list>

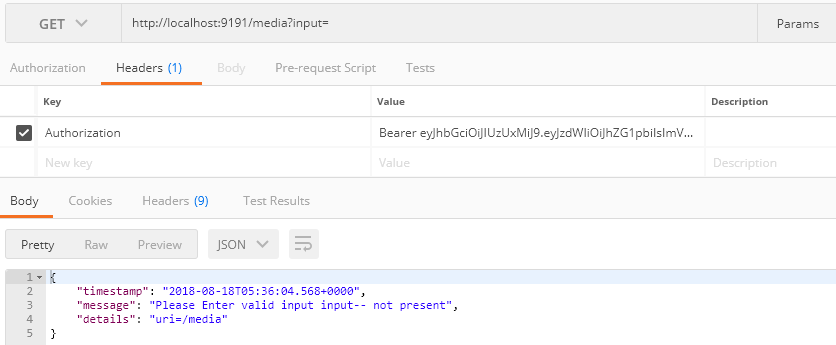
***It was asked to get all the records in 3 seconds , Performance that I have recorded in my implementation , initially it is taking around 1.5 seconds however in subsequent requests it is taking round .5 miliseconds to 1 seconds.***

***Added Exception Handling***

If user do not pass the input then below error is visible to User

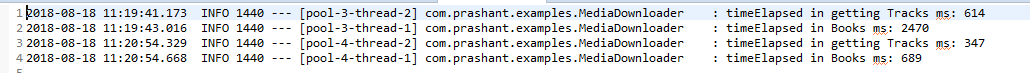


If Empty input string passed then also user get the error message



**Logging for Monitoring**

As of now both the upstream apis access time being logged . Later on ELK can be integrated for better monitoring.



**To Access the Health Status of the API:**

http://localhost:9191/actuator/health

**To access api documentation**<http://localhost:9191/swagger-ui.html#/media-controller>