# User Guide

Libmgr is a desktop app for managing the inventory of libraries, optimised for use via a Command Line Interface.

View on GitHub

# User Guide

## Introduction

Libmgr is a desktop app for managing the inventory of libraries, optimised for use via a Command Line Interface (CLI). Designed for fast typists, it can help to augment the day-to-day tasks of a librarian and can help them to get tasks done in an efficient manner.

## Quick Start

1. Ensure that you have Java `11` or above installed (can be verified by opening a command prompt or terminal and running `java -version`).
2. Down the latest version of `libmgr.jar` from here.
3. Copy the file to the folder you want to use as the *home folder* for Libmgr.
4. Open a terminal/command prompt and run `java -jar libmgr.jar` to start the application.
5. You will be greeted with the banner

```
     _.-"\
 _.-"     \
 ,-"          \
( \            \
 \ \            \
  \ \            \
\ \         _.-;
 \ \    _.-"   :
  \ \,-"    _.-"
   \(   _.-"  libmgr
    `__"

Welcome to libmgr
Type 'help' to view available commands
```

6. View Features to see what you can do in libmgr

## Features

### Notes about command formats

> ℹ️ Words in `UPPER_CASE` are parameters to be supplied by the user.
> e.g. In `add b t/TITLE ...`, `TITLE` is a parameter that can be used like `add b t/Don Quixote ...`

> ℹ️ Parameters can be in any order
> e.g. `add a t/TITLE i/ID a/ARTIST d/DURATION` can be used as `add a a/ARTIST i/ID t/TITLE d/DURATION`

> ⚠️ Parameters must be separated with a space
> e.g. For `add b t/TITLE i/ID ...`, entering `add a t/The Great Gatsbyi/0125` is invalid

> ⚠ Command word must be in lower case e.g. For `add b t/TITLE i/ID ...`, entering `ADD a t/The Great Gatsby i/0125` is invalid

> ⚠ Attribute flags must be in lower case e.g. For `add b t/TITLE i/ID ...`, entering `ADD a T/The Great Gatsby I/0125` is invalid

## Add an item

Add a new item to the catalogue. Items can fall under one of four categories: Audio, Books, Magazines, Videos

### Add an audio item

Add a new audio item to the catalogue

Format: `add a t/TITLE i/ID a/ARTIST d/DURATION`

#### Usage Example:

`add a t/Thriller i/5920 a/Michael Jackson d/42:16`

Adds an audio item to the catalogue titled `Thriller` with ID of `5920` by `Michael Jackson` and a duration of `42:16`

#### Expected Output:

```
> add a t/Thriller i/5920 a/Michael Jackson d/42:16
  (+) Added new audio item to the catalogue
  [A] 5920 | AVAILABLE | Thriller | Michael Jackson | 42:16
```

### Add a book item

Add a new book item to the catalogue

Format: `add b t/TITLE i/ID a/AUTHOR`

#### Usage Example:

`add b t/To Kill a Mockingbird i/2551 a/Harper Lee`

Adds a book item to the catalogue titled `To Kill a Mockingbird` with an ID of `2551`, authored by `Harper Lee`

#### Expected Output:

```
> add b t/To Kill a Mockingbird i/2551 a/Harper Lee
  (+) Added new book item to the catalogue
  [B] 2551 | AVAILABLE | To Kill a Mockingbird | Harper Lee
```

### Add a magazine item

Add a new magazine item to the catalogue

Format: `add m t/TITLE i/ID p/PUBLISHER e/EDITION`

#### Usage Example:

`add m t/Time Magazine i/58720a p/Time USA e/oct252021`

Adds a magazine item to the catalogue titled `Time Magazine` with an ID of `58720a`, published by `TimeUSA`, edition of `oct252021`

#### Expected Output:

```
> add m t/Time Magazine i/58720a p/Time USA e/oct252021
  (+) Added new magazine item to the catalogue
  [M] 58720a | AVAILABLE | Time Magazine | Time USA | oct252021
```

### Add a video item

Add a new video item to the catalogue

Format: `add v t/TITLE i/ID p/PUBLISHER d/DURATION`

#### Usage Example:

`add v t/Casino Royale i/095680 p/Sony Pictures d/144 minutes`

Adds a video item to the catalogue titled `Casino Royale` with an ID of `095680`, published by `Sony Pictures` and with a duration of `144 minutes`

