

# Book View

This application allows logged in users to view the book database and moderate information about their account, and users with administrator rights to moderate the list of books and users.

1. Download files from <https://github.com/baqardoS/backend/tree/project> and extract from zip.
2. Launch the Docker application
3. Open console and go to extracted folder
4. Enter the command: `docker-compose up`
5. Wait while the environment is installed
6. Then go to application documentation: <https://documenter.getpostman.com/view/16690213/Uyxbpp6a>
7. Click “Run in Postman” button to clone endpoints to your Postman

The screenshot shows the Postman API client interface. On the left, there is a sidebar with a 'BOOKS VIEW' menu. The main panel displays the 'Books View' documentation, which includes a 'Users' section with 'Users related endpoints'. Two endpoints are listed: 'GET Get User' and 'GET Get All Users'. The 'GET Get User' endpoint has a URL template: `{{URL}}/users/api/v1/users/6293c8782f4d8f0014214a00`. The 'GET Get All Users' endpoint has a URL template: `{{URL}}/users/api/v1/users`. Below the URL templates, there is a 'PARAMS' section with a table of parameters:

PARAMS	
page	2
limit	1
fields	role, name
sort	name
role	user
name	user

On the right side of the interface, there is a 'Run in Postman' button. A red arrow points to this button. Below the button, there is a section for 'Example Request' and 'Get User'/'Get All Users' endpoints. The 'Get User' endpoint has a request body with the following code:

```
var requestOptions = {
  method: 'GET',
  redirect: 'follow'
};

fetch("{{URL}}/users/api/v1/users/6293c8782f4d8f0014214a00", requestOptions)
  .then(response => response.text())
  .then(result => console.log(result))
  .catch(error => console.log('error', error));
```

The 'Get All Users' endpoint has a request body with the following code:

```
var requestOptions = {
  method: 'GET',
  redirect: 'follow'
};

fetch("{{URL}}/users/api/v1/users", requestOptions)
  .then(response => response.text())
  .then(result => console.log(result))
  .catch(error => console.log('error', error));
```

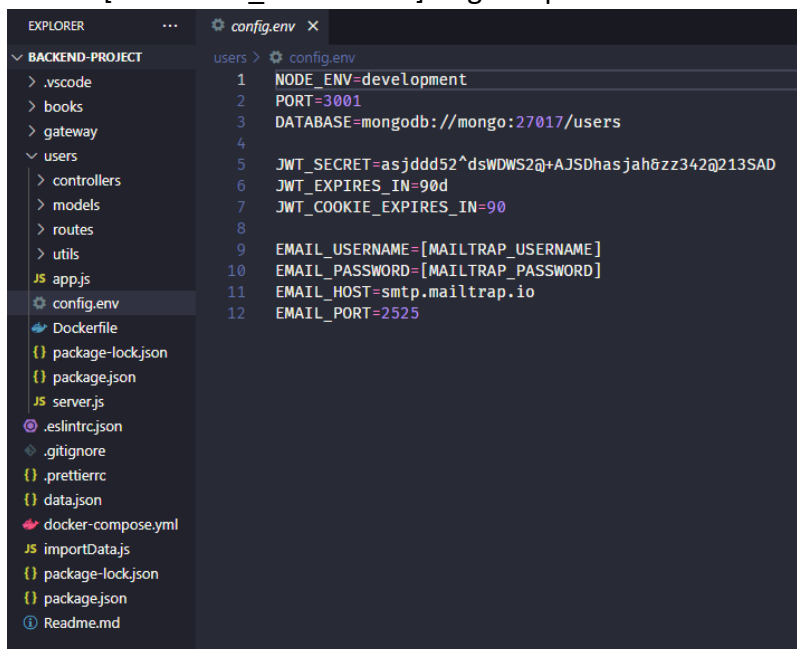
8. In repository files will be screenshot of documentation
9. Add new environment to the Postman with:
  - a. “URL” variable equal to: `“http://localhost:3000”`
  - b. “password” variable equal to any eight-character string

New Environment			Edit
VARIABLE	INITIAL VALUE	CURRENT VALUE	
URL	http://localhost:3000	http://localhost:3000	
password	zaq1@WSX	zaq1@WSX	

10. Now you can test application on your local machine
11. First you need to create new account using “Sign Up” endpoint
12. And now you have normal user permissions, to elevate permissions use “Become Administrator (only for development version)” endpoint
13. Now you can administrate data in databases
14. To check “Forgot Password” endpoint you need to create a MailTrap account on page: <https://mailtrap.io/> then go to <https://mailtrap.io/inboxes> click in “My Inbox”

