

TodoList

/openapi.json

Authorize



The Todo List API is a RESTful web service that allows users to create, read, update, and delete items on a to-do list. The API utilizes HTTP requests to send and receive data in JSON format. The endpoints available in this API are as follows:

GET /tasks - Retrieves a list of all the to-do items in the database.

GET /tasks/{id} - Retrieves a specific to-do item by its ID.

POST /tasks - Creates a new to-do item in the database.

PUT /tasks/{id} - Updates a specific to-do item by its ID.

DELETE /tasks/{id} - Deletes a specific to-do item by its ID

Authentication is required to access the Todo List API

The Todo List API provides a straightforward and flexible way for developers to add to-do list functionality to their applications. With its simple endpoints and data model, developers can easily integrate the API into their existing applications or build new ones from scratch.

default



POST /sign_up Sign Up



Parameters

Try it out

No parameters

Request body **required**

application/json

Example Value Schema

```
{
  "username": "string",
  "password": "string"
}
```

This endpoint is used to register user and save their credentials in database after encrypting the password

Responses

Code	Description	Links
200	Successful Response	No links
<div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div>"string"</div>		

Code	Description	Links
422	Validation Error	No links
	<div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div><pre>{ "detail": [{ "loc": ["string", 0], "msg": "string", "type": "string" }]}</pre></div>	

POST /token Login

Parameters

Try it out

No parameters

Request body required

application/x-www-form-urlencoded

grant_type

String
pattern: password

username * required
string

password * required
string

scope
string

client_id
string

client_secret
string

This endpoint is responsible for taking input credentials from user to Login and based on it ,user is authenticated and a Token is generated . this token has a lifespan after which it is discarded

Responses

Code	Description	Links
200	<div>Successful Response</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div>"string"</div>	No links
422	<div>Validation Error</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div>{ "detail": [{ "loc": ["string", 0], "msg": "string", "type": "string" }] }</div>	No links

GET / Home



Parameters

Try it out

No parameters

This endpoint is for home page which is expecting a token value corresponding to user .this token can be used further to verify user

Responses

Code	Description	Links
200	Successful Response	No links

Media type

application/json

Controls Accept header.

Example Value Schema

"string"

GET /tasks Read Tasks

Parameters

Try it out

No parameters

This endpoint shows all the tasks stored in Database along with their Details

Responses

Code	Description	Links
200	Successful Response	No links

Media type

application/json

Controls Accept header.

Example Value Schema

"string"

POST /tasks Create Task

Parameters

Try it out

No parameters

Request body required

application/json

Example Value Schema

```
{
  "id": 0,
  "title": "string",
  "description": "string",
  "done": false
}
```

This Endpoint Create task and store them in Database

Responses

Code	Description	Links
200	<div>Successful Response</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div>"string"</div>	No links
422	<div>Validation Error</div> <div>Media type</div> <div>application/json</div> <div>Example Value Schema</div> <div>{ "detail": [{ "loc": ["string", 0], "msg": "string", "type": "string" }] }</div>	No links

GET /tasks/{task_id} Read Task



Parameters

Try it out

Name Description

task_id * required
integer
(path)

task_id

This endpoint show the specific task details based on task Id

Responses

Code	Description	Links
200	Successful Response	No links

Media type

application/json

Controls Accept header.

Example Value Schema

```
{
  "id": 0,
  "title": "string",
  "description": "string",
  "done": false
}
```

422	Validation Error	No links
-----	------------------	----------

Media type

application/json

Example Value Schema

```
{
  "detail": [
    {
      "loc": [
        "string",
        0
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

PUT

/tasks/{task_id}

Update Task

Parameters

Try it out

Name	Description
task_id * required integer (path)	<div>task_id</div>

Request body required

application/json

Example Value

Schema

```
{
  "id": 0,
  "title": "string",
  "description": "string",
  "done": false
}
```

This endpoint is responsible for updating the task which is already stored in database

Responses

Code	Description	Links
200	<div>Successful Response</div> <div><div>Media type</div><div>application/json</div><div>Controls Accept header.</div></div> <div><div>Example Value</div><div>Schema</div></div> <div><pre>{ "id": 0, "title": "string", "description": "string", "done": false }</pre></div>	No links
422	<div>Validation Error</div> <div><div>Media type</div><div>application/json</div></div>	No links

Code	Description	Links
------	-------------	-------

Example Value Schema

```
{
  "detail": [
    {
      "loc": [
        "string",
        0
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

DELETE /tasks/{task_id} Delete Task



Parameters

Try it out

Name	Description
------	-------------

task_id * required
integer
(path)

task_id

This endpoint Delete the task from Database using Task ID

Responses

Code	Description	Links
------	-------------	-------

200	Successful Response	No links
-----	---------------------	----------

Media type

application/json

Controls Accept header.

Example Value Schema

```
"string"
```

422	Validation Error	No links
-----	------------------	----------

Media type

Code	Description	Links
	<div>application/json</div> <div>Example Value Schema</div> <pre>{ "detail": [{ "loc": ["string", 0], "msg": "string", "type": "string" }]}</pre>	

Schemas ^

Body_login_token_post {

grant_type	Grant Type	[...]
username*	Username	[...]
password*	Password	[...]
scope	Scope	[...]
client_id	Client Id	[...]
client_secret	Client Secret	[...]

}

HTTPValidationError

Task {

id	Id	[...]
title*	Title	[...]
description	Description	[...]
done	Done	[...]

}

User

{

username*

password*

}

Username

[...]

Password

[...]

ValidationError