TO-DO-API_BACKEND DOCUMENTATION USING POSTMAN

1) Check a server is properly running using "GET" Method.

=

1. Open **Postman**
2. Create a **New Request**
3. Select HTTP Method → **GET**

□ Enter the Server URL

In the URL field, enter:

```
http://localhost:5000/
```

□ Important:

- Use **http**, not https
- Use the correct port (**5000**)

□ Send the Request

Click the **Send** button.

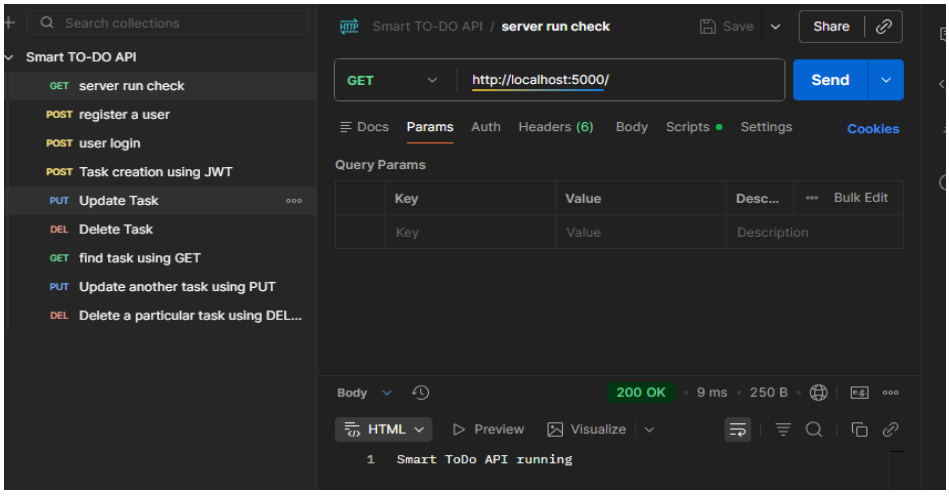
✓Check the Response

Expected Response:

```
Smart ToDo API running
```

Status Code:

```
200 OK
```



2) Register a User Using “Post” Method.

1. Launch **Postman**
2. Click **New → HTTP Request**
3. Select HTTP Method → **POST**

□ Enter the Registration API URL

In the URL field, enter:

```
http://localhost:5000/api/auth/register
```

□ Set Request Headers

Go to the **Headers** tab and add:

Key	Value
Content-Type	application/json

□ Add Request Body

1. Click the **Body** tab
2. Select **raw**
3. Choose **JSON** from the dropdown
4. Enter the following data:

```
{
  "name": "Ranjan Das",
  "email": "Ranjan@example.com",
  "password": "123457"
}
```

□ Send the Request

Click the **Send** button.

✅Check the Response

Successful Response:

{

"message": "User created",

"user": {

"id": "694f85046b6d20e0c06c1784",

"name": "Ranjan Das",

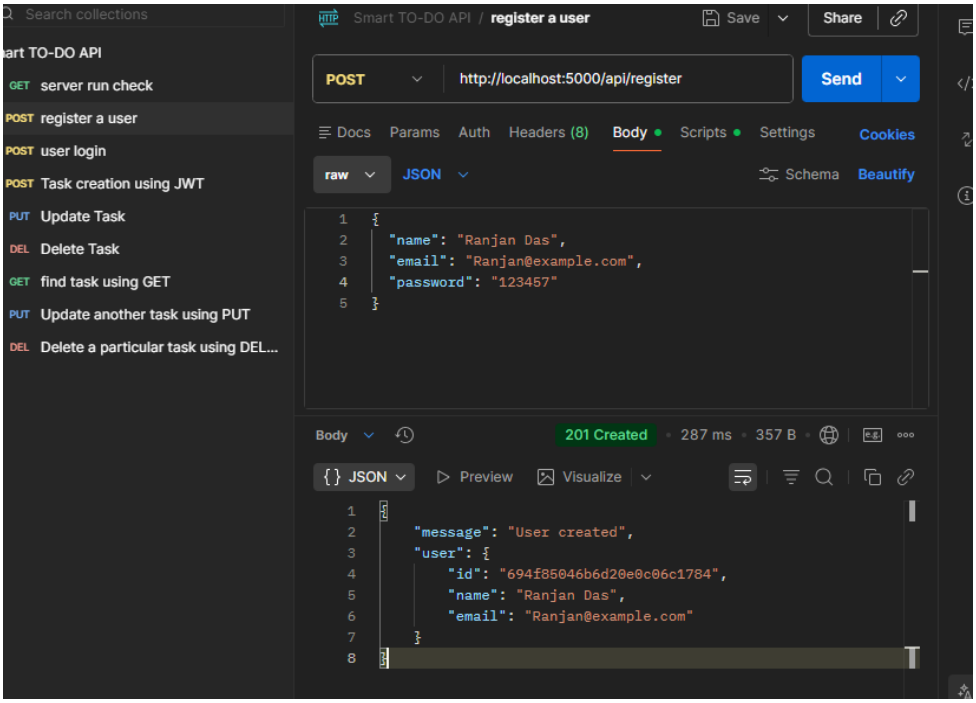
"email": "Ranjan@example.com"

}

}

Status Code:

`201 Created`



3) Login a User Using "Post" Method.

1. Open **Postman**
2. Click **New → HTTP Request**
3. Select HTTP Method → **POST**

□ Enter Login API URL

`http://localhost:5000/api/auth/login`

□ Set Request Headers

Go to **Headers** tab and add:

Key	Value
Content-Type	application/json

□ Add Request Body

1. Go to **Body**
2. Select **raw**
3. Choose **JSON**
4. Enter login credentials:

```
{
  "email": "john@example.com",
  "password": "123456"
}
```

□ Send the Request

Click **Send**.

✅Check the Response

Successful Response:

```
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "user": {
    "id": "694eac8a34fbd2cab2a32ef3",
    "name": "John Doe",
    "email": "john@example.com"
  }
}
```

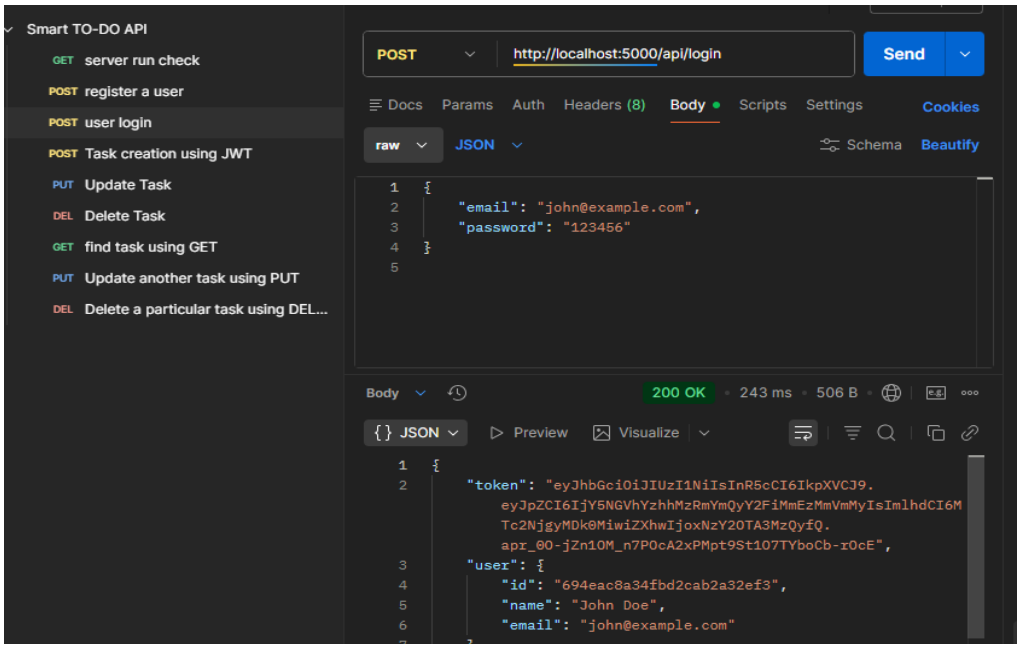
Status Code:

```
200 OK
```

Important: Save the JWT Token

Copy the token value.
You will use it as **Bearer Token** for:

- Create Task
- Update Task
- Delete Task



4) Task creation using JWT , “Post” Method.

