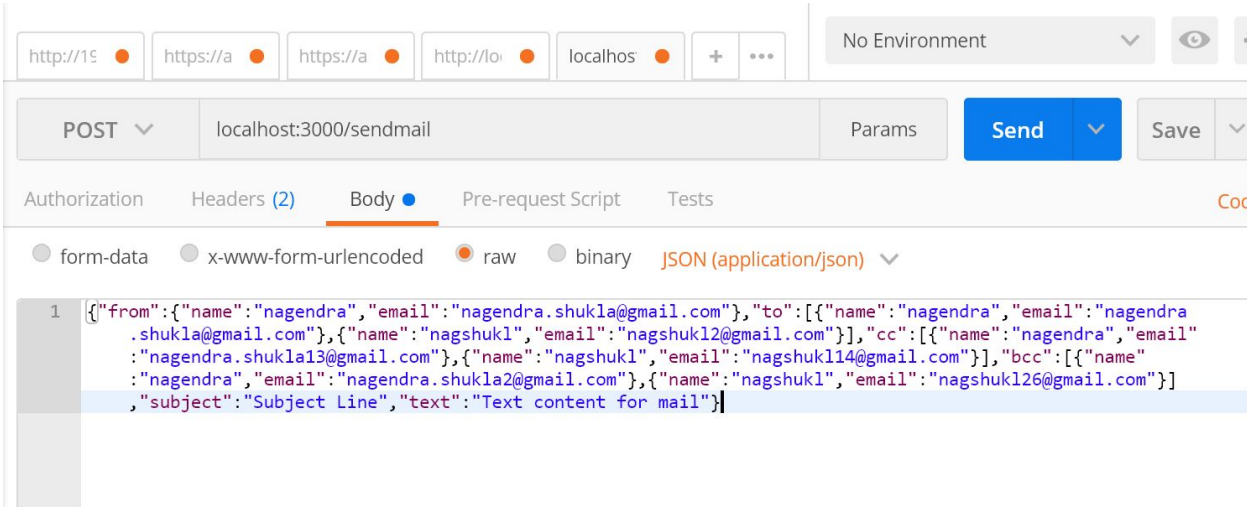
Email Sender API.

1. **In Application**, no external package is used. Dependencies section is empty in package.json. While dev-dependencies has packages related to testing framework.
2. **As per requirement that API should failover capability**
   In the solution provided, I have implemented in a way that, if due to any reason a mail provider is not able to send mail, the System will automatically try to use other listed provider.
   In order to add a new provider we need to update providers list in **Provider.js** file. This file already has 4 providers listed. The first 2 provider are fake provider for testing purposes, while the third and fourth are MailGun and SandGrid respectively.

3. **Validation Rules (listing important once)**
   a. From, to, subject and text inputs are mandatory.
   b. Validates for duplicate emailId's
   c. Validates for valid email ids.
   d. Names are not mandatory; If missing it will use some hard coded names.
4. **Unit testing Implemented,** which has full endpoint coverage but may not be 100% code coverage.
5. **Front-end Application,** I am not sure, if implementing web client for this API, was the part of assignment or not. But that's not included in this delivery considering the fact that i am already submitting another separate assignment for frontend.
   If front application is need - please update me, so that I can implement and send it over quickly.
6. **Environment and installation commands.**
   a. Developed using node 10.15.3
   b. Please unzip this package to some location and move to root directory and execute:
   c. $ npm install to install required dev-dependencies.
   d. $ npm start to launch api application on port 3000. This command sets the NODE-ENV to dev; hence in console we can see logs.
   e. $ npm test to execute unit test cases associated.
7. **Testing api,** to test this api please use postman or any other tool. Api endpoint is **/sendmail**
   A set of valid and invalid input data can be found in the readMe.txt file attached in zipped project.
   Valid JSON:
   {"from":{"name":"nagendra","email":"nagendra.shukla@gmail.com"},"to":[{"name":"nagendra","email":"nagendra.shukla@gmail.com"},{"name":"nagshukl","email":"nagshukl2@gmail.com"}],"cc":[{"name":"nagendra","email":"nagendra.shukla13@gmail.com"},{"name":"nagshukl","email":"nagshukl14@gmail.com"}],"bcc":[{"name":"nagendra","email":"nagendra.shukla2@gmail.com"},{"name":"nagshukl","email":"nagshukl26@gmail.com"}],"subject":"Subject form mail","text":"Text for email"}
   Invalid JSON:

{"from":{"name":"nagendra","email":"nagendra.shukla"},"to":[{"name":"nagendra","email":"nagendra.shukla@gmail.com"},{"name":"nagshukl","email":"nagshukl.gmail.com"}],"cc":[{"name":"nagendra","email":"nagendra.shukla1gmail.com"},{"name":"nagshukl","email":"nagshukl1@gmail.com"}],"bcc":[{"name":"nagendra","email":"nagendra.shukla@gmail.com"},{"name":"nagshukl","email":"nagshukl2gmail.com"}],"subject":"Subject form mail","text":"Text for email"}

8. **Screenshots**
   a. **Postman screen shot**



   b. **Unit testing results**

```
> emailmanager@1.0.0 test C:\jsr\nagShukl_Git\assignments\siteminder\jsrMail
SendAPI
> mocha --timeout 10000



  /POST sendmail
(node:15252) [DEP0005] DeprecationWarning: Buffer() is deprecated due to sec
urity and usability issues. Please use the Buffer.alloc(), Buffer.allocUnsaf
e(), or Buffer.from() methods instead.
    √ it should send mail (response 200) with valid data and content-type (6
309ms)
    √ it should send mail (response 200) without CC list (5573ms)
    √ it should send mail (response 200) without bCC list (5570ms)
    √ it should send mail (response 200) without cc & bCC list (5578ms)
    √ it should respond 400, when data is valid but invalid content-type
    √ it should respond 400, when body has no Subject
    √ it should respond 400, when body has blank Subject
    √ it should respond 400, when body has no text
    √ it should respond 400, when body has blank text
    √ it should respond 400, when body has no from field
    √ it should respond 400, when body has from field but with invalid email
    √ it should respond 400, when from field is with invalid email
    √ it should respond 400 and error message, when To field is not availabl
e
    √ it should respond 400 and error message, when To email is not valid
    √ it should respond 400 and error message, when CC email is not valid
    √ it should respond 400 and error message, when bCC email is not valid
    √ it should respond 400 and error message, there is a duplicate email an
ywhere
    √ it should respond 400 and error message, there is a duplicate email in
 to list
    √ it should respond 400 and error message, there is a duplicate email in
 cc list
    √ it should not send mail without a body

  /GET sendmail
    √ it should response with 405 error

  /GET for any route
    √ it should response with 405 error

  /POST for route other then /sendmail
    √ it should response with 404 error


  23 passing (23s)
```