

# TODO List API Documentation

## Introduction:

This API is designed to allow users to create, read, update, and delete TODO items. It is built using FastAPI and MySQL.

## Endpoints:

The following endpoints are available:

### GET /todos:

This endpoint returns a list of all TODO items.

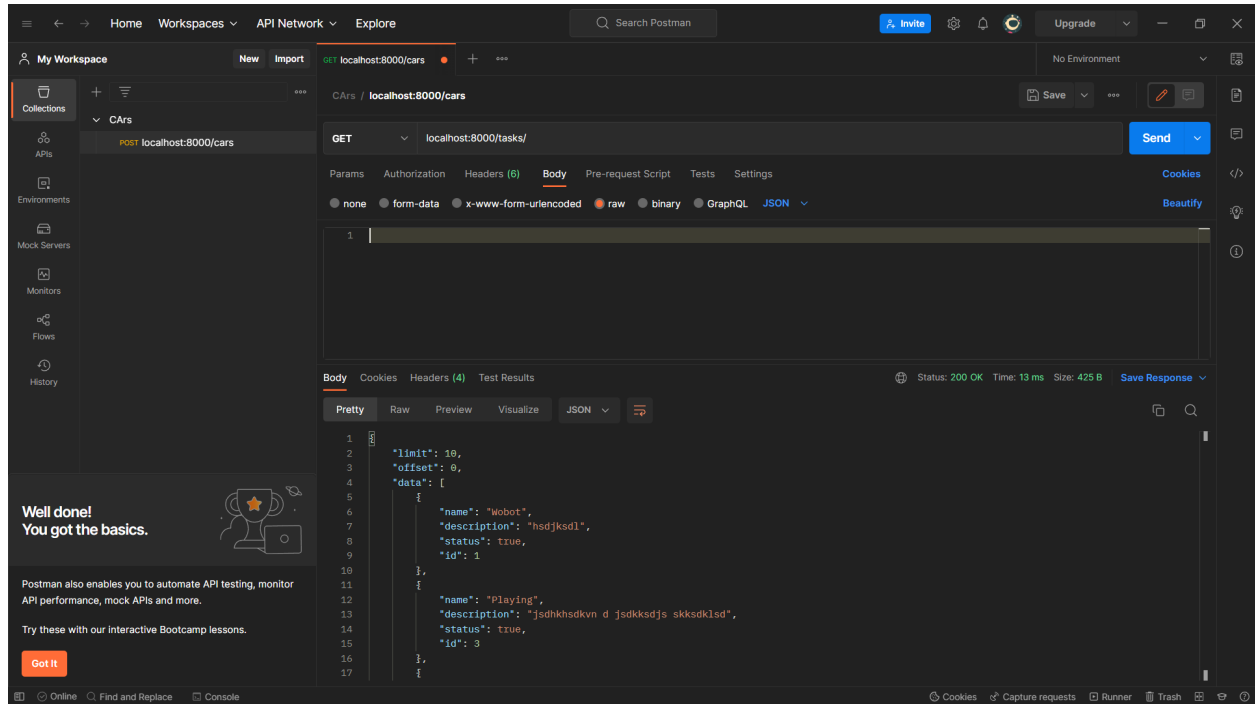
```
# API to get the list of task info
@router.get("/tasks", response_model=PaginatedTaskInfo)
def list_tasks(self, limit: int = 10, offset: int = 0):

    tasks_list = get_all_tasks(self.session, limit, offset)
    response = {"limit": limit, "offset": offset, "data":
tasks_list}

    return response
```

```
# API endpoint to get info of a particular task
@router.get("/tasks/{task_id}", response_model=Task)
def get_task_info(task_id: int, session: Session =
Depends(get_db)):

    try:
        task_info = get_task_info_by_id(session, task_id)
        return task_info
    except TodoItemException as cie:
        raise HTTPException(**cie.__dict__)
```



