



ACTORBASE

SPREADING ACTORS ALL OVER THE WORLD

User Manual

Document Information

Version	1.0.0
Editing	Alberto De Agostini
Verification	Marco Boseggia
Approval	Michael Munaro
Use	External
Distribution list	ScalateKids Prof. Tullio Vardanega Prof. Riccardo Cardin



History log

Version	Author	Role	Date	Description
1.0.0	Michael Munaro	Project Manager	2016-05-09	Document approved
0.1.0	Marco Boseggia	Verifier	2016-05-07	Verified CLI section
0.0.2	Alberto De Agostini	Programmer	2016-05-04	Added CLI section
0.0.1	Alberto De Agostini	Programmer	2016-05-04	Created document structure



Contents

1 Summary	1
1.1 Document purpose	1
1.2 Product purpose	1
2 Actorbase by Command Line Interface	2
2.1 DSL	2
2.2 Error management	2
2.3 Password requirements	2
2.4 Configuration of the actorbase CLI	2
2.5 Operations	2
2.5.1 Standard user operations	3
2.5.1.1 Exit from CLI	3
2.5.2 Unauthenticated user	3
2.5.2.1 Login	3
2.5.2.2 Help - generic help	3
2.5.3 Authenticated user	3
2.5.3.1 Help - generic help	3
2.5.3.2 Help - specific help	4
2.5.3.3 Create a collection	4
2.5.3.4 List collections	4
2.5.3.5 Rename a collection	4
2.5.3.6 Delete a collection	5
2.5.3.7 Add a contributor to a collection	5
2.5.3.8 Remove contributor from a collection	5
2.5.3.9 Export one or more collections to file	5
2.5.3.10 Insert item	6
2.5.3.11 Import from file	6
2.5.3.12 Remove item	6
2.5.3.13 Find item	7
2.5.3.13.1 Search items in one or more collections	7
2.5.3.13.2 Search items in the entire database	7
2.5.3.13.3 Search for all the items of one or more collections	7
2.5.3.13.4 Get all database items grouped by collection	8
2.5.3.14 Change password	8
2.5.3.15 Logout	8
2.5.4 Administrator user	8
2.5.4.1 Add user	8
2.5.4.2 Remove user	8
2.5.4.3 Reset password	9
A JSON file format for import	10



1 Summary

1.1 Document purpose

This document represents the user manual for the *actorbase* application and describes in detail every single feature of it. Actorbase is a database that comes with two different software modules capable of interacting with it; the user can use both these modules, so this document will present them in two different sections.

1.2 Product purpose

The purpose of the project is the realization of a key-value type NoSQL¹ database, massively concurrent and oriented to the management of heavy load of data by using the actor² model.

¹A NoSQL database provides a mechanism for storage and retrieval of data which is modeled in means other than the tabular relations used in relational databases. Key-value (KV) stores use the associative array (also known as a map or dictionary) as their fundamental data model.

²A computational entity that, in response to a message it receives, can send messages to other actors, create new ones, change its behaviour for the next message incoming



2 Actorbase by Command Line Interface

2.1 DSL

The DSL (Domain System Language), upon which the *actorbase* CLI (Command Line Interface) is based, has been developed trying to emulate a SQL language. By doing so we achieved an easily understandable language that's close to spoken English.

2.2 Error management

There are two kinds of errors that are shared by every command which will be now explained:

- In case of an unrecognized command, or in case of a typographical error, the CLI will show an unrecognized command error followed by a list of all the commands available to the user;
- In case of a connection error, which can be caused by not receiving a response from the server in a pre-determined time, the CLI will show a timeout connection error.

There are some errors that are specific to some commands, these will be explained in the section dedicated to the commands that cause them.

2.3 Password requirements

Every password must contain at least one uppercase, one lowercase letter and one digit and be at least 8 characters long.

2.4 Configuration of the actorbase CLI

The *actorbase* CLI is designed specifically to operate with the actorbase server software.

To work properly an actorbase server instance must be running (anywhere) and the user must configure the access point on the configuration file.

2.5 Operations

The following sections requires the actorbase CLI to be configured and running (see [Configure CLI](#)).

