

POST /api/v1/users – add users

POST localhost:8080/api/v1/users Send

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "userId": "3",
3   "userName": "Henry stark123",
4   "userPhoneNumber": "32764815234",
5   "userEmailAddress": "henrystark123@gmail.com"
6 }
```

Status: 200 OK Size: 117 Bytes Time: 14 ms

Response Headers 4 Cookies Results Docs {} ≡

```
1 {
2   "userId": 3,
3   "userName": "Henry stark123",
4   "userPhoneNumber": "32764815234",
5   "userEmailAddress": "henrystark123@gmail.com"
6 }
```

Copy

GET /api/v1/users/{id} – get details about a user

Welcome Untitled-1 TC localhost:8080/api/v1/use.. X

GET localhost:8080/api/v1/users/1 Send

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "userId": "3",
3   "userName": "Henry stark123",
4   "userPhoneNumber": "32764815234",
5   "userEmailAddress": "henrystark123@gmail.com"
6 }
```

Status: 200 OK Size: 97 Bytes Time: 20 ms

Response Headers 4 Cookies Results Docs {} ≡

```
1 {
2   "userId": 1,
3   "userName": "Henry",
4   "userPhoneNumber": "32764815",
5   "userEmailAddress": "henry@gmail.com"
6 }
```

Copy

Welcome Untitled-1 TC localhost:8080/api/v1/use.. X

GET localhost:8080/api/v1/users/2 Send

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "userId": "3",
3   "userName": "Henry stark123",
4   "userPhoneNumber": "32764815234",
5   "userEmailAddress": "henrystark123@gmail.com"
6 }
```

Status: 200 OK Size: 117 Bytes Time: 10 ms

Response Headers 4 Cookies Results Docs {} ≡

```
1 {
2   "userId": 2,
3   "userName": "Henry stark123",
4   "userPhoneNumber": "32764815234",
5   "userEmailAddress": "henrystark123@gmail.com"
6 }
```

Copy

PUT /api/v1/users/{id} – update a user

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

- Method:** PUT
- URL:** localhost:8080/api/v1/users/2
- Body:** A JSON object with the following content:

```
1 {
2   "userId" : "3",
3   "userName": "Henry Nonestark123",
4   "userPhoneNumber": "32764815234",
5   "userEmailAddress": "henrystark123@gmail.com"
6 }
```
- Status:** 200 OK
- Size:** 121 Bytes
- Time:** 14 ms
- Response:** A JSON object with the following content:

```
1 {
2   "userId": 2,
3   "userName": "Henry Nonestark123",
4   "userPhoneNumber": "32764815234",
5   "userEmailAddress": "henrystark123@gmail.com"
6 }
```

DELETE /api/v1/users/{id} – delete a user

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

- Method:** DELETE
- URL:** localhost:8080/api/v1/users/2
- Body:** A JSON object with the following content:

```
1 {
2   "userId" : "3",
3   "userName": "Henry Nonestark123",
4   "userPhoneNumber": "32764815234",
5   "userEmailAddress": "henrystark123@gmail.com"
6 }
```
- Status:** 200 OK
- Size:** 0 Bytes
- Time:** 21 ms
- Response:** An empty response (1 line).

Details Of the Tables :

RunRun SelectedAuto completeClear

SQL statement:

SELECT * FROM USER_TABLE |

SELECT * FROM USER_TABLE;

USER_ID	USER_EMAIL_ADDRESS	USER_NAME	USER_PHONE_NUMBER
1	henry@gmail.com	Henry	32764815
2	henrystark123@gmail.com	Henry stark123	32764815234
3	henrystark123@gmail.com	Henry stark123	32764815234

(3 rows, 1 ms)

RunRun SelectedAuto completeClear

SQL statement:

SELECT * FROM USER_TABLE

SELECT * FROM USER_TABLE;

USER_ID	USER_EMAIL_ADDRESS	USER_NAME	USER_PHONE_NUMBER
1	henry@gmail.com	Henry	32764815
3	henrystark123@gmail.com	Henry stark123	32764815234

(2 rows, 1 ms)