15. Then get data for connecting app with MailTrap

16. Modify data accordingly in config.env file in “backend-project\users” by changing:
- [MAILTRAP\_USERNAME] to given user
  - [MAILTRAP\_PASSWORD] to given password

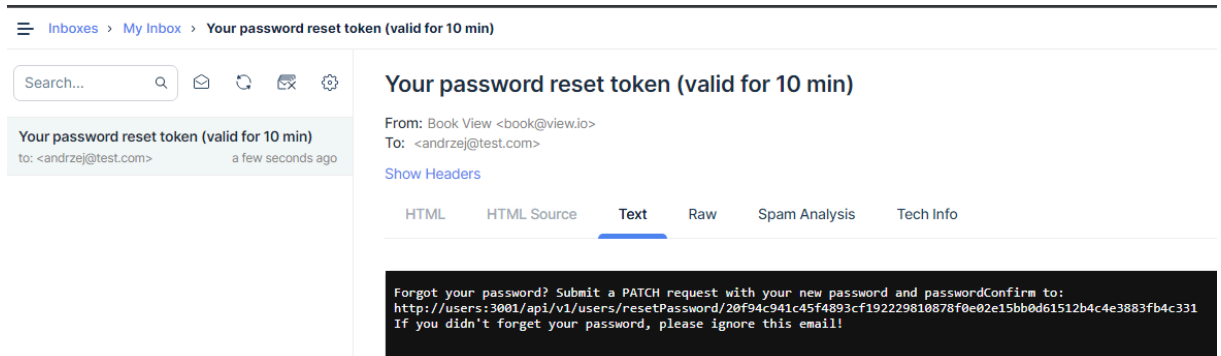


The screenshot shows the VS Code Explorer on the left with the 'BACKEND-PROJECT' folder expanded, showing subfolders like .vscode, books, gateway, users, controllers, models, routes, utils, app.js, config.env, Dockerfile, package-lock.json, package.json, server.js, .eslintrc.json, .gitignore, .prettierrc, data.json, docker-compose.yml, importData.js, package-lock.json, package.json, and Readme.md. The 'config.env' file is selected and its content is displayed in the main editor. The file contains environment variables for development, including database connection, JWT secrets, and email configuration.

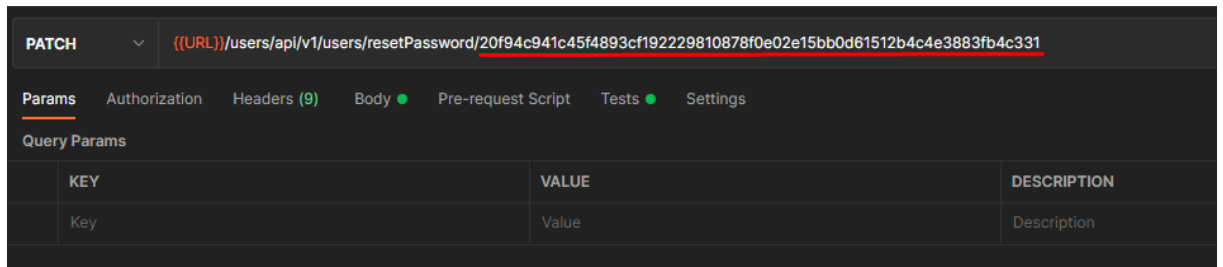
```
1 NODE_ENV=development
2 PORT=3001
3 DATABASE=mongodb://mongo:27017/users
4
5 JWT_SECRET=asjddd52^dsWDWS2@+AJSDhasjah8zz342@213SAD
6 JWT_EXPIRES_IN=90d
7 JWT_COOKIE_EXPIRES_IN=90
8
9 EMAIL_USERNAME=[MAILTRAP_USERNAME]
10 EMAIL_PASSWORD=[MAILTRAP_PASSWORD]
11 EMAIL_HOST=smtp.mailtrap.io
12 EMAIL_PORT=2525
```

17. Now stop application and use again command in main app directory: **docker-compose up –build**

18. Now you should be able to use “Forgot Password” endpoint and see mails in your MailTrap mailbox



19. Now to reset your password you need to send new “password” and “passwordConfirm” to “Reset Password” endpoint and paste into link the last part of the link from the email



20. To check production version you need add new environment to Postman with:
- a. “URL” variable equal to: “https://book-view-gateway.herokuapp.com”
  - b. “password” variable equal to “zaq1@WSX”
21. There are 2 accounts [user@test.com](mailto:user@test.com) and [admin@test.com](mailto:admin@test.com) both have password “zaq1@WSX”