The features described are divided by the type of the user.



2.5.1 Standard user operations

2.5.1.1 Exit from CLI

Description: every user can exit from the actorbase² CLI using this procedure.

How to: type exit on the CLI

2.5.2 Unauthenticated user

2.5.2.1 Login

Description: this is the procedure an unauthenticated user must follow to authenticate.

How to:

- Type login <your username>
- The CLI will ask you to insert your password (it will be masked during the typing)
- Type <your password>
- The CLI will display a message notifying the user if the login was successfull or not

Possible errors: the CLI will answer with self explanatory error messages

- If the credentials you entered are not recognized by the server

2.5.2.2 Help - generic help

Description: this feature allows the user to ask for help.

The CLI will answer with a list of all the possible operations alongside their right syntax.

The list of functionalities differs with the user type (e.g. authenticated user has more features than an unauthenticated one).

How to:

- Type help
- The CLI will display a help message

2.5.3 Authenticated user

2.5.3.1 Help - generic help

See [hyperref\[sec:generichelp\]](#)Generic help



2.5.3.2 Help - specific help

Description: this feature allows the authenticated user to ask for help for a specific command. The CLI will answer with a message explaining how to use that command and which parameters it needs.

How to:

- Type `help <command name>`
- The CLI will display a message explaining how to use the command and his syntax

Possible errors: the CLI will answer with self explanatory error messages

- If the command name the user typed is not a command of the actorbase CLI

2.5.3.3 Create a collection

Description: this feature allows the user to create a collection³ into the database.

How to:

- Type `createCollection <collection name>`
- The CLI will display a message explaining how to use the command and what parameters it needs

Possible errors: the CLI will answer with self explanatory error messages

- If the collection³ the user wants to create is already in the server the CLI will answer with a message

2.5.3.4 List collections

Description: this feature allows the user to see a list of the ³ he has permission on.

How to:

- Type `listCollection`
- The CLI will display a list of all the ³ the user has permission on

2.5.3.5 Rename a collection

Description: this feature allows the user to rename a collection³ of his own.

How to:

- Type `renameCollection <old collection name> to <new collection name>`

Possible errors: the CLI will answer with self explanatory error messages

- If the old collection³ name the user typed correspond with no collection³ inside the actorbase server
- If the new collection³ name the user typed is already taken by a collection³ inside the actorbase server

³A data-structure constituted by a list of key-value items



2.5.3.6 Delete a collection

Description: this feature allows the user to remove a collection³ and all the items inside it from the database.

How to:

- Type `deleteCollection <collection name>`

2.5.3.7 Add a contributor to a collection

Description: this feature allows the user to add a contributor to a collection³ of his own. The contributor will have the permission to operate on the selected collection³ depending on the what permissions the user decide to give him.

How to:

- Type `addContributor <username> <permissions> to <collection name>`

Possible errors: the CLI will answer with self explanatory error messages

- If the collection³ the user wants to add the contributor to does not exists on the server or it is not a collection³ the user own
- If the username the user typed does not correspond with a user registered in actorbase
- If the permissions the user typed are not possible permissions

2.5.3.8 Remove contributor from a collection

Description: this feature allows the user to remove a contributor from a collection³ of his own.

How to:

- Type `removeContributor <username> from <collection name>`

Possible errors: the CLI will answer with self explanatory error messages

- If the collection³ the user wants to remove the contributor from does not exists on the server or it is not a collection³ the user owns
- If the username the user typed does not correspond with a user registered in actorbase

2.5.3.9 Export one or more collections to file

Description: this feature allows the user to export one or more³ he has access to.

The filename the user insert can be a relative or absolute path; if the path is not present, the file will be created at the user local work directory.

The collection³ list can contain zero or more³ name, if empty it will export all the³ the user has access to (possibly resulting in an expensive operation). **How to:**

- Type `export <collection name1>, <collection name2>, ... , to <filename>`

Possible errors: the CLI will answer with self explanatory error messages



- If at least one of the ³ name the user inserted does not exist on the server
- If the filename the user inserted contains a path that is not valid

The produced JSON file follow some rules explained in [JSON format](#).

