

SWEENEYTHREADS

ACTORBASE

A NoSQL DB BASED ON THE ACTOR MODEL

---

# Manuale Utente

---

*Redattori:*

Maino Elia

Bortolazzo Matteo

Tommasin Davide

*Approvazione:*

Bonato Paolo

*Verifica:*

Bonato Paolo

Nicoletti Luca



Versione 1.0.0

16 maggio 2016

# Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
1.1	Scopo del documento . . . . .	3
1.2	Scopo del prodotto . . . . .	3
<b>2</b>	<b>Actorbase</b>	<b>3</b>
<b>3</b>	<b>Requisiti di sistema</b>	<b>4</b>
<b>4</b>	<b>Installazione</b>	<b>4</b>
<b>5</b>	<b>Applicativo server</b>	<b>5</b>
5.1	Configurazione del server . . . . .	5
5.2	Interfaccia server . . . . .	5
<b>6</b>	<b>Applicativo client</b>	<b>6</b>
6.1	Gestione della connessione . . . . .	6
6.1.1	Comando <code>connect</code> . . . . .	6
6.1.2	Comando <code>disconnect</code> . . . . .	6
6.2	Aiuto inline . . . . .	6
6.2.1	Comando <code>help</code> generico . . . . .	7
6.2.2	Comando <code>help</code> specifico . . . . .	7
6.3	Comandi per operazioni a livello server . . . . .	7
6.3.1	Comando <code>listdb</code> . . . . .	7
6.3.2	Comando <code>selectdb</code> . . . . .	7
6.3.3	Comando <code>createdb</code> . . . . .	7
6.3.4	Comando <code>deletedb</code> . . . . .	7
6.4	Comandi per operazioni a livello database . . . . .	8
6.4.1	Comando <code>listmap</code> . . . . .	8
6.4.2	Comando <code>selectmap</code> . . . . .	8
6.4.3	Comando <code>createmap</code> . . . . .	8
6.4.4	Comando <code>deletemap</code> . . . . .	8
6.5	Comandi per operazioni a livello mappa . . . . .	8
6.5.1	Comando <code>keys</code> . . . . .	9
6.5.2	Comando <code>find</code> . . . . .	9
6.5.3	Comando <code>remove</code> . . . . .	9
6.5.4	Comando <code>insert</code> . . . . .	9
6.5.5	Comando <code>update</code> . . . . .	9
6.6	Operazioni comandi amministratore . . . . .	9
6.6.1	<code>listuser</code> command . . . . .	10
6.6.2	<code>adduser</code> command . . . . .	10
6.6.3	<code>removeuser</code> command . . . . .	10
6.6.4	<code>listpermission</code> command . . . . .	10
6.6.5	<code>addpermission</code> command . . . . .	10
6.6.6	<code>removepermission</code> command . . . . .	10
6.6.7	<code>setmaxrows</code> command . . . . .	11
6.6.8	<code>setninja</code> command . . . . .	11
6.6.9	<code>setwarehouseman</code> command . . . . .	11
6.6.10	<code>setmaxstorekeeper</code> command . . . . .	11
6.6.11	<code>setmaxstorefinder</code> command . . . . .	11
	<b>Elenco delle figure</b>	<b>12</b>
	<b>Elenco delle tabelle</b>	<b>13</b>

