

Step 1: Register a User

- **Method:** POST
- **URL:** <http://localhost:8080/auth/register>
- **Body (JSON):**

```
{  
  "name": "John Doe",  
  "phoneNumber": "1234567890",  
  "email": "john.doe@example.com",  
  "password": "password123"  
}
```
- Get the token from the response.

(Email parameter can be skipped also.)

Step 2: Log in to Get a Token

- **Method:** POST
- **URL:** <http://localhost:8080/auth/login>
- **Body (JSON):**

```
{  
  "phoneNumber": "1234567890",  
  "password": "password123"  
}
```
- Get the token from the response.

(We can also add spamCount and user)

Important

- Since all the endpoints are secured, all the API calls expect <http://localhost:8080/auth/register> and <http://localhost:8080/auth/login> will require a token.
- Once a token is received, Go to Postman-> Authorization -> Bearer Token-> Add the token.

Step 3: Add a Contact

- **Method:** POST
- **URL:** http://localhost:8080/contacts/add
- **Headers:**
 - phone: 1234567890
- **Body (JSON):**

```
{  
  "name": "Name of Contact",  
  "phoneNumber": "Phone Number"  
}
```
- Put Token

Response: Contact added successfully

(Same number can be saved with different names.)

(The phone number in header should be a registered user.)

Step 4: Mark a Number as Spam

- **Method:** POST
- **URL:** http://localhost:8080/contacts/mark-spam
- **Params:**
 - Key:** phoneNumber
 - Value:** Number to be in spam.
- Put Token

Response: Number marked as spam

(If phone number already exists, then spamCount increases else a new contact is created with the phone number and spamCount=1)

Step 5: Search for a Contact by Name

- **Method:** GET
- **URL:** <http://localhost:8080/search/by-name>
- **Params:**
 - Key:** name
 - Value:** Name to be searched
- Put Token
- **Response:** Search results display the name, phone number and spamCount for each result matching that name completely or partially.

Step 6: Search for a Contact by Phone Number

- **Method:** GET
- **URL:** <http://localhost:8080/search/by-phone>
- **Params:**
 - Key:** phoneNumber
 - Value:** Phone Number to be searched.
- Put Token
- **Response:** If there is a registered user with that phone number, shows only that result. Otherwise, shows all results matching that phone number completely.

For testing, the database is pre populated with the following data:

Users Table

ID	Name	Phone Number	Email	Password (Hashed)
1	John Doe	1234567890	john.doe@example.com	[hashed password]
2	Jane Smith	0987654321	jane.smith@example.com	[hashed password]

Contacts Table

ID	Name	Phone Number	Spam Count	User ID (Owner)
1	Alice	1111111111	0	1 (John Doe)
2	Bob	2222222222	1	1 (John Doe)
3	Charlie	3333333333	2	2 (Jane Smith)
4	Dave	4444444444	0	2 (Jane Smith)