

# G1B3\_BED\_GradedProject4 – PostMan Screenshots

## 1) Adding new user by Admin User

The screenshot shows the Postman interface for a POST request to `http://localhost:8068/v1/users/create/`. The request is configured with form-data and includes the following parameters:

Key	Value	Description
<input checked="" type="checkbox"/> firstName	Akthar	
<input checked="" type="checkbox"/> lastName	Hussain	
<input checked="" type="checkbox"/> email	ahussain@gmail....	
<input checked="" type="checkbox"/> password	AdminAkthar@123	
<input checked="" type="checkbox"/> role	ROLE_ADMIN	
<input checked="" type="checkbox"/> _csrf	_tOXoCLJVxJ1pv...	
Key	Value	Description

The response is a 200 OK status with a body containing the message: `User Added successfully!`

## 2) Login - Normal User

The screenshot shows a REST client interface with a dark theme. The top bar displays a list of requests: GET Login Get CSRF, POST Login User (selected), GET Get All Users, GET Get All Employees, GET Get All Employees, and POST http://localhost. The main area is titled 'Graded Project / Login User'. The request is configured as a POST to 'http://localhost:8068/login'. The 'Body' tab is active, showing a 'form-data' type. A table lists the form fields: \_csrf, username, and password. The response area on the right shows a status of 302 Found, 199 ms, and 504 B. The bottom status bar includes icons for Postbot, Runner, Start Proxy, Cookies, Trash, and a help icon.

GET Login Get CSRF | POST Login User | GET Get All Users | GET Get All Employees | GET Get All Employees | POST http://localhost:8068/login | Insurpay ip\_dev Greg

Graded Project / Login User

POST http://localhost:8068/login

Params Auth Headers (9) Body Pre-req. Tests Settings Cookies

form-data

	Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	_csrf	Text	Gd_ZGX_bChReoU...		
<input checked="" type="checkbox"/>	username	Text	user@ems.com		
<input checked="" type="checkbox"/>	password	Text	NormalUser@456		
	Key	Text	Value	Description	

Body

302 Found 199 ms 504 B Save as example

Pretty Raw Preview Visualize Text

1

Postbot Runner Start Proxy Cookies Trash

### 3) Get all Users

The screenshot shows a REST client interface with a GET request to `http://localhost:8068/v1/users/`. The response is a JSON array of three user objects, displayed in the 'Body' tab. The status is 200 OK, 22 ms, and 728 B.

**Query Params**

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

**Body**

```
1 {
2   {
3     "firstName": "EMS",
4     "lastName": "Admin",
5     "email": "admin@ems.com",
6     "password": null,
7     "role": "ROLE_ADMIN"
8   },
9   {
10    "firstName": "EMS",
11    "lastName": "User",
12    "email": "user@ems.com",
13    "password": null,
14    "role": "ROLE_USER"
15  },
16  {
17    "firstName": "Akthar",
18    "lastName": "Hussain",
19    "email": "ahussain@gmail.com",
20    "password": null,
21    "role": "ROLE_ADMIN"
22  }
23 }
```

Postbot Runner Start Proxy Cookies Trash

## 4) Create User not allowed for Normal User

The screenshot shows a REST client interface with a POST request to `http://localhost:8068/v1/users/create/`. The request body is form data with the following fields:

Key	Type	Value	Description
<input checked="" type="checkbox"/> firstName	Text	Akthar	
<input checked="" type="checkbox"/> lastName	Text	Hussain	
<input checked="" type="checkbox"/> email	Text	ahussain@gmail.com	
<input checked="" type="checkbox"/> password	Text	AdminAkthar@123	
<input checked="" type="checkbox"/> role	Text	ROLE_ADMIN	

The response is a 403 Forbidden status with the following JSON body:

```
{
  "timestamp": 170771729452,
  "status": 403,
  "error": "Forbidden",
  "message": "Forbidden",
  "path": "/v1/users/create/"
}
```

## 5) Create User not allowed for Normal User

The screenshot shows a REST client interface with a POST request to `http://localhost:8068/v1/users/create/`. The request body is form data with the following fields:

Key	Type	Value	Description
<input checked="" type="checkbox"/> firstName	Text	Akthar	
<input checked="" type="checkbox"/> lastName	Text	Hussain	
<input checked="" type="checkbox"/> email	Text	ahussain@gmail.com	
<input checked="" type="checkbox"/> password	Text	AdminAkthar@123	
<input checked="" type="checkbox"/> role	Text	ROLE_ADMIN	

The response is a 403 Forbidden status with the following JSON body:

```
{
  "timestamp": 170771729452,
  "status": 403,
  "error": "Forbidden",
  "message": "Forbidden",
  "path": "/v1/users/create/"
}
```

## 6) Get all users forbidden for Normal User

The screenshot shows a REST client interface with a tab for 'GET Get All Users'. The URL is `http://localhost:8068/v1/users/`. The response status is **403 Forbidden** with a 190 ms response time and 11.68 KB of data. The response body is displayed in JSON format, showing a 'Forbidden' error with a detailed stack trace.