## Diario delle modifiche

Versione	Data	Autore	Descrizione
1.0.0	2016-05-14	<i>Progettista</i> Bonato Paolo	Documento approvato.
0.2.0	2016-05-14	<i>Verificatore</i> Bonato Paolo	Documento verificato
0.1.2	2016-05-14	<i>Progettista</i> Bortolazzo Matteo	Aggiunti i comandi utente.
0.1.1	2016-05-13	<i>Progettista</i> Maino Elia	Apportate modifiche secondo lista di errori riscontrati durante la verifica
0.1.0	2016-05-13	<i>Verificatore</i> Nicoletti Luca	Effettuata la verifica del documento, e inviato il file contenente la lista degli errori trovati.
0.0.2	2016-05-08	<i>Progettista</i> Maino Elia	Aggiornamento della sezione di introduzione, stesura della sezione relativa all'applicativo server, incremento della sezione relativa all'applicativo client.
0.0.1	2016-05-05	<i>Progettista</i> Tommasin Davide	Stesura dei paragrafi 2.1 e 2.2 relativi ai requisiti hardware e software del prodotto. Elen- cato e specificato i vari comandi da poter uti- lizzare nel nostro prodotto nel paragrafo 3.1 e successivi
0.0.0	2016-05-02	<i>Progettista</i> Tommasin Davide	Creazione scheletro documento e stesura introduzione

Tabella 1: Diario delle modifiche

# 1 Introduzione

## 1.1 Scopo del documento

Questo documento rappresenta il manuale utente per l'utilizzo del database NoSQL *Actorbase*. In questo documento verranno descritte dettagliatamente tutte le caratteristiche dell'applicativo utilizzabili dall'utente. Il manuale è diviso in tre sezioni relative alle diverse componenti di *Actorbase*: Client, Server e Driver.

## 1.2 Scopo del prodotto

Lo scopo del progetto è la realizzazione di un DataBase NoSQL key-value basato sul modello ad Attori con l'obiettivo di fornire una tecnologia adatta allo sviluppo di moderne applicazioni che richiedono brevissimi tempi di risposta e che elaborano enormi quantità di dati. Lo sviluppo porterà al rilascio del software sotto licenza MIT.

# 2 Actorbase

*Actorbase* è un database NoSQL key-value basato sul modello matematico ad attori che garantisce un alto livello di scalabilità, resilienza e prestazioni. Permette di gestire facilmente e flessibilmente i propri dati usufruendo dei principali vantaggi offerti dal modello ad attori, in modo da supportare lo sviluppo di applicazioni moderne e performanti.

*Actorbase* fornisce un'interfaccia client a linea di comando che offre una facile gestione dei dati come stringhe, grazie ad essa è possibile comunicare rapidamente e intuitivamente con un server.

Per una gestione più flessibile delle query è disponibile il driver per *Scala*, integrabile con qualsiasi applicazione.



Figura 1: Logo di Actorbase

### 3 Requisiti di sistema

L'esecuzione corretta di *Actorbase* è garantita su macchine che soddisfano le seguenti caratteristiche hardware e software.

Sistema operativo:

- Windows 7 o superiori
- OS X 10.7 o superiori
- Ubuntu 14.04 o superiori

Java Virtual Machine (JVM) versione 8 o superiore.

RAM:

- Applicativo client: 2GB minimo
- Applicativo server: 4GB minimo, 8GB consigliato

Non sono presenti espliciti vincoli per quanto riguarda il processore, chiaramente l'utilizzo di hardware eccessivamente datato può influenzare le prestazioni del sistema.

### 4 Installazione

*Actorbase* viene eseguito su JVM, per tale motivo non è necessaria una procedura di installazione per utilizzare il software.

## 5 Applicativo server

L'applicativo server consente di avviare un'istanza *Actorbase* server sulla macchina. Una volta avviato permette ai client di effettuare connessioni alla macchina e di interrogare un database.

### 5.1 Configurazione del server

La configurazione della macchina server avviene tramite la modifica del file `server.conf`. In esso vanno impostati indirizzo IP e porta di connessione al server.

Nel caso si tentasse di avviare l'applicativo server senza aver definito opportunamente i parametri di configurazione, il server mostrerà un messaggio di errore e non si avvierà.

