**Project Release Document**

**Introduction:**

I have used spring.io to create SpringBoot application setup and implemented business logic as per project requirement.

**Database:**

Database- : Derby

Query:

**CREATE** **TABLE** app.USER\_INFO (

EMAIL **varchar**(50) **PRIMARY** **KEY** ,

PASSWORD **varchar**(45) **NOT** **NULL**,

CREATE\_TIME **DATE**,

LAST\_UPDATE\_TIME **DATE**

)

**insert** **into** app.USER\_INFO(EMAIL,PASSWORD,CREATE\_TIME,LAST\_UPDATE\_TIME) **values**('raj@gmail.com', 'pass',current **date**,current **date**)

**SELECT** \* **FROM** APP.USER\_INFO;

**CREATE** **TABLE** app.NOTES (

TITLE **varchar**(30) **PRIMARY** **KEY** ,

NOTE **varchar**(1000),

EMAIL **varchar**(50),

CREATE\_TIME **DATE**,

LAST\_UPDATE\_TIME **DATE**,

**CONSTRAINT** fk\_USERINFO **FOREIGN** **KEY** (EMAIL) **REFERENCES** APP.USER\_INFO (EMAIL)

)

**SELECT** \* **FROM** app.NOTES

**API’S DETAILS:**

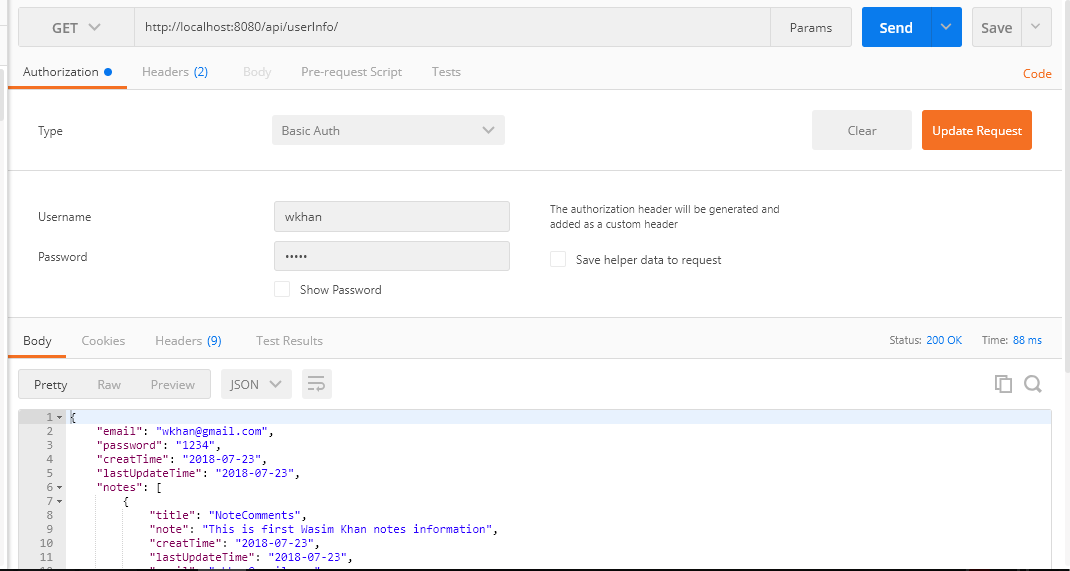
CRUD API:

User Note information as per request:

<http://localhost:8080/api/userInfo/>

Only valid user can get information

1. GetUserDetails as per user:



API2- SAVE NOTE INFORMATION:

Post Method- <http://localhost:8080/api/notes>

Input JSON Format-- {

"email": "khan@gmail.com",

"title": "KhanComment",

"note": "This is first Khan notes information",

"creatTime": "2018-07-23",

"lastUpdateTime": "2018-07-23"

}

API3- Update NOTE INFORMATION:

Put Method- <http://localhost:8080/api/updateNotes>

Input JSON Format- {

"email": "Khan@gmail.com",

"title": "KhanComment",

"note": "This is first Raj notes information updated one",

"creatTime": "2018-07-23",

"lastUpdateTime": "2018-07-23"

}

API2- Delete NOTE INFORMATION:

Delete Method - [http://localhost:8080/api/deleteNote/{noteId}](http://localhost:8080/api/deleteNote/%7bnoteId%7d)