#### Expected Output:

```
> add v t/Casino Royale i/095680 p/Sony Pictures d/144 minutes
  (+) Added new video item to the catalogue
  [V] 095680 | AVAILABLE | Casino Royale | Sony Pictures | 144 minutes
```

### Add a miscellaneous item

For any other item that cannot be categorised into the above-mentioned ones, they can be added using this command

Format: `add i t/TITLE i/ID`

Usage Example:

`add i t/Scrabble i/0513895`

Adds a miscellaneous item to the catalogue titled `Scrabble` with an ID of `0513895`

Expected Output:

```
> add i t/Scrabble i/0513895
  (+) Added new item to the catalogue
  [-] 0513895 | AVAILABLE | Scrabble
```

### List items

List out all items in the catalogue.

Format: `list`

Usage Example:

`list`

Expected Output:

```
  (+) Listing out all items in library
  ========================================
  [M] 58720a | AVAILABLE | Time Magazine | Time USA | oct252021
  [V] 095680 | AVAILABLE | Casino Royale | Sony Pictures | 144 minutes
  ========================================
```

### List items that are due today

List the loaned items that need to be returned today

Format: `deadline today`

Usage Example:

`deadline today`

Expected Output:

```
(+) Listing out loaned items that have to be returned today
========================================
[M] 58720a | LOANED (martin TILL 31-10-2021) | Time Magazine | Time USA | oct25
========================================
```

## List items that are overdue

List the loaned items that already overdue but have not been returned yet

Format: `deadline overdue`

Usage Example:

`deadline overdue`

Expected Output:

```
(+) Listing out loaned items that are overdue
========================================
[B] 2551 | LOANED (johnsmith TILL 01-01-2021) | To Kill a Mockingbird | Harper
========================================
```

## List items that are due on a specific date

List the loaned items that are due on a specific date given in the input

> ℹ `DUE_DATE` must be in the format of `dd-mm-yyyy` in order to be valid

Format: `deadline d/dd-mm-yyyy`

Usage Example:

`deadline d/03-11-2021`

List the items with a due date of `03 November 2021`

Expected Output:

```
(+) Listing out loaned items that are due this date: 03-11-2021
========================================
[M] 58720a | LOANED (Sam TILL 03-11-2021) | Time Magazine | Time USA | oct25202
========================================
```

## Search items

Search items by ID, Title, Status, and Category.

For searching by ID, the ID given by the search input must be exactly the same as the ID in database, that is, it is **case-sensitive**, and any difference in space or special characters is not allowed.

For searching by Status, the Status given by the search input must be one of “AVAILABLE”, “LOANED” or “RESERVED”. It is **case-insensitive**.

For searching by Category, the Category given by the search input must be one of “Book”, “Audio”, “Video”, “Magazine”, or “Misc”. It is **case-insensitive**.

For searching by Title, the Title given by the search input only needs to partially match the title stored in database. It is **case-insensitive** as well.

Any items matching more than one keyword will be listed. Items matching more searching criteria will be listed at a higher position. Items matching fewer searching criteria will be listed at a lower position.

Format: Subset of `search i/ID t/TITLE s/STATUS c/CATEGORY`

### Usage Example:

`search s/loaned c/book`

Searches the catalogue for items with the property of “loaned” and “book”. An item that is both loaned and is a book will be listed at the top. An item that is loaned but not a book, or an item that is a book but not loaned is also listed at a lower position. An item that is neither loaned nor a book will not be in the search result.

### Expected Output:

```
(+) Here are the searching results in library
========================================
[B] 43 | LOANED (Chuan Kai TILL 25-11-2021) | The Three Body Problem | Liu Cixi
[A] A A | LOANED (Silin TILL 30-12-2021) | Thriller | Michael Jackson | 2h
[B] 123 | AVAILABLE | Harry Potter | JK Rowling
========================================
```

In the sample output, “The Three Body Problem” is both loaned and a book, matching both of the criteria, it is listed first; “Thriller” is loaned but not a book, “Harry Potter” is a book but not loaned, they satisfy only one criterion, therefore they are listed at a lower position. The rest of items do not match any search keyword, and they are not listed.