### 5.2 Interfaccia server

Una volta modificato il file di configurazione è possibile avviare il server cliccando sull'apposita icona. L'applicativo server fornisce all'utente un'interfaccia a linea di comando che mostra in tempo reale il log delle operazioni effettuate sul server *Actorbase*.

```
[DEBUG] [05/15/2016 16:08:11.343] [main] [EventStream(akka://System)] logger log2-Logging$DefaultLogger started
[DEBUG] [05/15/2016 16:08:11.344] [main] [EventStream(akka://System)] Default Loggers started
[INFO] [05/15/2016 16:08:11.390] [main] [Server$(akka://System)] Users loaded
[INFO] [05/15/2016 16:08:11.395] [main] [Server$(akka://System)] Permissions loaded
[INFO] [05/15/2016 16:08:11.399] [main] [Server$(akka://System)] Database loaded
[INFO] [05/15/2016 16:08:11.400] [main] [Server$(akka://System)] Server started
[INFO] [05/15/2016 16:08:11.413] [System-akka.actor.default-dispatcher-3] [akka://System/user/$b] Map users created
[INFO] [05/15/2016 16:08:11.426] [System-akka.actor.default-dispatcher-3] [akka://System/user/$b] Map permissions created
[INFO] [05/15/2016 16:08:11.431] [System-akka.actor.default-dispatcher-2] [akka://System/deadLetters] Message [Ljava.lang.String] from Actor[akka://System/
user/$b#416210096] to Actor[akka://System/deadLetters] was not delivered. [1] dead letters encoded.
This logging can be turned off or adjusted with configuration settings 'akka.log-dead-letters' and 'akka.log-dead-letters-during-shutdown'.
[INFO] [05/15/2016 16:08:11.432] [System-akka.actor.default-dispatcher-2] [akka://System/deadLetters] Message [Ljava.lang.String] from Actor[akka://System/
user/$b#416210096] to Actor[akka://System/deadLetters] was not delivered. [2] dead letters encountered. This logging can be turned off or adjusted with co
nfiguration settings 'akka.log-dead-letters' and 'akka.log-dead-letters-during-shutdown'.
[DEBUG] [05/15/2016 16:08:11.507] [System-akka.actor.default-dispatcher-4] [akka://System/system/IO-TCP/selectors/$a/0] Successfully bound to /127.0.0.1:8
181
[INFO] [05/15/2016 16:08:11.509] [System-akka.actor.default-dispatcher-5] [akka://System/user/$c] Port 8181 openedPlease connect first
```

Figura 2: Interfaccia di log del server

## 6 Applicativo client

L'applicativo client fornisce all'utente un'interfaccia a linea di comando per connettersi ad un server *Actorbase* ed effettuare interrogazioni sul database. Come per il server, l'avvio dell'interfaccia client avviene cliccando sull'apposita icona. Appena avviato il client presenta all'utente un banner di benvenuto seguito da una breve descrizione della configurazione software utilizzata (sistema operativo e JVM).

