

# HowToDoit

## A simple to-do list app written in java

---

### 1. Introduction

HowToDoit is a to-do list app written in java. The user will be able to interact with a virtual task manager (via command line) that is able to load data from and save data into local history (using Java's Serializable interface).

### 2. Statement of scope

The app will be built using Gradle to a JAR file and run in JRE 11. It currently only has a CLI (command line interface). We will add a GUI in later phases of the project if time allows.

### 3. Architectural and component-level design

User:

- A user has a username and a password, which the user can change. Users must register or login to use our program.
- Each user has a collection of teams they belong to, a collection of upcoming tasks, and a collection of projects at hand.

Team:

- A team has a collection of users and knows which users are admins.
- The user can create a team, which they will join automatically as an admin, or leave a team.
- Only admins can complete the following actions: change team name, add teammates, promote teammates to admins, delete team, and assign tasks to other teammates.
- Normal teammates can only complete tasks assigned to them.

Task:

- The user can create tasks with a (unique) name, a due date, and a project it belongs to.
- Tasks can be modified (name, due date, description) and completed.

Project:

- A project is basically a directory that stores tasks; a task must be in exactly one project.
- There are two default projects called "General" and "Assigned to me" which cannot be modified or deleted.
- The user can create projects with a (unique) name. User-created projects can be modified (name, tasks) and deleted. When deleted, all tasks stored inside this project will also be deleted.

### 4. Use cases and user interface design

The user can use the following (case-sensitive) commands to interact with the system. Note that arguments are separated by the character ";".

General commands:

- *register;**<username>;<password>* Add new user with given *<username>* and *<password>* and automatically log in.
- *login;**<username>;<password>* Check whether *<username>* matches *<password>*.
- *logout* Log out.
- *exit* Quit the program and save data.
- *save* Save data.
- *undo* Undo an action.
- *redo* Redo an action.

Commands that display information:

- *viewTeams* Show all teams that the user is in.
- *viewMemsInTeam;**<name>* Show all members in the team called *<name>*.
- *viewProjs* Show all projects for the user.
- *viewTasks* Show all upcoming tasks in all projects in chronological order for the user; starred tasks will have stars in front of them.
- *viewTasksInProj;**<name>* Show all tasks in the project called *<name>*.

Commands on user account:

- *modUsn;**<name>* Modify the username from its current name to *<name>*.
- *modPwd;**<pw1>;<pw2>* Modify the password from *<pw1>* to *<pw2>*.

Commands on projects:

- *newProj;**<proj name>* Create a project called *<proj name>*.
- *delProj;**<name>* Delete the project called *<name>* and delete all its tasks..
- *modProj;**<name1>;<name2>* Change a project's name from *<name1>* to *<name2>*.

Commands on teams:

- *newTeam;**<name>* Create a new team called *<name>* and join the team automatically as an admin.
- *delTeam;**<name>* Delete a team called *<name>*; only admins of the team can perform this action.
- *modTeam;**<name1>;<name2>* Rename the team from *<name1>* to *<name2>*; only admins of the team can perform this action.
- *addMem;**<team name>;<username>* Add a user called *<username>* to the team called *<teamname>*; only admins of the team can perform this action.
- *leaveTeam;**<team name>* Remove the current user from the team *<team name>*.
- *addAdmin;**<team name>;<username>* Promote the user called *<username>* to an admin of the team called *<teamname>*; only admins of the team can perform this action.

Commands on tasks:

- *newTask;**<task name>;<due date>;<proj name>* Create a new task called *<task name>* with *<due date>* and add it to a project called *<proj name>*; a valid example of *<time>* would be "2021-10-15"; the user can leave the *<proj name>* field blank to put the task in "General".
- *assignTask;**<team name>;<username>;<task name>;<due date>* Create a new task called *<task name>* for a teammate called *<username>* in a team called *<team name>* with *<due date>*.
- *completeTask;**<name>* Remove a task called *<name>* from its project.
- *star;**<name>* Star a task called *<name>*.

- *unstar;*<name> Unstar a task called <name>.
- *rename;*<name1>;<name2> Change the name of a task from <name1> to <name2>.
- *retime;*<name>;<time> Change the due date of a task called <name> to <time>.
- *redesc;*<name>;<desc> Change the description of a task called <name> to <desc>.