

## Practice Module 6

Tested on both the systems below-

Macbook Pro (13 inch, 2017, Two Thunderbolt 3 ports)

Operating system: mac Catalina (10.15.7)

Processor: 2.3 GHz Dual-Core Intel Core i5

Memory 8GB 2133 MHz LPDDR3

Graphics Intel Iris Plus Graphics 540 1536 MB

Safari Version 14.0

Terminal 2.1.0

Operating system: Windows 10 home (20H2) – created on this system

OS build:19042.662

Experience: Windows feature experience pack 120.2212.551.0

HP ENVY X360 convertible 15-DS1XXX

Processor AMD Rizen 5 4500U with Radeon graphics 2.38GHz

Memory 16GB

Browser version: Microsoft edge (87.0.664.57)

Tested using windows CMD terminal

**TypeScript – 4.1.2**

**Node.js version (v14.15.1)**

### Set up project:

1. Packet.json file has all these dependencies listed below.

#### Dependencies:

```
"dependencies": {
  "emailjs-imap-client": "^3.1.0",
  "express": "^4.17.1",
  "mailparser": "^3.0.1",
  "nedb": "^1.8.0",
  "nodemailer": "^6.4.16"
}
"devDependencies": {
  "@types/express": "^4.17.9",
  "@types/mailparser": "^3.0.0",
  "@types/nedb": "^1.8.11",
  "@types/node": "^14.14.10",
  "@types/nodemailer": "^6.4.0",
  "nodemon": "^2.0.6",
  "typescript": "^4.1.2"
}
```

2. To install these dependencies, navigate to the server folder in terminal and run: **npm install**
3. Update the username and password, host and port for both SMTP and IMAP for your personal email application such as gmail, yahoo, Hotmail etc.
4. To run the server, go into server folder and type on console: **npm run dev**
5. Code is in server/src and executable in server/dist
6. Main code is in src/ main.ts and main executable is in dist/main.js

To test this, download insomnia core application from <https://insomnia.rest/download>  
we used version (2020.5.1)

### Expected Functionalities

1. Start this server with: **npm run dev** in a terminal and open insomnia core.
2. We will make different requests to the server using the insomnia client. Insomnia creates new request, press create button using get method and put address localhost/mailboxes, click send
3. This should fetch list of inboxes in your email and show on the right.
4. Create another request with get method to display a mailbox for e.g., if INBOX is one of the returned mailbox in the previous step then put the address besides the get method as localhost/mailboxes/inbox. Click send. All the emails from the mailbox of INBOX should be displayed in the preview window
5. Each email should have an id, date, from and subject.
6. To get a specific email body, create a new get request in insomnia with the address localhost/messages/INBOX/<id>, replace <id> with the id of that desired message to read. Click send.
7. To delete a message, create a new request with delete method in insomnia and provide the address localhost/messages/INBOX/<id> and replace <id> with the id of the method which you want to delete. Click send and a confirmation that message is deleted should be visible in the preview window.
8. To send a message, create a new post request in insomnia and give the address localhost/messages for the body of the message select json from the dropdown menu and provide the following details:  

```
{  
  "to": "receiver@domain.com",  
  "from": "sender@domain.com",  
  "subject": "test subject",  
  "text": "whatever you want to write"  
}
```

  
Under header menu, make sure a new header content type with value application/json. Click send and message sent should be displayed.
9. To get list of contacts, create a new get request with the address localhost/contacts and all the contact should be displayed.
10. To add a new contact. Create a new post request with the address localhost/contacts. In the body select json and add the following texts  

```
{  
  "Name": "name of friend",  
  "email": "name@domain.com"  
}
```