## Reserve items

Reserve an item for a specific person.

Format: `res i/ID u/USERNAME`

#### Usage Example:

`res i/2551 u/johnsmith`

Sets aside an item with ID of `2551` for user `johnsmith`

#### Expected Output:

```
> res i/2551 u/johnsmith
  (+) You have successfully reserved an item:
  [B] 2551 | RESERVED (johnsmith) | To Kill a Mockingbird | Harper Lee
```

## Unreserve items

Make a previously reserved item available again.

Format: `unres ID`

#### Usage Example:

`unres 2551`

Makes a previously `RESERVED` item with ID of `2551` as `AVAILABLE`

#### Expected Output:

```
> unres 2551
  (+) Item unreserved:
  [B] 2551 | AVAILABLE | To Kill a Mockingbird | Harper Lee
```

## Loan items

Loan out an item to an individual until a specific due date.

> ℹ Only `AVAILABLE` items can be loaned out

> ℹ Items that have been previously reserved by an individual can only be loaned out to the same individual

> ℹ `DUE_DATE` must be in the format of `dd-mm-yyyy` in order to be valid

Format: `loan i/ID u/USER d/DUE_DATE(dd-mm-yyyy)`

Usage Example:

`loan i/2551 d/12-11-2021 u/johnsmith`

Mark item with ID of `2551` as `LOANED` to a user `johnsmith`, due to be returned on `12 November 2021`

Expected Output:

```
> loan i/2551 d/12-11-2021 u/johnsmith
  (+) Item has been loaned out:
  [B] 2551 | LOANED (johnsmith TILL 12-11-2021) | To Kill a Mockingbird | Harper
```

## Return items

Mark a previously loaned item as returned, making it available again.

> ℹ Only `LOANED` items can be returned. Unsuccessful message will be displayed when you return items that are not loaned out.

Format: `return ID`

Usage Example:

`return 2551`

Mark item with ID of `2551` which was previously `LOANED` as `AVAILABLE`

Expected Output:

```
> return 2551
  (+) Item has been returned:
  [B] 2551 | AVAILABLE | To Kill a Mockingbird | Harper Lee
```

## Edit an item

Edit the attributes of an existing item in the catalogue

> ℹ️ Multiple attributes can be inputted, in any order.
> e.g. For an existing book with ID `123`, title `Harry Potter` and author `JK Rowling`, it is possible to edit both the title and author simultaneously using `edit 123 t/NEW TITLE a/NEW AUTHOR`.

> ℹ️ Attributes cannot be changed to the same existing attribute.
> e.g. If the title of an existing book with ID 123 is already `Harry Potter`, calling `edit 123 t/Harry Potter` will throw a warning.

> ℹ️ Trying to change the id of the item into an id that is already associated with another existing item in the library will throw a warning and not be allowed.

Format: `edit ID KEY/ATTRIBUTE`

- For Book, valid key/attribute(s) are: t/TITLE, i/ID and a/AUTHOR
- For Audio, valid key/attribute(s) are: t/TITLE, i/ID, a/ARTIST and d/DURATION
- For Magazine, valid key/attribute(s) are: t/TITLE, i/ID, p/PUBLISHER and e/EDITION
- For Video, valid key/attribute(s) are: t/TITLE, i/ID, p/PUBLISHER and d/DURATION
- For Miscellaneous item, valid key/attribute(s) are: t/TITLE and i/ID

Usage Example:

`edit 123 t/Harry Potter a/JK Rowling`

Edits the title of an item with ID 123 to Harry Potter, and the author to JK Rowling

Expected Output:

```
E.g.
  [B] 123 | AVAILABLE | The Hunger Games | Suzanne Collins
```

```
> edit 123 t/Harry Potter a/JK Rowling
  (+) Edited item details:
  [B] 123 | AVAILABLE | Harry Potter | JK Rowling
```

## Remove an item

Remove an existing item from the catalogue.

Format: `rm ID`

- The item specified by `ID` will be deleted.

#### Usage Example

`rm 095680`

Removes item with ID of `095680` from the catalogue

#### Expected Output:

```
> rm 095680
  (+) Removed the following item:
  [V] 095680 | AVAILABLE | Casino Royale | Sony Pictures | 144 minutes
```

## Display library information

Display library information: total number of items in library, number of items by category and number of items by status.

