TodoList

/openapi.json



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.

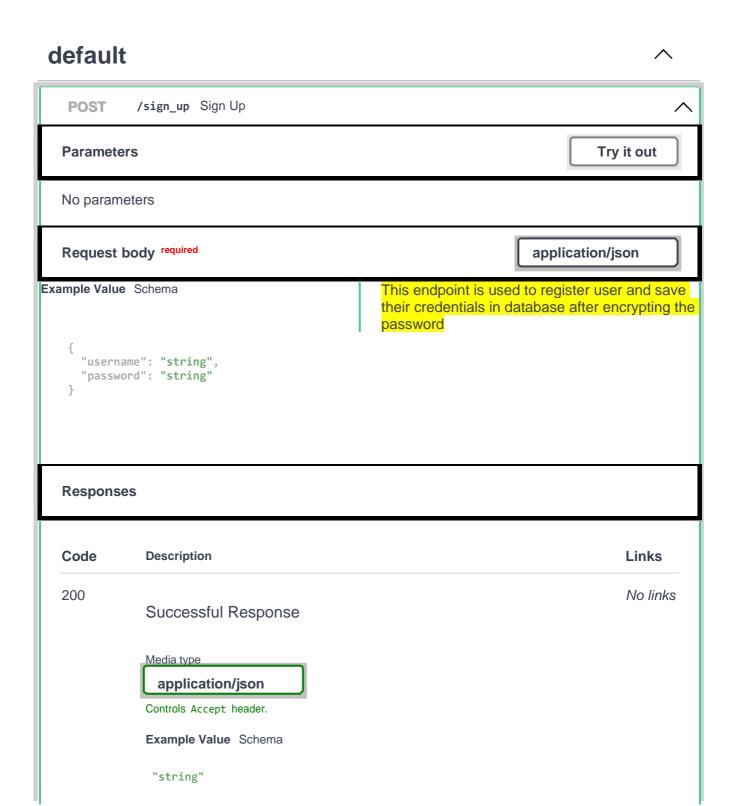
POST /tasks - Creates a new to-do item in the database.

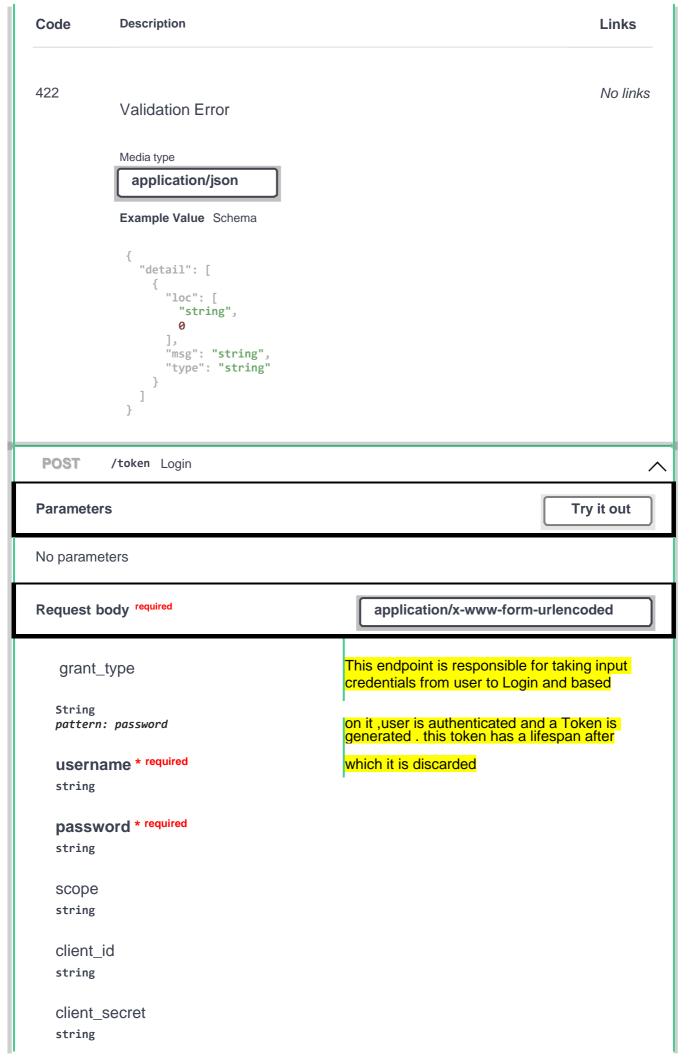
PUT /tasks/{id} - Updates a specific to-do item by its ID.