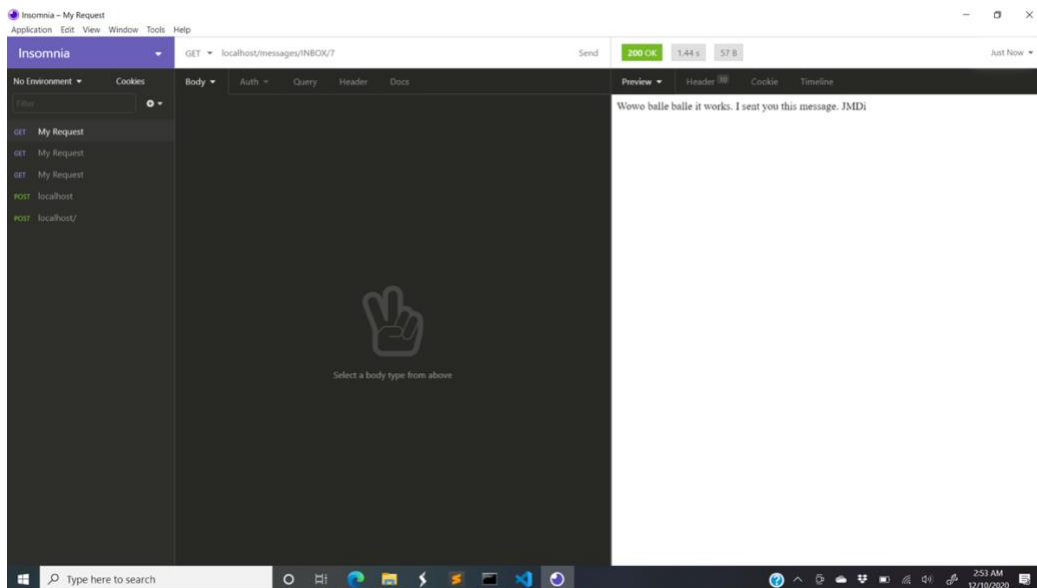
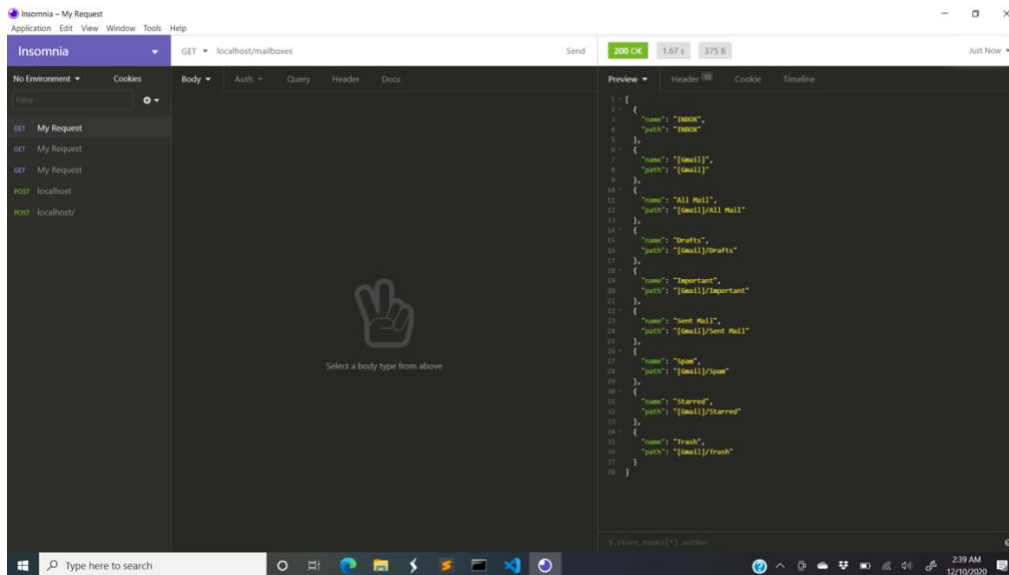
  
Make sure in header menu new header with content type = applications/json is added. Click send and the contact added contact should be displayed in the preview window.
11. To delete a contact first find their Id using the list of contacts then create a new delete request with the address localhost/contacts/<id> and replace <id> with the <id> of the contact to be deleted. Click delete and confirmation of deleting the contact should be displayed in the preview window.
12. To update, create a new request select put method and provide the url localhost/contacts with method put and body json, now find the id of the contact you want to update by looking at the list of contacts. Then in the json body you can put 