Ensure User Is Logged In

- Login using the **POST /api/auth/login** API
- Copy the **JWT token** from the response

Example:

```
"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
```

□ Open Postman

1. Open **Postman**
2. Create a **New HTTP Request**
3. Select HTTP Method → **POST**

□ Enter Task Creation API URL

```
http://localhost:5000/api/tasks
```

□ Set Authorization (JWT)

1. Go to **Authorization** tab
2. Select **Type** → `Bearer Token`
3. Paste the **JWT token** (without quotes)

✔Postman will automatically set the header:

```
Authorization: Bearer <JWT_TOKEN>
```

□ Set Request Headers

Key	Value
Content-Type	application/json

□ Add Request Body

1. Go to **Body**
2. Select **raw**
3. Choose **JSON**
4. Enter task details:

```
{
  "title": "Finish Smart ToDo API project",
  "completed": false
}
```

□ Send the Request

Click **Send**.

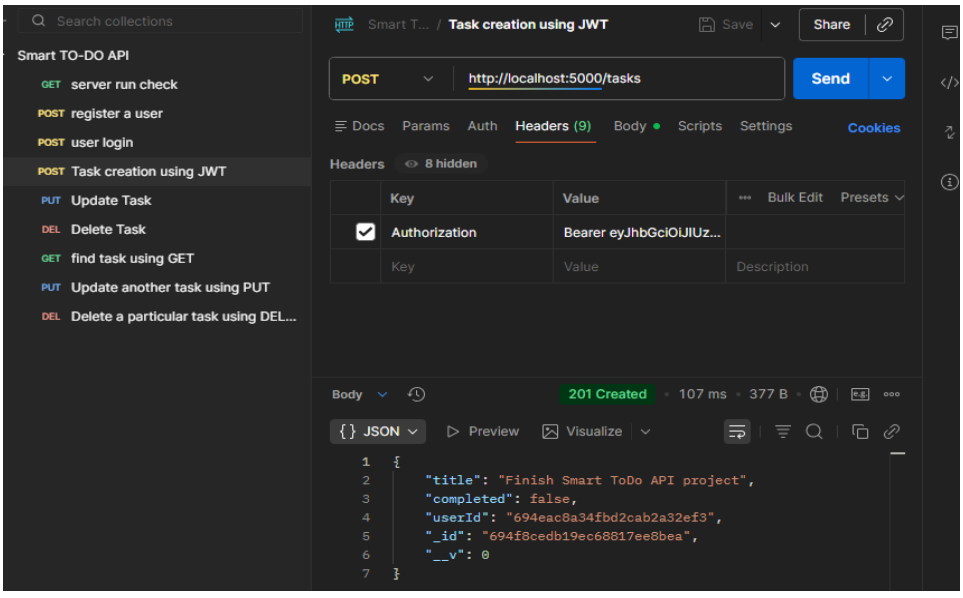
✓Check the Response

Successful Response:

```
{
  "_id": "694eb99da964b59aa33ab7cb",
  "title": "Finish Smart ToDo API project",
  "completed": false,
  "userId": "694eac8a34fbd2cab2a32ef3",
  "__v": 0
}
```

Status Code:

```
201 Created
```



5) update task using “PUT” Method.

