ACTORBASE
SPREADING ACTORS ALL OVER THE WORLD

User Manual

Document Information

Version	2.0.0
Editing	Alberto De Agostini
Verification	Michael Munaro
Approvation	Davide Trevisan
Use	External
Distribution list	ScalateKids
	Prof. Tullio Vardanega
	Prof. Riccardo Cardin

History log

Version	Author	Role	Date	Description
2.0.0	Davide Trevisan	Project Manager	2016-07-15	Document Approved
1.2.0	Michael Munaro	Verifier	2016-07-02	Verified changes on section 2 and 3
1.1.2	Alberto De Agostini	Programmer	2016-07-01	Added section Default Users
1.1.1	Alberto De Agostini	Programmer	2016-07-01	Modified section 2 to reflect changes on CLI
1.1.0	Michael Munaro	Verifier	2016-06-30	Verified glossary
1.0.1	Alberto De Agostini	Programmer	2016-05-30	Added glossary
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 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 $NoSQL_G$ database, massively concurrent and oriented to the management of heavy load of data by using the $actor_G$ model.

1.3 Glossary

Every technical word, some words that can be misunderstood and all of acronyms are defined in the glossary in the appendix **??**. Every word that is in the glossary will be marked with a subscript "G".

2   Default users

When *actorbase* is launched for the first time it creates two default users:

- **admin**: this user has admin privileges, he can add and remove users, reset the password of an user to "Actorb4se" and he is the only one who has access to all of the collections of the database;

- **anonymous**: this user is the default non-admin user. Everything that has been done by an anonymous user will not be persisted.

Both of these users will be created with the default password "Actorb4se". It is recommended for the admin to change his password after the first login.

3   Actorbase by Command Line Interface

3.1   DSL

The DSL (Domain System Language), upon which the *actorbase* CLI (Command Line Interface) is based, has been developed trying to make it as close as possible to spoken English in order to make it easier to understand.

3.2   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 predetermined 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.

3.3   Password requirements

Every password must contain at least one uppercase, one lowercase letter and one digit and be at least 8 characters long.
The default password of actorbase for users that are just added or after a password reset is `Actorb4se`.

3.4   Running and configurating the actorbase CLI

To run the Actorbase CLI some requirements needs to be fullfilled before starting it.
The user has to download and install all the software requirements (see link) and an instance of actorbase server must be running.
When all those requirements are fullfilled to start the CLI the use has to open a terminal and type:

```
java -jar <pathToActorbaseCLIJar> -u <username> -h <host> -p <port>
```

Where the username is the username with which you want to login in actorbase, host is the ip where actorbase is listening for requests and the port is the port where actorbase is listening for requests.
All those parameters can be omitted, if omitted the CLI will try to connect with the default user "anonymous" in the default host "localhost" with default port "9999".

3.5   Operations on collections not of your own

If the user is a contributor to a collection the syntax to make every operations that involves selecting one or more collections$_G$ is a little different. When the user has to insert the name of a collection$_G$ instead of just typing "`<collection name>`" the user has to insert "`<collection owner username>.<collection name>`".
This way the user tells actorbase to operate on a collection not of his own.
This is also to let the user to create a collection$_G$ with the same name of a collection$_G$ of which he has access to but it is not of his own.

3.6   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.

3.6.1   Acceptable values

Every operations that accepts parameters like the collection name or the key of an item has some limitations on the value:

- Keys and collection names must contains only alphanumeric characters.

3.6.2   Standard user operations

3.6.2.1   Exit from CLI

Description: every user can exit from the actorbase CLI using this procedure.
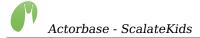**How to:** type `exit` on the CLI

3.6.2.2   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.
How to:

- Type `help`
- The CLI will display a help message

3.6.2.3   Help - specific help

Description: this feature allows the 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

3.6.2.4   Create a collection

Description: this feature allows the user to create a collection$_G$ 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 collection$_G$ the user wants to create is already in the server the CLI will answer with a message

3.6.2.5   List collections

Description: this feature allows the user to see a list of the collections$_G$ he has permission on.
How to:

- Type `listCollection`

- The CLI will display a list of all the collections$_G$ the user has permission on

3.6.2.6   Delete a collection

Description: this feature allows the user to remove a collection$_G$ and all the items inside it from the database.
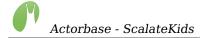**How to:**