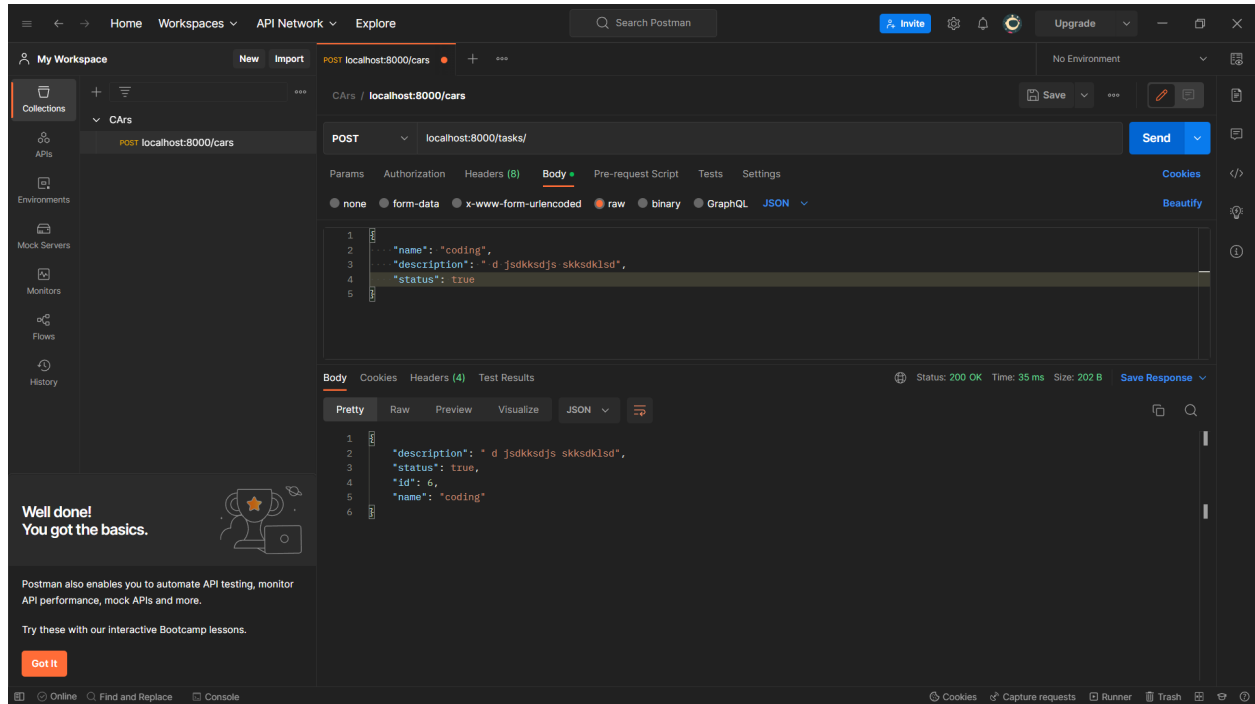
## POST /todos:

This endpoint allows you to create a new TODO item.

```
# API endpoint to add a task info to the database
@router.post("/tasks")
def add_task(self, task_info: CreateAndUpdateTask):

    try:
        task_info = create_task(self.session, task_info)
        return task_info
    except TodoItemException as cie:
        raise HTTPException(**cie.__dict__)
```

