

Method - POST (Create)

1. Valid details passed in Request

POST http://localhost:8080/todo/item

Params

Send

```
1 {
2   "requests": [
3     {
4       "itemDescription": "Shopping",
5       "tagName": "Shop",
6       "reminder": "30/04/2021 10:30:00"
7     }, {
8       "itemDescription": "Studying",
9       "tagName": "Study",
10      "reminder": "30/04/2021 10:30:00"
11    }
12  ]
13 }
```

Body Cookies Headers (5) Test Results Status: 201 Created

Pretty Raw Preview JSON

```
1 {
2   "todoList": [
3     {
4       "id": 1,
5       "itemDescription": "Shopping",
6       "tagName": "Shop",
7       "reminder": "30/04/2021 10:30:00",
8       "message": "Item added to TODO List Successfully",
9       "done": false
10    },
11    {
12      "id": 2,
13      "itemDescription": "Studying",
14      "tagName": "Study",
15      "reminder": "30/04/2021 10:30:00"
16    }
17  ]
18 }
```

H2 Snapshot

jdbc:h2:mem:testdb

TODO_ITEMS

INFORMATION_SCHEMA

Sequences

Users

H2 1.4.200 (2019-10-14)

Run Run Selected Auto complete Clear SQL statement:

SELECT * FROM TODO_ITEMS;

ID	ITEM_DESC	TAG_NAME	REMINDER	IS_DONE	MODIFIED_ON
1	Shopping	Shop	2021-04-30	0	2021-04-15
2	Studying	Study	2021-04-30	0	2021-04-15

(2 rows, 3 ms)

Edit

2. Item description if passed empty in request

POST http://localhost:8080/todo/item Params

Authorization Headers (1) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "requests": [
3     {
4       "itemDescription": "",
5       "tagName": "Shop",
6       "reminder": "30/04/2021 10:30:00"
7     }
8   ]
9 }
10
```

Body Cookies Headers (4) Test Results Status: !

Pretty Raw Preview JSON

```
1 {
2   "timestamp": "2021-04-15T00:27:23.344+00:00",
3   "message": "Item Description cannot be empty",
4   "details": "uri=/todo/item"
5 }
```

3. If given item already exists

POST http://localhost:8080/todo/item Params

Authorization Headers (1) Body Pre-request Script Tests

form-data x-www-form-urlencoded raw binary JSON (application/json)

```
1 {
2   "requests": [
3     {
4       "itemDescription": "Shopping",
5       "tagName": "Shop",
6       "reminder": "30/04/2021 10:30:00"
7     }
8   ]
9 }
10
```

Body Cookies Headers (5) Test Results

Pretty Raw Preview JSON

```
1 {
2   "timestamp": "2021-04-15T00:28:01.649+00:00",
3   "message": "TODO Item Description already exists :: Shopping",
4   "details": "uri=/todo/item"
5 }
```

Method - PUT (Update)

1. Valid Data passed in request

The screenshot shows a REST client interface with a PUT request to `http://localhost:8080/todo/item`. The request body is a JSON object with a `requests` array containing one item. The response status is `200 OK` and the response body is a JSON object with a `todoList` array containing the updated item and a success message.

```
PUT http://localhost:8080/todo/item

{
  "requests": [
    {
      "id": "1",
      "itemDescription": "Studying New",
      "tagName": "Study",
      "reminder": "01/05/2021 10:30:00",
      "done": 1
    }
  ]
}
```

Body Cookies Headers (5) Test Results Status: 200 OK Time:

Pretty Raw Preview JSON

