

TaskCLI Documentation

Contents

1 Project Information	2
1.1 Key Features	2
1.2 Technologies Used	2
1.3 Installation Requirements	2
1.4 Project Structure	2
2 Comprehensive README	2
2.1 Features	3
2.2 Technologies	3
3 Installation	3
3.1 Prerequisites	3
3.2 Setup Steps	3
4 Usage Guide	4
4.1 Creating a Task	4
4.2 Viewing Tasks	4
4.3 Updating Task Status	4
4.4 Viewing Statistics	4
5 Step-by-Step Task Lifecycle Guide	4
5.1 Step 1: Create a Task	4
5.2 Step 2: Find the Task ID	4
5.3 Step 3: Update Progress	4
5.4 Step 4: Mark as Done	4
6 Common Mistakes	5
7 Troubleshooting	5
8 Frequently Asked Questions (FAQ)	5
8.1 Getting Started	5
8.2 Features	5
8.3 Advanced Usage	5

1 Project Information

Project Name: TaskCLI

Description: TaskCLI is a command-line task management tool designed for developers who prefer managing their workflow directly from the terminal.

1.1 Key Features

- Create, update, and track tasks from the terminal
- Organize tasks into projects and tags
- Set priorities and deadlines
- Daily summary emails
- GitHub integration
- Time tracking

1.2 Technologies Used

- Node.js
- Commander.js
- Inquirer.js
- SQLite

1.3 Installation Requirements

- Node.js 10+
- npm

1.4 Project Structure

```
/bin          - CLI entry point
/commands     - Command implementations
/models       - Data models
/utils        - Utility functions
/db           - Database handlers
/services    - External service integrations
/config      - Configuration files
```

2 Comprehensive README

Task Manager CLI is a robust local command-line interface for managing tasks, built with Node.js. It allows users to organize work, track deadlines, set priorities, and view statistics directly from the terminal, with all data stored locally.

2.1 Features

- Task creation, update, and deletion
- Priority assignment from Low to Urgent
- Status workflow: `todo` → `in_progress` → `review` → `done`
- Due dates and overdue tracking
- Tag-based organization
- Search and filtering
- Productivity statistics
- Persistent local storage

2.2 Technologies

- Runtime: Node.js
- CLI Framework: Commander.js
- ID Generation: UUID
- Storage: Local JSON file

3 Installation

3.1 Prerequisites

- Node.js (v14 or higher recommended)
- npm

3.2 Setup Steps

1. Clone the repository:

```
git clone <repository-url>
cd task-manager-cli
```

2. Install dependencies:

```
npm install commander uuid
```

3. Run the CLI:

```
node cli.js --help
```

4 Usage Guide

4.1 Creating a Task

Basic Command

```
node cli.js create "Finish project documentation"
```

Advanced Command

```
node cli.js create "Buy groceries" -d "Milk, Eggs, Bread" -p 3 -u  
2023-12-25 -t "personal, shopping"
```

4.2 Viewing Tasks

```
node cli.js list
```

4.3 Updating Task Status

```
node cli.js status <task_id> in_progress  
node cli.js status <task_id> done
```

4.4 Viewing Statistics

```
node cli.js stats
```

5 Step-by-Step Task Lifecycle Guide

5.1 Step 1: Create a Task

```
node cli.js create "Submit Q3 Report" -d "Include financial summary" -p  
4 -u 2023-11-30 -t "work, finance"
```

5.2 Step 2: Find the Task ID

```
[ ] a1b2c3d4 - !!!! Submit Q3 Report  
Include financial summary  
Due: 2023-11-30 | Tags: work, finance  
Created: 2023-11-15
```

5.3 Step 3: Update Progress

```
node cli.js status a1b2c3d4 in_progress  
node cli.js tag a1b2c3d4 urgent
```

5.4 Step 4: Mark as Done

```
node cli.js status a1b2c3d4 done
```

6 Common Mistakes

- Incorrect date format (must be YYYY-MM-DD)
- Invalid status names (use exact values)
- Forgetting task IDs (use list command)

7 Troubleshooting

Problem	Cause	Solution
Task not found	Incorrect ID	Run list and copy correct ID
Invalid date	Wrong date format	Use YYYY-MM-DD
Changes not saving	Permission issues	Check file permissions
App crashes	Corrupt JSON file	Delete tasks.json (data loss)

8 Frequently Asked Questions (FAQ)

8.1 Getting Started

Q: How do I install TaskCLI? A: Install Node.js, navigate to the project folder, and run `npm install`.

Q: Where are tasks stored? A: Tasks are stored locally in a `tasks.json` file.

Q: Does it require internet access? A: No. The application runs fully offline.

8.2 Features

Q: How do priorities work? A:

- Low (1): !
- Medium (2): !!
- High (3): !!!
- Urgent (4): !!!!

Q: How do I mark a task as complete? A: Use `node cli.js status <task_id> done`.

Q: Can I add tags later? A: Yes, using `node cli.js tag <task_id> <tag>`.

8.3 Advanced Usage

Q: Can I update multiple fields at once? A: No. Each property must be updated with a separate command.

Q: How are task IDs generated? A: IDs are generated using UUID v4. The CLI displays the first 8 characters.