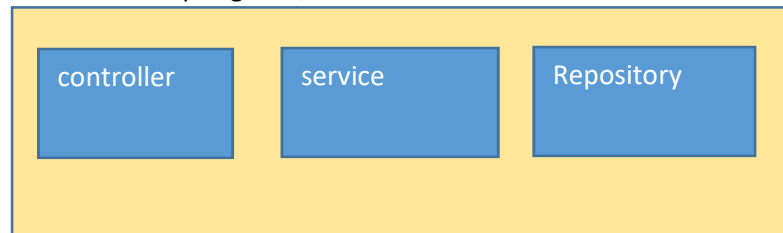


email application

springboot, RESTAPI



DB  
to save email details

sendMail()

email-app

s1



auth-app

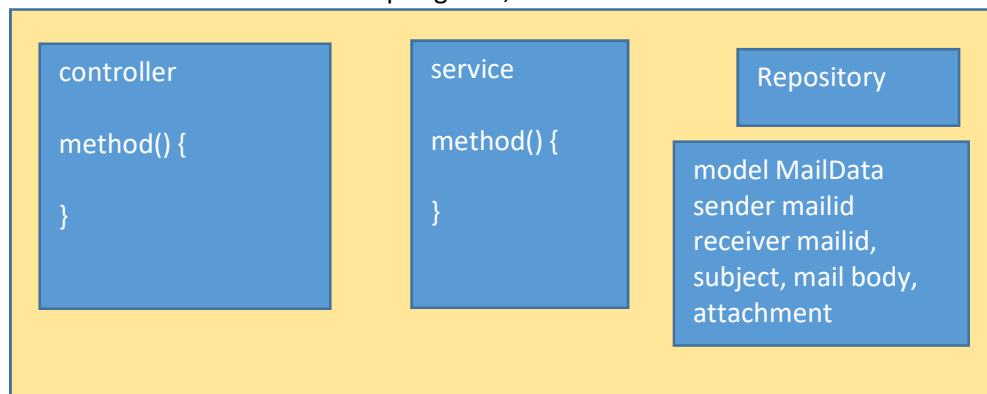
register-user(success: mail to be sent)  
login-check  
recover-password

product-app

new product added to user

email application

springboot, RESTAPI



DB  
to save email details

Note: sender email id is fixed

create one dummy email account

do necc changes in security settings

[abc@gmail.com](mailto:abc@gmail.com)

12345

request handler method

@PostMapping("/send-mail")

public ResponseEntity<?> sendMail(@RequestBody MailData mailData)

{

}

arun

[arun@gmail.com](mailto:arun@gmail.com)

registration done

thankyou .....

## MailSenderServiceImpl

```
public boolean sendMail(MailData mailData)
{
    sout(mailData);
    // save mailData to DB thru Repository
    // mail can be send according to mailData using springboot mail services
}
```

The screenshot displays a REST client interface. At the top, a POST request is configured for the URL `http://localhost:85500/mail-app-v1/send-mail`. The 'Body' tab is selected, showing a JSON payload: `{ "receiver": "vigneshkumars3171@gmail.com", "messageBody": "thankyou1", "subject": "registration1", "attachment": null }`. Below the request, the 'Body' tab of the response is selected, showing the text `Mail Sent to vigneshkumars3171@gmail.com`.

POST `http://localhost:85500/mail-app-v1/send-mail`

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ☐

```
1 {
2   "receiver": "vigneshkumars3171@gmail.com",
3   "messageBody": "thankyou1",
4   "subject": "registration1",
5   "attachment": null
6 }
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview Visualize Text ☐

```
1 Mail Sent to vigneshkumars3171@gmail.com
```