```
{
  "todoList": [
    {
      "id": 1,
      "itemDescription": "Studying New",
      "tagName": "Study",
      "reminder": "01/05/2021 10:30:00",
      "message": "Item Details updated Successfully",
      "done": true
    }
  ]
}
```

H2 Snapshot

The screenshot shows an H2 database snapshot. At the top, there are tabs for 'Full', 'Full Snapshot', 'Full Complete', and 'Full SQL statements'. The 'Full' tab is selected, showing the SQL query `SELECT * FROM TODO_ITEMS`. Below the query, the results are displayed as a table with 6 columns: ID, ITEM_DESC, TAG_NAME, REMINDER, IS_DONE, and MODIFIED_ON. There are 2 rows of data. The first row has ID 1, ITEM_DESC 'Studying New', TAG_NAME 'Study', REMINDER '2021-05-01', IS_DONE 1, and MODIFIED_ON '2021-04-15'. The second row has ID 2, ITEM_DESC 'Studying', TAG_NAME 'Study', REMINDER '2021-04-30', IS_DONE 0, and MODIFIED_ON '2021-04-15'. Below the table, it says '(2 rows, 9 ms)'. At the bottom, there is an 'Edit' button.

```
SELECT * FROM TODO_ITEMS;
```

ID	ITEM_DESC	TAG_NAME	REMINDER	IS_DONE	MODIFIED_ON
1	Studying New	Study	2021-05-01	1	2021-04-15
2	Studying	Study	2021-04-30	0	2021-04-15

(2 rows, 9 ms)

Edit


2. Invalid Request passed in input

PUT ▼ http://localhost:8080/todo/item Params Send ▼ Save ▼

form-data x-www-form-urlencoded raw binary JSON (application/json) ▼

```
1 {
2   "requests": [
3
4     {
5       "id": "1112121",
6       "itemDescription": "Studying New",
7       "tagName": "Study",
8       "reminder": "31/04/2021 10:30:00",
9       "done": 1
10     }
11   ]
12 }
```

Body Cookies Headers (5) Test Results Status: 404 Not Found Time: 32 ms

Pretty Raw Preview JSON ▼  Save Response

```
1 {
2   "timestamp": "2021-04-15T00:25:31.532+00:00",
3   "message": "TODO Item doesn't exist for the given ID :: 1112121",
4   "details": "uri=/todo/item"
5 }
```


Method - DELETE

1. Valid Data

DELETE ▼ http://localhost:8080/todo/item Params Send ▼ Save ▼

```
1 {}
2
3 "requests": [
4   {
5     "id": "2",
6     "itemDescription": "Shopping",
7     "tagName": "Study",
8     "reminder": "",
9     "done": 0
10   }
11 ]
12 }
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 38 ms

Pretty Raw Preview JSON ▼  Save Response

```
1 {
2   "todoList": [
3     {
4       "id": 2,
5       "itemDescription": "Shopping",
6       "tagName": "Study",
7       "reminder": null,
8       "message": "Item deleted successfully from TODO List",
9       "done": false
10     }
11   ]
12 }
```

H2 Snapshot (Item 2 deleted)

```
SELECT * FROM TODO_ITEMS
```

```
SELECT * FROM TODO_ITEMS;
```

ID	ITEM_DESC	TAG_NAME	REMINDER	IS_DONE	MODIFIED_ON
1	Studying New	Study	2021-05-01	1	2021-04-15

(1 row, 3 ms)

Edit

2. Invalid Id passed in input


DELETE ▼ http://localhost:8080/todo/item Params Send ▼ Save

Authorization Headers (1) **Body** Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼


```
1 {
2   "requests": [
3     {
4       "id": "121212",
5       "itemDescription": "Shopping",
6       "tagName": "Study",
7       "reminder": "",
8       "done": 0
9     }
10  ]
11 }
```

Body Cookies Headers (5) Test Results Status: 404 Not Found Time: 34

Pretty Raw Preview JSON ▼  Save Respor

```
1 {
2   "timestamp": "2021-04-15T00:23:54.041+00:00",
3   "message": "TODO Item doesn't exist for the given ID :: 121212",
4   "details": "uri=/todo/item"
5 }
```

Method - Get To Do Items

GET 

http://localhost:8080/todo/item

Authorization


Headers

Body

Pre-request Script

Tests

Type

No Auth 

Body

Cookies


Headers (5)


Test Results

Pretty

Raw

Preview

JSON 



1

2

3

4

5

6

7

8

9

10

[

{

"id": 1,

"itemDescription": "Studying New",

"tagName": "Study",

"reminder": "30/04/2021 14:00:00",

"message": null,

"done": true

}

]