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.





Responses

Code

200		No links

Successful Response

Media type

Description

application/json

Controls Accept header.

Example Value Schema

"string"

422 No links
Validation Error

Media type

application/json

Example Value Schema

GET / Home

^ 🔒

Links

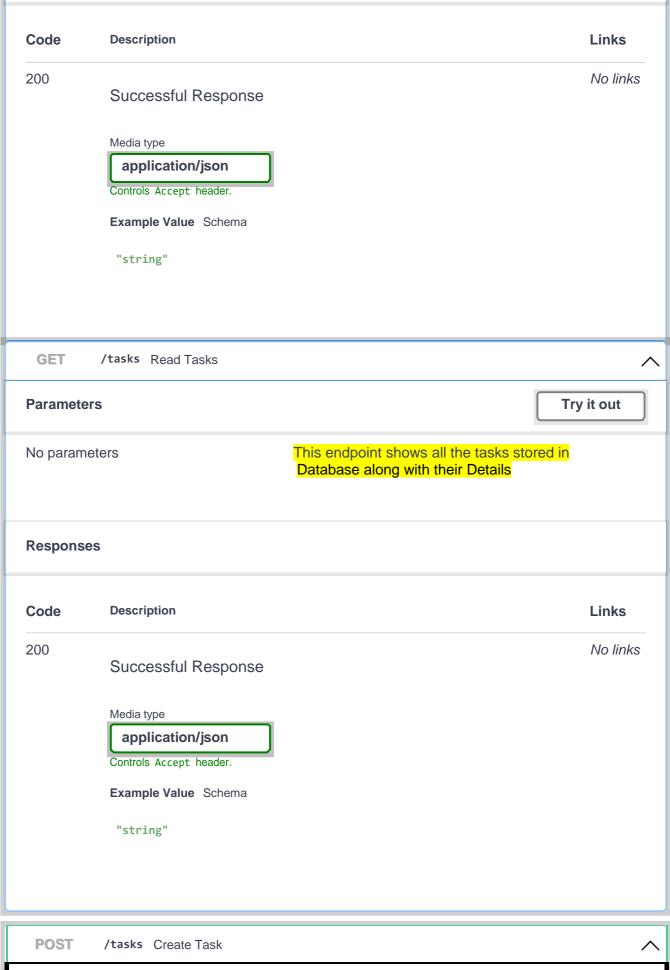
Parameters

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



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 No links
Successful Response

Media type

application/json

Controls Accept header.

Example Value Schema

"string"

