

Todo App API Documentation

API Documentation

This section is integrated with C#.NET RESTful API Implementations. For database use the Sqlite Db

API EndPoints

Base URL: <https://localhost:5001/api/todo-task>

- GET
 - **/get-all - Get List of Todos**

-> 200 OK - Successful Request. View Available Todos

Example output:

```
[
  {
    "title": "string",
    "description": "string",
    "id": 0,
    "isCompleted": true,
    "priority": 0
  }
]
```

-> 204 No Content - If not any todo in the database

-> 400 Bad Request - If any unhandled error is thrown

GET /api/todo-task/get-all

Parameters

No parameters

Try it out

Responses

Code	Description	Links
200	Success	No links

Media type: text/plain

Controls Accept header.

Example Value | Schema

```
[
  {
    "title": "string",
    "description": "string",
    "id": 0,
    "isCompleted": true,
    "priority": 0
  }
]
```

- **/id} - Get Todo by Id**

-> 200 OK - Successful Request. View Available Todos

Example output:

```
{
  "title": "string",
  "description": "string",
  "id": 0,
  "isCompleted": true,
  "priority": 0
}
```

-> 404 Not Found - Todo related to the given Id is not in the database

-> 400 Bad Request - If any unhandled error is thrown

Example output:

Here 1 means the given id number

```
1 task not found
```

The image shows a Swagger UI interface for the endpoint `GET /api/todo-task/{id}`. The interface is divided into two main sections: **Parameters** and **Responses**.

Parameters: This section shows a single path parameter named `id`. It is marked as `required` and has a type of `integer($int64)`. Below the parameter name is a text input field with the placeholder text `id`.

Responses: This section shows a single response with a status code of `200` and a description of `Success`. The response is associated with a media type of `text/plain`. Below the response information, there is a section for **Example Value** which displays a JSON object:

```
{
  "title": "string",
  "description": "string",
  "id": 0,
  "isCompleted": true,
  "priority": 0
}
```

- POST
 - /create - Add new Todo Task

Sample Input:

```
{
  "title": "string",
  "description": "string"
}
```

-> 201 OK - Create new Todo is Success

Sample output

```
{
  "title": "Test1",
  "description": "Test1"
}
```

-> 400 Bad Request - If any validation or unhandled error thrown

Validations : Title can't be null or only contain spaces

Sample output

```
Request can't be null or contain only spaces
```

Validations : Request can't be null

Sample output

```
{
  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
  "title": "One or more validation errors occurred.",
  "status": 400,
  "traceId": "00-4e71e4a3d6ce68bea50bc2be36a35b22-c97c19e1d1101e38-00",
  "errors": {
    "$": [
      "'n' is an invalid start of a property name. Expected a '\"'. Path: $ |
      LineNumber: 0 | BytePositionInLine: 1."
    ],
    "todoValues": [
      "The todoValues field is required."
    ]
  }
}
```

POST /api/todo-task/create

Parameters Try it out

No parameters

Request body application/json

Example Value | Schema

```
{
  "title": "string",
  "description": "string"
}
```

Responses

Code	Description	Links
200	Success	No links

- PUT

- /update - update relevant todo task

Sample input

```
{
  "title": "string",
  "description": "string",
  "id": 0,
  "isCompleted": true,
  "priority": 0
}
```

*Here Priorities has been declared as an enum as follows

Default = 0,

Low = 1,

Medium = 2,

High = 3

-> 200 OK - Update Todo is a Success

Sample output

```
{
  "title": "Test1",
  "description": "Test1"
}
```

-> 400 Bad Request - If any validation or unhandled error is thrown

Validations : Title can't be null or only contain spaces

Sample output

```
Request can't be null or contain only spaces
```

Validations : Request can't be null

Sample output

```
{
  "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
  "title": "One or more validation errors occurred.",
  "status": 400,
  "traceId": "00-4e71e4a3d6ce68bea50bc2be36a35b22-c97c19e1d1101e38-00",
  "errors": {
    "$": [
      "'n' is an invalid start of a property name. Expected a '\"'. Path: $ |
      LineNumber: 0 | BytePositionInLine: 1."
    ],
    "todoValues": [
      "The todoValues field is required."
    ]
  }
}
```

-> 404 Not Found - Todo related to the given Id is not in the database

Sample output : Here 0 means the Id of the given TodoUpdateDto

0 task not found

- DELETE

- /{id} - Delete given Todo

-> 200 OK - Delete Todo is a Success

Sample output

Task Deleted Successfully

-> 400 Bad Request - If any unhandled error is thrown

-> 404 Not Found - Todo related to the given Id is not in the database

Sample output : Here 0 means the Id of the given TodoUpdateDto

0 task not found

DELETE /api/todo-task/{id}		
Parameters		
Name	Description	
id * required	id	
Integer(\$int64)		
(path)		
Responses		
Code	Description	Links
200	Success	No links