### Display all information

Display all library information

Format: `info all`

#### Usage Example:

`info all`

Expected output:

```
E.g.
  [B] 123 | AVAILABLE | Harry Potter | JK Rowling
  [A] 124 | LOANED (John TILL 07-11-2021) | CD | Michael Jackson | 2h
```

```
> info all
  (+) Total Number of Items in Library: 2

  (+) Library Information by Item Category
  ========================================
  (+) Number of Audio Items: 1
  (+) Number of Book Items: 1
  (+) Number of Magazine Items: 0
  (+) Number of Video Items: 0
  (+) Number of Miscellaneous Items: 0

  (+) Library Information by Item Status
  ========================================
  (+) Number of Available Items: 1
  (+) Number of Loaned Items: 1
  (+) Number of Reserved Items: 0
```

### Display information by item category

Display total number of items in the library and the breakdown of items by category.

Format: `info category`

Usage Example:

`info category`

Expected output:

```
E.g.
  [B] 123 | AVAILABLE | Harry Potter | JK Rowling
  [A] 124 | LOANED (John TILL 07-11-2021) | CD | Michael Jackson | 2h
```

```
> info category
  (+) Total Number of Items in Library: 2

  (+) Library Information by Item Category
  ==========================================
  (+) Number of Audio Items: 1
  (+) Number of Book Items: 1
  (+) Number of Magazine Items: 0
  (+) Number of Video Items: 0
  (+) Number of Miscellaneous Items: 0
```

### Display information by item status

Display total number of items in the library and the breakdown of items by status.

Format: `info status`

Usage Example:

`info status`

Expected output:

```
E.g.
  [B] 123 | AVAILABLE | Harry Potter | JK Rowling
  [A] 124 | LOANED (John TILL 07-11-2021) | CD | Michael Jackson | 2h
```

```
> info status
  (+) Total Number of Items in Library: 2

  (+) Library Information by Item Status
  ========================================
  (+) Number of Available Items: 1
  (+) Number of Loaned Items: 1
  (+) Number of Reserved Items: 0
```

## Display valid user commands

Display a list of valid user commands for Libmgr.

Format: `help`

Usage Example:

`help`

Expected output:

```
> help
  ** Words in `UPPER_CASE` are parameters to be supplied by you
  ** Parameters can be in any order
  ** Parameters must be separated with a space
  (+) Add a new audio item: add a t/TITLE i/ID a/ARTIST d/DURATION
  (+) Add a new book item: add b t/TITLE i/ID a/AUTHOR
  (+) Add a new magazine item: add m t/TITLE i/ID p/PUBLISHER e/EDITION
  (+) Add a new video item: add v t/TITLE i/ID p/PUBLISHER d/DURATION
  (+) Add a new miscellaneous item: add i t/TITLE i/ID
  (+) List out all items: list
  (+) List out loaned items due today: deadline today
  (+) List out loaned items that are overdue: deadline overdue
  (+) Search items by ID, title, status or category: search [t|i|s|c]/ATTRIBUTE
  (+) Reserve an item: res i/ID u/USERNAME
  (+) Unreserve an item: unres ID
  (+) Loan out an item: loan i/ID u/USER d/DUE_DATE(dd-mm-yyyy)
  (+) Return a loaned item: return ID
  (+) Edit details of an existing audio item: edit ID [t|i|a|d]/ATTRIBUTE
  (+) Edit details of an existing book item: edit ID [t|i|a]/ATTRIBUTE
  (+) Edit details of an existing magazine item: edit ID [t|i|p|e]/ATTRIBUTE
  (+) Edit details of an existing video item: edit ID [t|i|p|d]/ATTRIBUTE
  (+) Edit details of an existing miscellaneous item: edit ID [t|i]/ATTRIBUTE
  (+) Remove an existing item: rm ID
  (+) Display all library information: info all
  (+) Display library information by item category: info category
  (+) Display library information by item status: info status
```

```
    (+) Get a list of valid user commands: help
    (+) Exit the program: exit
```

### Exit the program

Exit the program.