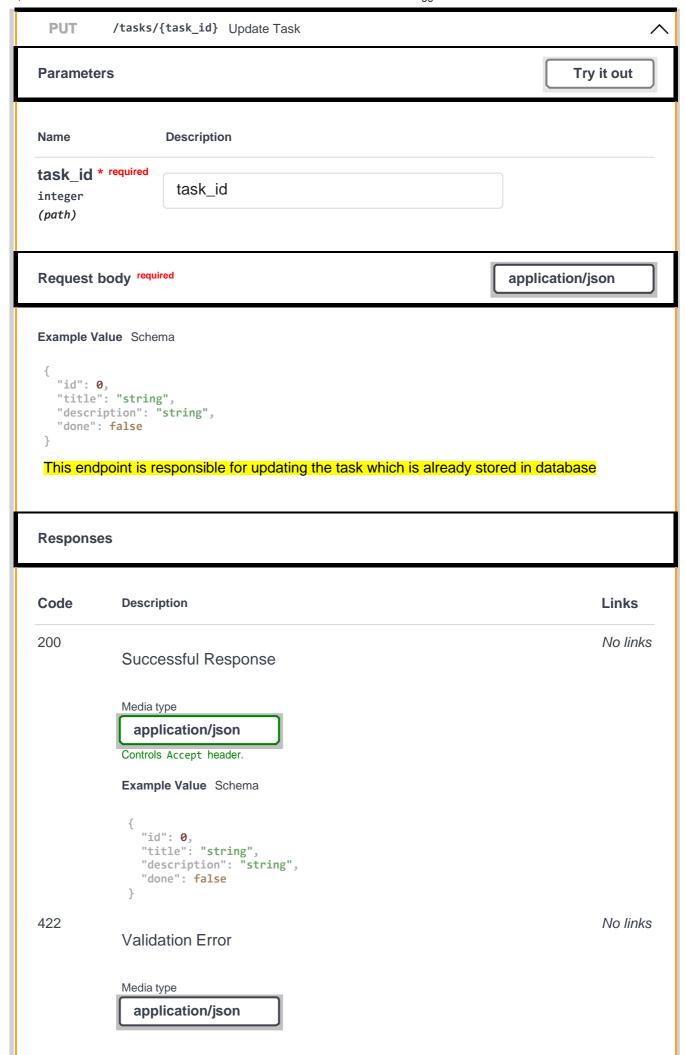
422 No links
Validation Error

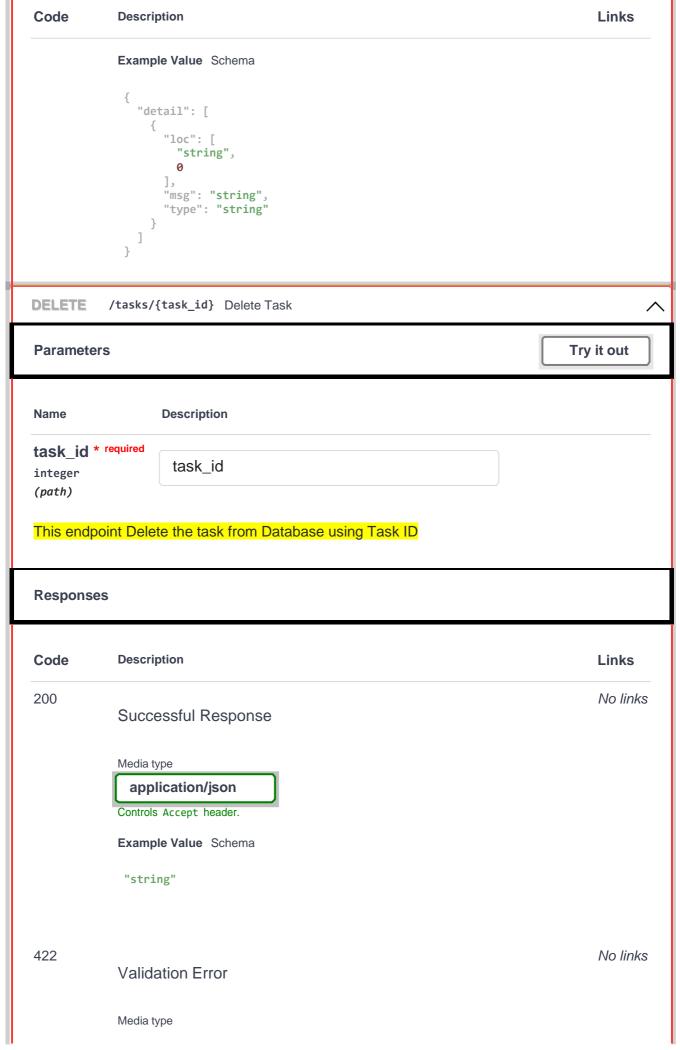
Media type

application/json

Example Value Schema

GET /tasks/{task_id} Read Task Try it out **Parameters** Name **Description** task_id * required task_id integer (path) This endpoint show the specific task details based on task Id Responses **Description** Links Code 200 No links Successful Response Media type application/json Controls Accept header. Example Value Schema "id": 0, "title": "string", "description": "string", "done": false 422 No links Validation Error Media type application/json Example Value Schema "detail": ["loc": ["string", "msg": "string", "type": "string" }





```
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
                          ld [...]
      title*
                          Title
                                  [...]
      description
                          Description [...]
      done
                          Done
                                 [...]
   }
```