**Query Params**

Key	Value	Description
Key	Value	Description

**Response Body (JSON)**

```
{
  "timestamp": 1707714851763,
  "status": 403,
  "error": "Forbidden",
  "trace": "org.springframework.security.access.AccessDeniedException: Access Denied\n\tat org.springframework.security.authorization.method.AuthorizationManagerBeforeMethodInterceptor.attemptAuthorization (AuthorizationManagerBeforeMethodInterceptor.java:256)\n\tat org.springframework.security.authorization.method.AuthorizationManagerBeforeMethodInterceptor.invoke (AuthorizationManagerBeforeMethodInterceptor.java:197)\n\tat org.springframework.aop.framework.ReflectiveMethodInvocation.proceed (ReflectiveMethodInvocation.java:184)\n\tat org.springframework.aop.framework.CglibAopProxy$CglibMethodInvocation.proceed (CglibAopProxy.java:765)\n\tat org.springframework.aop.framework.CglibAopProxy$DynamicAdvisedInterceptor.intercept (CglibAopProxy.java:717)\n\tat org.ems.employee_management_system.controllers.UserRegAPIController$$SpringCGLIB$$0.listOfUsers(<generated>)\n\tat java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)\n\tat java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:77)\n\tat java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)"
}
```

## 7) New Employee Created

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:8068/v1/employees/create
- Body Type:** form-data
- Form Data:**

Key	Value	Description
<input checked="" type="checkbox"/> fname	Zuber	
<input checked="" type="checkbox"/> lname	Mohd	
<input checked="" type="checkbox"/> email	zuber.mohd@...	
<input checked="" type="checkbox"/> _csrf	6phFURCmun...	
- Response:** 200 OK, 28 ms, 531 B. The response body is a JSON object:

```
{  "id": 1,  "fname": "Zuber",  "lname": "Mohd",  "email": "zuber.mohd@gmail.com",  "created_at": 1707718902238,  "new": true}
```

## 8) Display Employees in Ascending Order

GET Login Get CSRF | POST Login User | GET Get All Users | **GET Get All Employees** | GET Get All Employees | POST CreateUser | + | Insurpay ip\_dev Greg

Graded Project / Get All Employees with Sort Ascending

GET | http://localhost:8068/v1/employees/sort?order=asc | Send

Params • Auth Headers (7) Body Pre-req Tests Settings Cookies

Query Params

<input checked="" type="checkbox"/>	Key	Value	Description	*** Bulk Edit
<input checked="" type="checkbox"/>	order	asc		
	Key	Value	Description	

Body

200 OK 55 ms 1.37 KB Save as example

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "id": 1,
4     "fname": "Aditya",
5     "lname": "Chopra",
6     "email": "achopra@gmail.com",
7     "created_at": 1707717305773,
8     "new": true
9   },
10  {
11    "id": 2,
12    "fname": "Crystal",
13    "lname": "Fernandes",
14    "email": "cferna@gmail.com",
15    "created_at": 1707717331055,
16    "new": true
17  },
18  {
19    "id": 3,
20    "fname": "Jameel",
21    "lname": "Khan",
22    "email": "jkhan@gmail.com",
23    "created_at": 1707717353520,
24    "new": true
25  },
26  {
27    "id": 4,
28    "fname": "Jaspreet",
29    "lname": "Kaur",
30    "email": "jkaur@gmail.com",
31    "created_at": 1707717374180,
32    "new": true
33  },
34  {
35    "id": 5,
36    "fname": "Joseph",
37    "lname": "D'Souza",
```



## 9) Display Employees in Descending Order

The screenshot shows a REST client interface with a GET request to `http://localhost:8068/v1/employees/sort?order=desc`. The response is a JSON array of employee objects, sorted in descending order by ID. The response status is 200 OK, and the body is 1.37 KB.

**Query Params**

Key	Value	Description
order	desc	

**Body (JSON)**

```
{
  "id": 9,
  "fname": "Rahul",
  "lname": "Naik",
  "email": "rnaik@yahoo.com",
  "created_at": 1707717493154,
  "new": true
},
{
  "id": 8,
  "fname": "Micheal",
  "lname": "Jordan",
  "email": "mjordan@gmail.com",
  "created_at": 1707717465795,
  "new": true
},
{
  "id": 7,
  "fname": "Manish",
  "lname": "Dubey",
  "email": "mdubey@gmail.com",
  "created_at": 1707717446802,
  "new": true
},
{
  "id": 6,
  "fname": "Ketan",
  "lname": "Sharma",
  "email": "ksharma@gmail.com",
  "created_at": 1707717423090,
  "new": true
},
{
  "id": 5,
  "fname": "Joseph",
  "lname": "D'Souza",
  "new": true
}
```