# Kartik Kumar



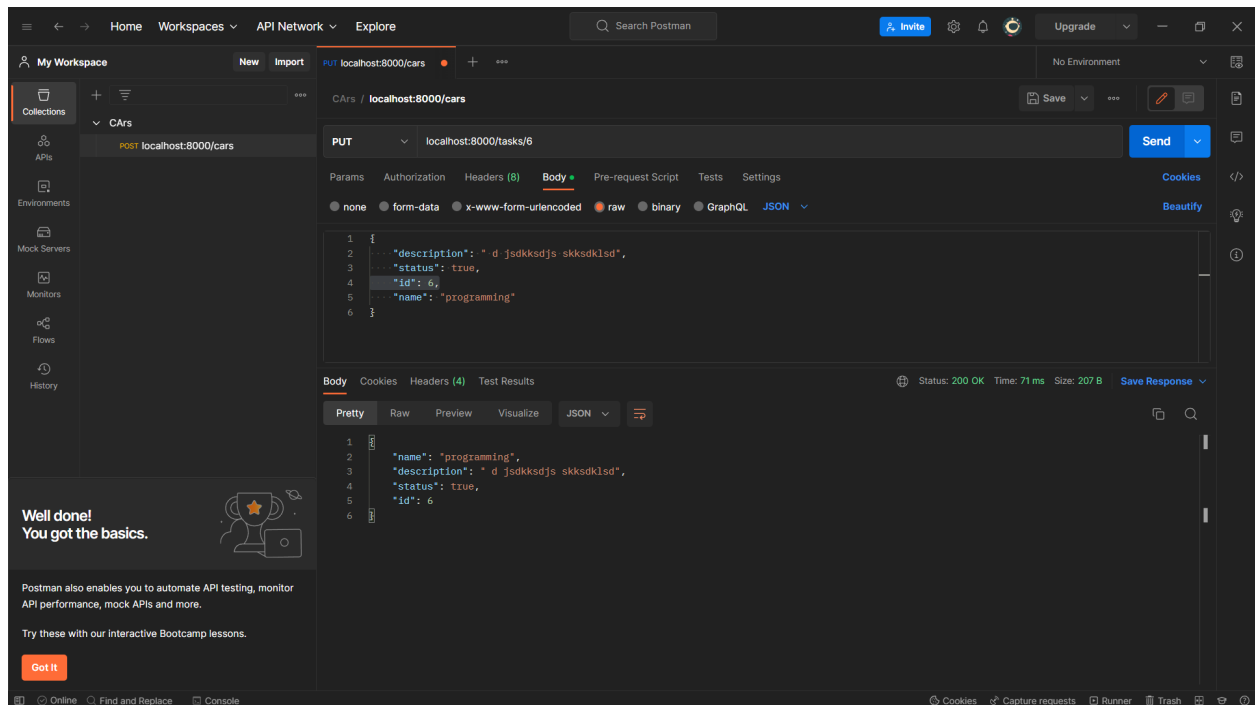
Kartik Kumar

## **PUT /todos/{id}:**

This endpoint allows you to update a TODO item.

```
# API to update a existing task info
@router.put("/tasks/{task_id}", response_model=Task)
def update_task(task_id: int, new_info: CreateAndUpdateTask,
session: Session = Depends(get_db)):

    try:
        task_info = update_task_info(session, task_id, new_info)
        return task_info
    except TodoItemException as cie:
        raise HTTPException(**cie.__dict__)
```

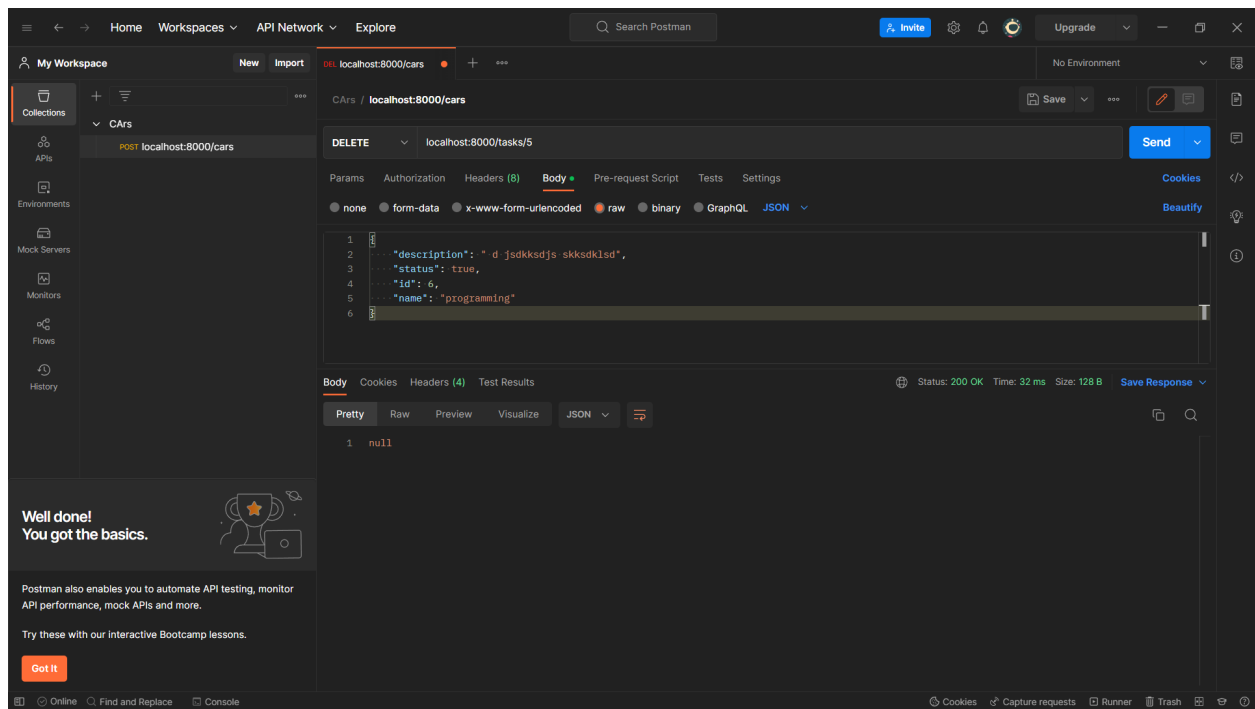


## DELETE /todos/{id}:

This endpoint allows you to delete a TODO item.

```
# API to delete a task info from the data base
@router.delete("/tasks/{task_id}")
def delete_task(task_id: int, session: Session = Depends(get_db)) :

    try:
        return delete_task_info(session, task_id)
    except TodoItemException as cie:
        raise HTTPException(**cie.__dict__)
```



## Conclusion

This API provides a simple and easy-to-use interface for managing TODO items. With its intuitive endpoints and straightforward documentation, users can quickly and easily create, read, update, and delete TODO items with ease.