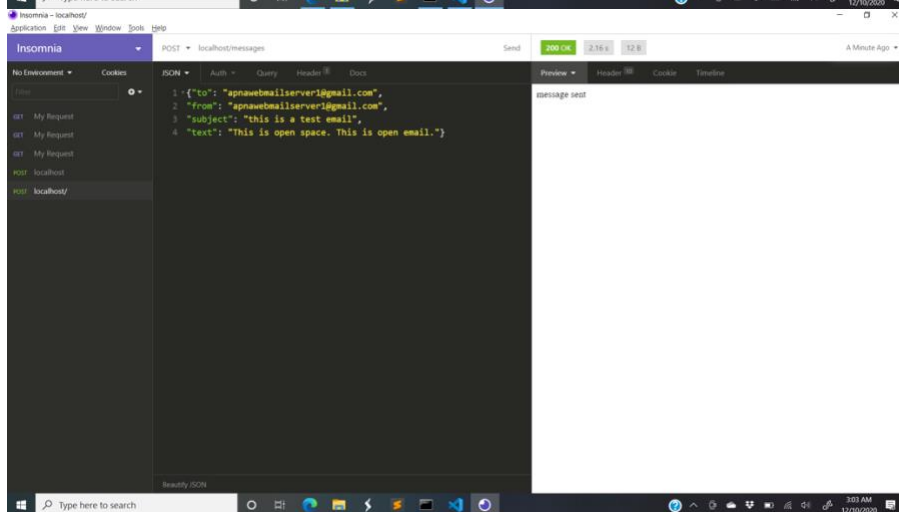
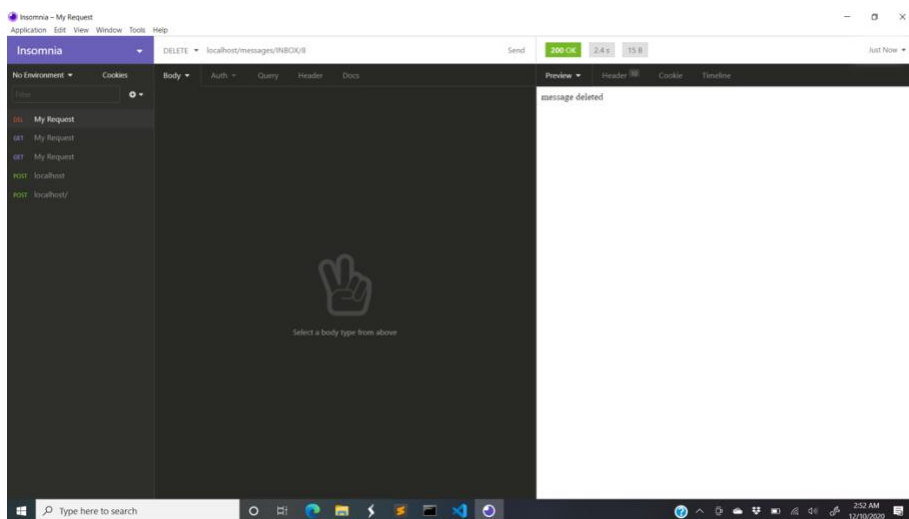
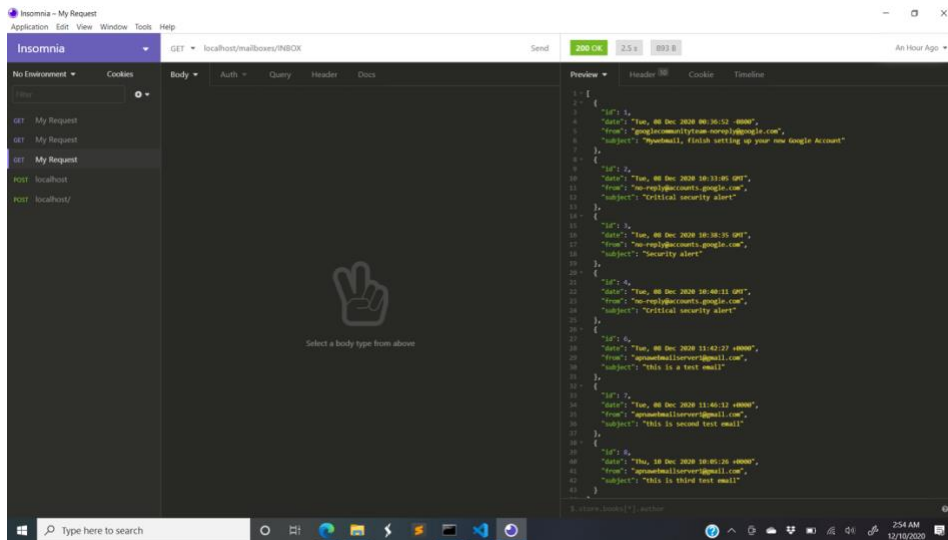
```
{  
  "_id": "<id>",  
  "name": "<name>",  
  "email": "<email>"  
}
```