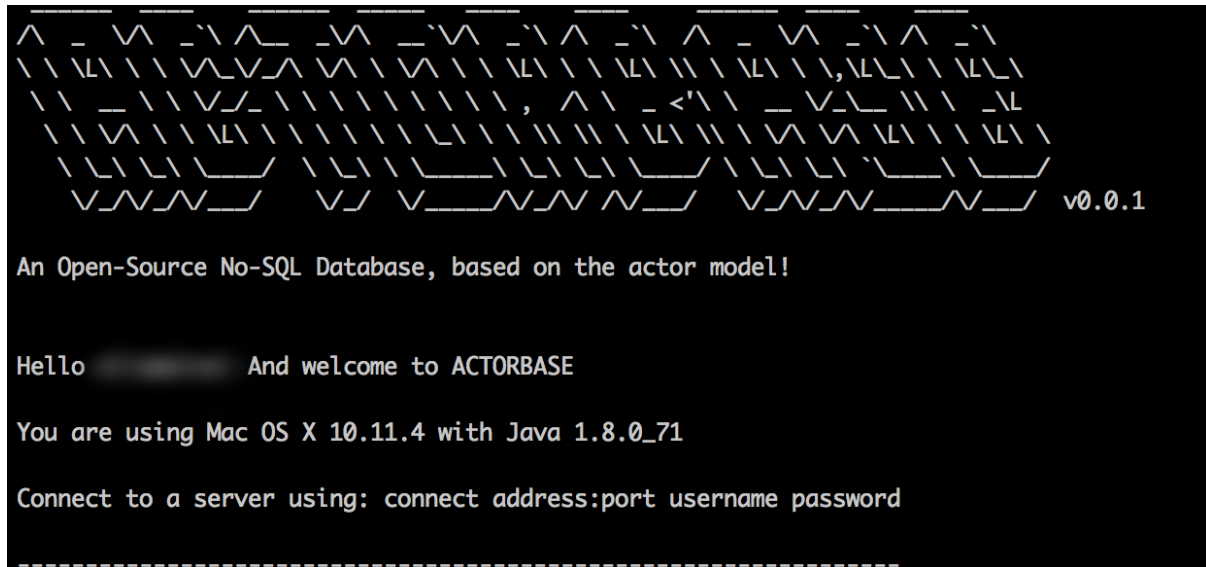


Figura 3: Informazioni di benvenuto applicativo client

L'interazione con l'interfaccia avviene tramite l'immissione di comandi testuali, essi possono essere composti da più campi separati da un carattere di spazio.

### 6.1 Gestione della connessione

Una volta avviato il client per prima cosa è necessario effettuare una connessione ad un server, la gestione della connessione si basa su due comandi: **connect** e **disconnect**. L'applicativo client fornito non permette di gestire più connessioni contemporaneamente.

#### 6.1.1 Comando connect

Questo comando permette all'utente di connettersi ad un server, la struttura del comando è la seguente:

```
actorbase> connect <indirizzo> <username> <password>
```

In particolare l'indirizzo del server deve essere fornito nel formato `indirizzoServer:porta`.

Nel caso la richiesta di connessione abbia successo l'utente riceve un messaggio di conferma ("You are connected!"), in caso contrario un messaggio di errore ("Connection failed!").

#### 6.1.2 Comando disconnect

Per effettuare la disconnessione dal server a cui si è connessi è necessario inserire il comando **disconnect** e premere invio.

### 6.2 Aiuto inline

È possibile ottenere un aiuto per quanto riguarda le operazioni effettuabili, direttamente a linea di comando. Il comando **help** permette di ottenere sia un aiuto generico che un aiuto specifico.

### 6.2.1 Comando help generico

Il comando di aiuto generico **help**, non richiede parametri ulteriori e stampa a video la lista dei comandi di *Actorbase*. Ogni comando è seguito da una breve descrizione che ne illustra il funzionamento.

### 6.2.2 Comando help specifico

Il comando di aiuto specifico presenta la seguente struttura:

```
actorbase> help <nomeComando>
```

Permette di richiedere informazioni per un comando specifico, ottenendone una breve descrizione.

## 6.3 Comandi per operazioni a livello server

Una volta connesso l'utente si trova a livello server. A tale livello si possono effettuare operazioni sui database quali:

- Visualizzazione della lista di database
- Selezione di un database
- Creazione di un database
- Rimozione di un database

### 6.3.1 Comando listdb

Questo comando permette di ottenere, stampata a video, la lista dei database di cui dispone di permessi di accesso (siano essi di visualizzazione o modifica). Tale comando non richiede parametri aggiuntivi.

```
actorbase> listdb
```

### 6.3.2 Comando selectdb

Questo comando permette di selezionare un database. Una volta selezionato un database l'utente può effettuare operazioni sulle mappe di esso. La struttura del comando di selezione database è la seguente:

```
actorbase> selectdb <nomeDatabase>
```

Nel caso l'utente disponga dei permessi necessari alla selezione del database richiesto (**x**), riceve a video un messaggio di conferma: "Database **x** selected". In caso contrario viene riportata un operazione non valida ("Invalid operation").