Format: `exit`

Usage Example:

`exit`

Expected Output:

```
> exit
See you soon!
```

## Data Storage

Upon starting the program, it will search for the presence of a file in the directory `./data/data.json`. This is a file on the disk that tracks the last recorded state of the catalogue.

- If the file exists, `libmgr` will attempt to read and parse all information within to load into the catalogue before completing the startup process
- If the file does not exist, `libmgr` will create a new file with an empty catalogue
- If the file exists, but the content is malformed or corrupted, users will be warned. Any commands input by the user after startup will result in the file being wiped with a new empty catalogue.

> ℹ️ Data is updated at the completion of execution of each command entered by the user

> ⚠️ While the `data.json` file can be edited manually, it is highly recommended that users **do not** attempt to do so unnecessarily

> ⚠️ **Do not** make changes to the `data.json` while `libmgr` is running. Any changes made while running will not be reflected by `libmgr` and such changes may be overwritten.

## FAQ

**Q**: What platforms are `Libmgr` supported on?

**A**: This application supports cross-platform operation, as long as Java has been properly configured.

**Q**: Can I port the program over to other devices

**A**: Yes, just copy over the `libmgr.jar` along with the `./data/data.json` files to the target computer, and make sure that they are stored within the same folder, all data will be preserved

**Q**: Can I manually edit the `data.json` file to change contents of the catalogue

**A**: While it is possible for users to manually edit the data using tools such as text editors, we highly recommend against doing so, as the risk of corrupting the file or entering malformed input is very high. When `libmgr` detects errors or corruptions in `data.json` it will overwrite the existing catalogue with a new, empty one.

## Command Summary

| Action | Purpose | Format and Examples |
|---|---|---|
| Add (Audio) | Add an audio item to the catalogue | `add a t/TITLE i/ID a/ARTIST d/DURATION`<br>E.g. `add a t/Thriller i/5920 a/Michael Jackson d/42:16` |
| Add (Book) | Add a book item to the catalogue | `add b t/TITLE i/ID a/AUTHOR`<br>E.g. `add b t/To Kill a Mockingbird i/2551 a/Harper Lee` |
| Add (Magazine) | Add a magazine item to the catalogue | `add m t/TITLE i/ID p/PUBLISHER e/EDITION`<br>E.g. `add m t/Time Magazine i/58720a p/Time USA e/oct252021` |
| Add (Video) | Add a video item to the catalogue | `add v t/TITLE i/ID p/PUBLISHER d/DURATION`<br>E.g. `add v t/Casino Royale i/095680 p/Sony Pictures d/144 minutes` |
| Add (Miscellaneous) | Add an item that cannot be categorised under any type | `add i t/TITLE i/ID`<br>E.g. `add i t/Scrabble i/0513895` |
| Deadline | View items due to be returned | `deadline [overdue/today/d/dd-mm-yyyy]`<br>E.g. `deadline today` |

| Action | Purpose | Format and Examples |
|---|---|---|
| Edit | Edit existing items within the catalogue | `edit ID KEY/ATTRIBUTE`<br>E.g. `edit 123 t/Harry Potter a/JK Rowling` |
| Exit | Exit the program | `exit` |
| Help | Display a list of valid user commands | `help` |
| Info | Display library information | `info [all/category/status]`<br>E.g. `info all` |
| List (all items) | Lists all items within the catalogue | `list` |
| Loan | Loan out an item to someone | `loan i/ID u/USER d/DUE_DATE(dd-mm-yyyy)`<br>E.g. `loan i/2551 d/12-11-2021 u/johnsmith` |
| Remove | Remove an item from the catalogue | `rm ID`<br>E.g. `rm 095680` |
| Reserve | Reserve an item for a specific person | `res i/ID u/USERNAME`<br>E.g. `res i/2551 u/johnsmith` |
| Return | Mark an item as returned and available for loan again | `return ID`<br>E.g. `return 2551` |
| Search | Search for items in the catalogue based on their attributes | `search KEY/ATTRIBUTE`<br>E.g. `search c/misc t/Junglebook s/available` |
| Un-reserve | "Un-reserve" an item and mark as available again | `unres ID`<br>E.g. `unres 2551` |

**tp is maintained by AY2122S1-CS2113-T16-1.**

This page was generated by GitHub Pages.