

TASK MANAGER PROJECT

Step 1: Create the Project Folder

Open your terminal and run:

```
mkdir task-manager-project
```

```
cd task-manager-project
```

Step 2: Initialize a Node.js Project

Run this command inside the task-manager-project folder:

```
npm init -y
```

This will generate a package.json file.

Step 3: Install TypeScript

To use TypeScript, install it globally (if not already installed):

```
npm install -g typescript
```

Now, install it locally in your project:

```
npm install --save-dev typescript
```

Step 4: Configure TypeScript

Run this command to generate a tsconfig.json file:

```
tsc -init
```

Now, open tsconfig.json and modify the settings like this:

```
{  
  "compilerOptions": {  
    "target": "ES6",  
    "module": "CommonJS",  
    "rootDir": "./src",  
    "outDir": "./dist",  
    "strict": true,
```

```
"sourceMap": true
}
}
```

Step 5: Create the Project Structure

Now, create the necessary folders and files:

```
mkdir src
mkdir src/models src/services
touch src/models/Task.ts
touch src/services/TaskManager.ts
touch src/index.ts
```

Step 6: Implement the Code

1. Define Task Model (Task.ts)

Open src/models/Task.ts and add:

```
typescript
CopyEdit
export class Task {
  constructor(
    public id: number,
    public title: string,
    public description: string,
    public completed: boolean = false
  ) {}

  markAsComplete(): void {
    this.completed = true;
  }
}
```

```
}
```

2. Implement Task Manager (TaskManager.ts)

Open src/services/TaskManager.ts and add:

typescript

```
import { Task } from "../models/Task";
```

```
export class TaskManager {
```

```
    private tasks: Task[] = [];
```

```
    private nextId: number = 1;
```

```
    addTask(title: string, description: string): Task {
```

```
        const task = new Task(this.nextId++, title, description);
```

```
        this.tasks.push(task);
```

```
        return task;
```

```
    }
```

```
    getTasks(): Task[] {
```

```
        return this.tasks;
```

```
    }
```

```
    markTaskAsComplete(id: number): boolean {
```

```
        const task = this.tasks.find(task => task.id === id);
```

```
        if (task) {
```

```
            task.markAsComplete();
```

```
            return true;
```

```
    }  
    return false;  
  }  
}
```

3. Implement Main File (index.ts)

Open src/index.ts and add:

```
import { TaskManager } from "../services/TaskManager";
```

```
const taskManager = new TaskManager();
```

```
// Adding tasks
```

```
const task1 = taskManager.addTask("Learn TypeScript", "Understand TypeScript basics  
and usage.");
```

```
const task2 = taskManager.addTask("Build a Task Manager", "Develop a simple task  
manager in TypeScript.");
```

```
// Display tasks
```

```
console.log("All Tasks:", taskManager.getTasks());
```

```
// Mark a task as complete
```

```
const taskIdToComplete = 1;
```

```
if (taskManager.markTaskAsComplete(taskIdToComplete)) {  
  console.log(`Task ${taskIdToComplete} marked as complete.`);  
} else {  
  console.log(`Task ${taskIdToComplete} not found.`);  
}
```

```
// Display updated tasks  
console.log("Updated Tasks:", taskManager.getTasks());
```

Step 7: Update package.json

Modify package.json and add build & start scripts:

json

CopyEdit

```
{  
  "name": "task-manager-project",  
  "version": "1.0.0",  
  "description": "A simple Task Manager built with TypeScript.",  
  "main": "dist/index.js",  
  "scripts": {  
    "build": "tsc",  
    "start": "node dist/index.js"  
  },  
  "dependencies": {},  
  "devDependencies": {  
    "typescript": "^4.5.5"  
  }  
}
```

Step 8: Compile TypeScript

Run:

sh

CopyEdit

npm run build

This will generate a dist/ folder with JavaScript files.

Step 9: Run the Project

Execute the compiled JavaScript file

npm start

Output

All Tasks: [

```
{ id: 1, title: 'Learn TypeScript', description: 'Understand TypeScript basics and usage.',  
completed: false },
```

```
{ id: 2, title: 'Build a Task Manager', description: 'Develop a simple task manager in  
TypeScript.', completed: false }
```

]

Task 1 marked as complete.

Updated Tasks: [

```
{ id: 1, title: 'Learn TypeScript', description: 'Understand TypeScript basics and usage.',  
completed: true },
```

```
{ id: 2, title: 'Build a Task Manager', description: 'Develop a simple task manager in  
TypeScript.', completed: false }
```

]

Step 10: Debugging (Optional)

If you want to debug the TypeScript files in VS Code:

1. Open Run & Debug in VS Code.
2. Add a launch.json file.
3. Set breakpoints in src/index.ts.
4. Run the debugger.