- Type `deleteCollection "<collection name>"`

3.6.2.7   Add a contributor to a collection

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

- Type `addContributor "<username>" to "<collection name>" <ReadOnly|ReadWrite>`

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

- If the collection$_G$ the user wants to add the contributor to does not exists on the server or it is not a collection$_G$ 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 ReadOnly or ReadWrite

3.6.2.8   Remove contributor from a collection

Description: this feature allows the user to remove a contributor from a collection$_G$ of his own.
How to:

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

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

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

3.6.2.9   Export one or more collections to file

Description: this feature allows the user to export one or more collections$_G$ he has access to.
The filename the user insert can be a relative or absolute path; if the path is not present, it will be created.
The collection$_G$ list can contain zero or more collections$_G$' name, if empty it will export all the collections$_G$ the user has access to (possibly resulting in an expensive operation).
To export collections that you are contributor on see contributor operations.
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 collections$_G$' 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.
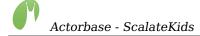
3.6.2.10   Insert item

Description: this feature allows the user to insert an item in a particular collection$_G$.
With this command it's not possible to overwrite an item, to do that see Update item.
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 collection$_G$ the server will automatically create that collection$_G$.
To insert in a collection that you are contributor on see contributor operations.
How to:

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

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

- If the collection$_G$ the user wants to insert the item in a collection$_G$ in which he is a contributor with no write premissions

- If the user decides to insert an item with an already existing key in that collection$_G$ and the overwrite option has not been chosen.

3.6.2.11   Update item

Description: this feature allows the user to insert an item in a particular collection$_G$.
With this command it's possible to overwrite an item already existing.
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 collection$_G$ the server will automatically create that collection$_G$.
To insert in a collections that you are contributor on see contributor operations.
How to:

- Type `update ( "<key>" -> "<value>" ) to "<collection name>"`

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

- If the collection$_G$ the user wants to insert the item in a collection$_G$ in which he is a contributor with no write premissions

3.6.2.12   Import from file

Description: this feature allows the user to insert a list of collections$_G$ and items from a file.
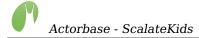The file has to be formatted as specified in (see JSON format)
How to:

- 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 collection$_G$ that the user can not create (see create collection)

- If the file specified the insertion of items that the user can not insert

3.6.2.13 Remove item

Description: this feature allows the user to remove an item from a collection$_G$.
<key> represents the key of the item the user wants to remove from the collection$_G$.
To remove items from collections that you are contributor on see contributor operations.
How to:

- Type `remove "<key>" from "<collection name>"`

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

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

3.6.2.14 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.
To search from collections that you are contributor on see contributor operations.

3.6.2.14.1 Search items in one or more collections

Description: this feature lets the user search for an item in one or more collections$_G$; the user must have the permission to read these collections$_G$.
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 collections$_G$ does not exist on the server or if the user has not read permission on some of the collections$_G$

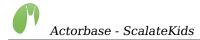##### 3.6.2.14.2 Search items in the entire database

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

- Type `find "<key>"`

3.6.2.14.3 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 collections$_G$.
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 collection$_G$ does not exist on the server or if the user doesn't have access to it

3.6.2.14.4 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 collection$_G$.
This operation could be expensive.
How to:

- Type `find`

3.6.2.15 Change password

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

- Type `changePassword`

- Type "`<old password>`"

- Type "`<new password>`"

- Type "`<repeat 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

- If the repeated new password is no the same as the new password typed before.

3.6.3 Administrator user operations

3.6.3.1 Add user

Description: this feature allows the administrator to add a new user to the database with a default password (see password section).
How to:

- Type `addUser` "`<username>`"

3.6.3.2 Remove user

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

- Type `removeUser "<username>"`

3.6.3.3 Reset password

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

- Type `resetPassword "<username>"`

3.6.3.4 List users

Description: this feature allows the administrator to get a list of all the users registrated in actorbase.
How to:

- Type `listUsers`

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.:

```json
{
 "owner": "Owner",
 "collectionName": "foo",
 "contributors":{
     "anotherUser": false,
     "thirdUser": true
 },
 "data": {
     "bar": "baz",
     "foobar": 42,
     "fooList": ["list","of","strings"]
 }
}
```

Figure 1: JSON well-formatted file

List of Figures

B   Glossary

B.1   A

Actor: A 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

B.2   C

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

B.3   N

NoSql: A 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.