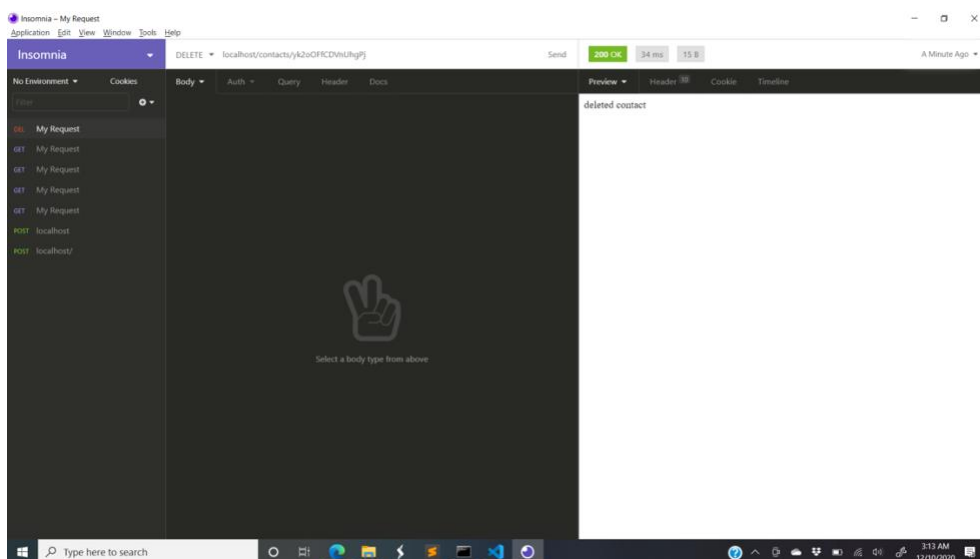
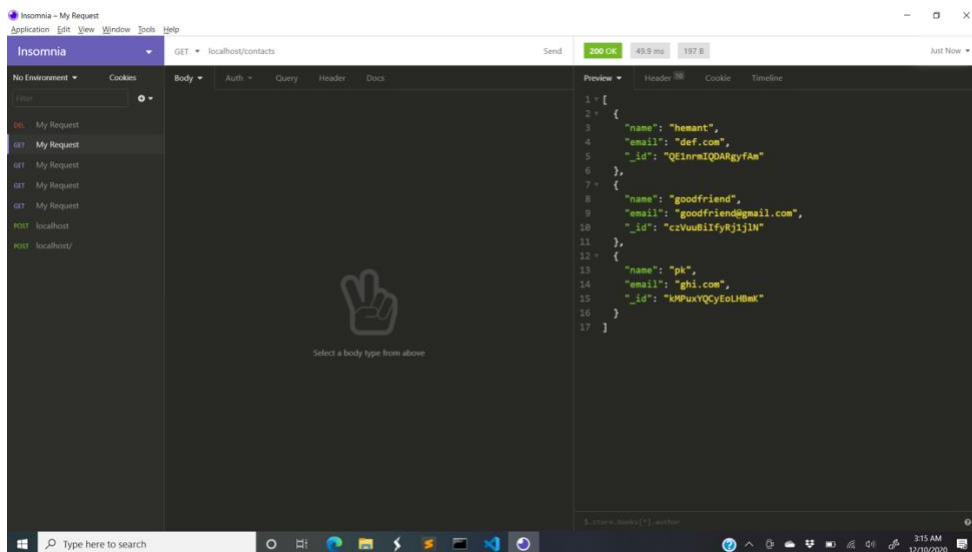
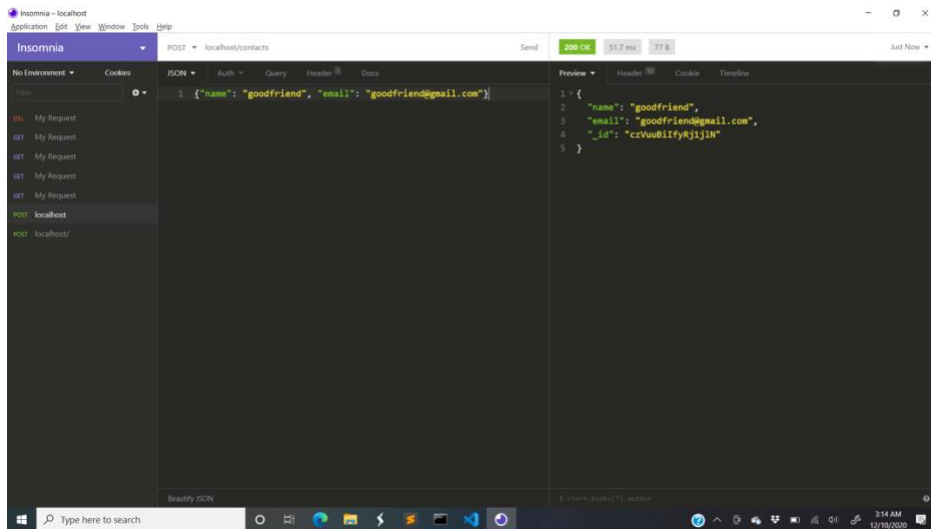
  
(respected figures of above points are attached below)

### Benefit of REST

1. REST is a set of best practices and patterns to create a webpage API in a standard way and the benefit of REST is in providing simplicity and reduction in overhead. So, we use REST in the following useful ways. We use URL's to refer to nouns such as contacts, mailboxes or message.
2. We use appropriate methods for the http request for e.g., we use get method to get the list of mailboxes, list of messages, list of contacts or specific msg. we use post method to create a new email or to create a new contact. We use the delete to delete a contact or an email.
3. For get or delete the information about the request is specified as part of the URL and not the body whereas for post is specified in the body.
4. Another benefit of the REST is that we also return http status codes such as 200 – OK or 404 – not found which help us debug our program.







**Additional function:**  
It updates the contact