2.5.3.10 Insert item

Description: this feature allows the user to insert an item in a particular collection³.

The last parameter (o) can be omitted; if omitted the insert operation will not overwrite existing items with the same key. It will overwrite them otherwise.

The <key> is a String representing the key of the item the user wants to insert.

The <value> can be anything the user wants to insert and it represent the value of the item itself.

If the user wants to insert the item in a non existing collection³ the server will automatically create that collection³.

How to:

- Type insert (<key> -> <value>) to <collection name> o

Possible errors: the CLI will answer with self explanatory error messages

- If the collection³ the user wants to insert the item in does not exist or user has not permission to write on it
- If the user decides to insert an item with an already existing key in that collection³ and the overwrite option has not been chosen.

2.5.3.11 Import from file

Description: this feature allows the user to insert a list of ³ and items from a file.

The file has to be formatted as specified in (see [JSON format](#))

How to:

- Type import <path/to/filename>

Possible errors: the CLI will answer with self explanatory error messages

- If the file is not a JSON file
- If the file is not formatted as specified in [JSON format](#)
- If the file specified the creation of collection³ that the user can not create (see [create collection](#))
- If the file specified the insertion of items that the user can not insert (see [JSON format](#))

2.5.3.12 Remove item

Description: this feature allows the user to remove an item from a collection³.

<key> represents the key of the item the user wants to remove from the collection³ **How to:**

- Type remove <key> from <collection name>



Possible errors: the CLI will answer with self explanatory error messages

- If the collection³ the user typed does not exist in the server or if the user does not have permission to remove items from

2.5.3.13 Find item

Description: this feature allows the user to search for particular items in the database. There are different types of search based on the parameters typed.

2.5.3.13.1 Search items in one or more collections

Description: this feature lets the user search for an item in one or more ³; the user must have the permission to read these ³.

How to:

- Type find <key> from <collection name 1>, <collection name 2>, ... , <collection name N>

Possible errors: the CLI will answer with self explanatory error messages

- If at least one of the ³ does not exist on the server

2.5.3.13.2 Search items in the entire database

Description: this feature lets the user search for an item in the entire database visible to him.

How to:

- Type find <key>

2.5.3.13.3 Search for all the items of one or more collections

Description: this feature lets the user retrieve all the items contained in one or more ³.

How to:

- Type find from <collection name 1>, <collection name 2>, ... , <collection name N>

Possible errors: the CLI will answer with self explanatory error messages

- If at least one of the collection³ does not exist on the server or if the user doesn't have access to it

2.5.3.13.4 Get all database items grouped by collection

Description: this feature lets the user to retrieve all the items contained in the section of the database that is visible to him grouped by collection³.

This operation could be expensive.

How to:



- Type find

2.5.3.14 Change password

Description: this feature allows the user to change his password.

How to:

- Type changePassword "<old password>" "<new password>"

Possible errors: the CLI will answer with self explanatory error messages

- If the old password typed was wrong
- If the new password typed doesn't follow the rules explained [here](#).

2.5.3.15 Logout

Description: this feature allows the user to logout from the server becoming an unauthenticated user.

How to:

- Type logout

2.5.4 Administrator user

2.5.4.1 Add user

Description: this feature allows the administrator to add a new user to the database with a default password.

How to:

- Type addUser <username>

2.5.4.2 Remove user

Description: this feature allows the administrator to remove an user from the database.

How to:

- Type removeUser <username>

2.5.4.3 Reset password

Description: this feature allows the administrator to change the password of an user to the one that was set by default.

How to:

- Type resetPassword <username>



A JSON file format for import

Exporting data from **Actorbase** generate a well-formatted JSON file, and the same format must be used for import data to filesystem, e.g.:

```
{
  owner: "Owner",
  collection: "foo",
  "permissions": "read-write",
  "map": {
    "bar": "baz",
    "foobar": 42,
    "fooList": ["list", "of", "strings"]
  }
}
```

Figure 1: JSON well-formatted file



List of Tables



List of Figures

1	JSON well-formatted file	10
---	------------------------------------	----