= Login and Get JWT Token

1. Login using:

```
POST /api/auth/login
```

2. Copy the **JWT token** from the response.

□ Get Task ID

1. Send a **GET request**:

```
GET http://localhost:5000/api/tasks
```

2. Copy the `_id` of the task you want to update.

Example:

```
"_id": "694eb99da964b59aa33ab7cb"
```

□ Open Postman

1. Create a **New Request**
2. Select HTTP Method → **PUT**

□ Enter Update Task API URL

Replace `<taskId>` with the actual task ID:

```
http://localhost:5000/api/tasks/694eb99da964b59aa33ab7cb
```

□ Set Authorization (JWT)

1. Go to **Authorization** tab
2. Select **Type** → `Bearer Token`
3. Paste the JWT token

□ Set Request Headers

Key	Value
Content-Type	application/json

□ Add Request Body

Go to **Body → raw → JSON** and enter updated fields:

```
{
  "title": "Finish Smart ToDo API project (Updated)",
  "completed": true
}
```

You can update:

- `title`
- `completed`

□ Send the Request

Click **Send**.

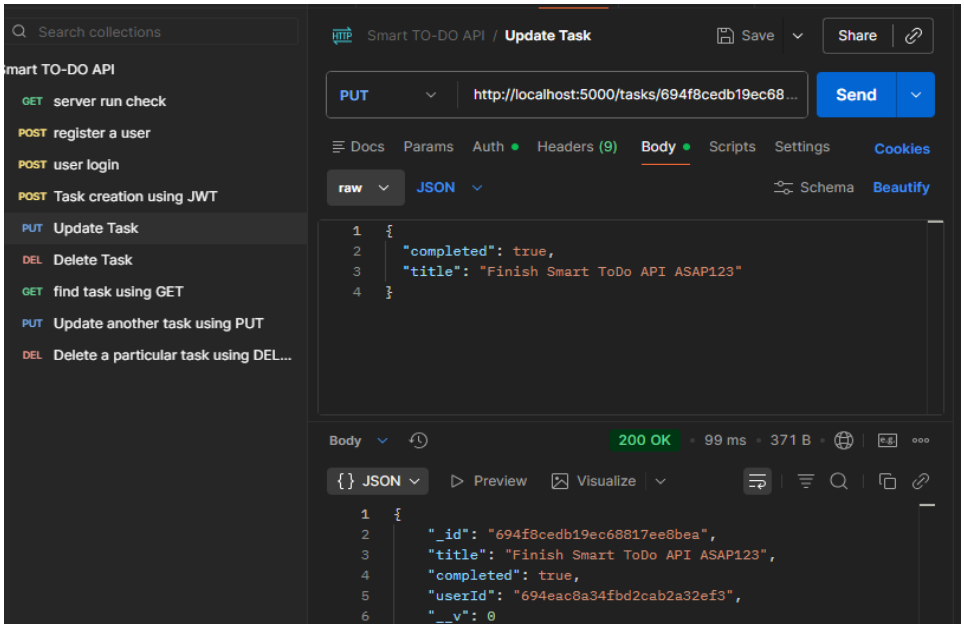
✅Check the Response

Successful Response:

```
{
  "_id": "694eb99da964b59aa33ab7cb",
  "title": "Finish Smart ToDo API project (Updated)",
  "completed": true,
  "userId": "694eac8a34fbd2cab2a32ef3",
  "__v": 0
}
```

Status Code:

200 ok



6) Delete Task using "DELETE" method.

= Login and Obtain JWT Token

1. Send a login request:

```
POST http://localhost:5000/api/auth/login
```

2. Copy the **JWT token** from the response.

□ Get the Task ID

1. Fetch all tasks:

```
GET http://localhost:5000/api/tasks
```

2. Copy the `_id` of the task you want to delete.

Example:

```
"_id": "694eb99da964b59aa33ab7cb"
```

□ Open Postman

1. Create a **New HTTP Request**
2. Select HTTP Method → **DELETE**

□ Enter Delete Task API URL

Replace `<taskId>` with the actual task ID:

```
http://localhost:5000/api/tasks/694eb99da964b59aa33ab7cb
```

□ Set Authorization (JWT)

1. Go to **Authorization** tab
2. Select **Type** → `Bearer Token`
3. Paste the JWT token

□ Send the Request

Click **Send**.

✅Check the Response

Successful Response:

```
{
  "message": "Task deleted successfully"
}
```

Status Code:

`200 OK`