

Create the base application structure

Task Manager v101 - Creating Base Application

Let's create the Task Manager Application Step By Step by Step

Create New Application

```
nest new taskApi
```

Remove the Default Controller and Services from the Application

- Delete the `app.controller.ts`
- Delete the `app.service.ts`
- Delete the references of `app.controller` and `app.services` from the `app.module.ts` file

Add the Initial Files Required for the project

- Add a task module to the application

```
nest g module task
```

- Add a controller for the application

```
nest g controller task
```

Adding routes to the Application

- Add the routes for the task which return a simple text message

```
import { Controller, Delete, Get, Param, Patch, Post } from '@nestjs/common';

@Controller('task')
export class TaskController {

  @Get()
  getAll(): string {
    return 'Gets all the tasks'
  }

  @Post()
  createTask(): string {
    return 'Creating New Task'
  }
}
```

```

    }

    @Patch('/:id')
    updateTask(@Param('id') id:string):string{
        return `Updating Task ${id}`
    }

    @Delete('/:id')
    deleteTask(@Param('id') id:string):string{
        return `Deleting Task ${id}`
    }
}

```

Create a dto for create and update tasks

DTO's --> Data Transfer Objects, are used to pass data into the API endpoints and as response. They are created so that all the underlying fields in the entity are not exposed to the consumer's of the API endpoints.

- create a file `create-task.dto.ts` with the code :

```

export class CreateTaskDto{
    title:string;
    description:string;
}

```

- create a file `update-task.dto.ts` with the code:string

```

export class CreateTaskDto{
    title:string;
    description:string;
}

```

- Update the code to read the value from the body of Post and Patch requests

```

@Post()
createTask(@Body() createTaskDto:CreateTaskDto):string{
    const {title, description} = createTaskDto
    return `Creating New Task {'title':'${title}'
,'description':'${description}'}\`
}

@Patch('/:id')
updateTask(@Param('id') id:string,@Body() updateTaskDto:UpdateTaskDto):string{
    const {title, description} = updateTaskDto
    return `Updating Task ${id} with data title:${title}
description:${description}\`
}

```

```
}

```