### 6.3.3 Comando createdb

Questo comando permette di creare un nuovo database col nome specificato, nel caso un database con tale nome non fosse già presente.

```
actorbase> createdb <nomeDatabase>
```

Nel caso esista già un database con il nome inserito la creazione fallisce, l'utente riceve un messaggio di errore: "A database with the requested name already exists".

### 6.3.4 Comando deletedb

Questo comando permette di eliminare un database dal server:

```
actorbase> deletedb <nomeDatabase>
```

Nel caso si tenti di rimuovere un database inesistente o di cui non si dispone dei permessi di modifica, l'utente riceve un messaggio di operazione non valida ("Invalid operation").



## 6.4 Comandi per operazioni a livello database

Una volta selezionato un database tramite il comando `selectdb`, l'utente si trova a livello database. A tale livello si possono effettuare queste operazioni:

- Visualizzazione della lista delle mappe che compongono il database
- Selezione di una mappa
- Creazione di una mappa
- Eliminazione di una mappa

### 6.4.1 Comando `listmap`

Questo comando stampa una lista di tutte le mappe che compongono il database precedentemente selezionato.

```
actorbase> listmap
```

### 6.4.2 Comando `selectmap`

Questo comando permette di selezionare una mappa, utilizzando la seguente sintassi:

```
actorbase> selectmap <nomeMappa>
```

La selezione viene confermata in caso di successo con il messaggio `"Map x selected"`, altrimenti si riceve un messaggio di operazione non valida (`"Invalid operation"`).

### 6.4.3 Comando `createmap`

Questo comando permette di creare una mappa con il nome inserito, all'interno del database selezionato.

```
actorbase> createmap <nomeMappa>
```

La creazione viene confermata in caso di successo con il messaggio `"Map x created"`. Nel caso non si disponga dei permessi di modifica al database, o esista già una mappa con il nome inserito, si riceve un messaggio di operazione non valida (`"Invalid operation"`).

### 6.4.4 Comando `deletemap`

Questo comando permette di eliminare una mappa dal database selezionato.

```
actorbase> deletemap <nomeMappa>
```

L'eliminazione viene confermata in caso di successo con il messaggio `"Map x removed"`. Nel caso non si disponga dei permessi di modifica al database, o la mappa richiesta non esista, si riceve un messaggio di operazione non valida (`"Invalid operation"`).

## 6.5 Comandi per operazioni a livello mappa

Una volta selezionata una mappa tramite il comando `selectmap`, l'utente si trova a livello mappa. A tale livello si possono effettuare queste operazioni:

- Visualizzazione della lista delle chiavi
- Ricerca di un item per chiave
- Inserimento di un item
- Aggiornamento di un item
- Rimozione di un item

### 6.5.1 Comando keys

Questo comando permette di visualizzare la lista di tutte le chiavi degli item che compongono la mappa selezionata.

```
actorbase> keys
```

### 6.5.2 Comando find

Questo comando permette di ottenere il valore di un item della mappa, effettuando la ricerca tramite la sua chiave.

```
actorbase> find '<key>'
```

Nel caso sia presente un item corrispondente alla chiave inserita, si riceve il valore dell'item in formato stringa, stampato a video. Se l'item non viene trovato si riceve un messaggio di errore.

### 6.5.3 Comando remove

Questo comando permette di rimuovere un item dalla mappa, ricercandolo tramite la chiave.

```
actorbase> remove '<key>'
```

L'eliminazione viene confermata in caso di successo con il messaggio **"Item removed"**. Nel caso non si disponga dei permessi di modifica al database, o l'item richiesto non esista, si riceve un messaggio di operazione non valida (**"Invalid operation"**).

### 6.5.4 Comando insert

Questo comando permette di inserire un item (coppia chiave-valore) nella mappa:

```
actorbase> insert '<key>' <value>
```

L'inserimento viene confermato in caso di successo con il messaggio **"Item inserted"**. Nel caso non si disponga dei permessi di modifica al database, o sia già presente un item con la chiave inserita, si riceve un messaggio di operazione non valida (**"Invalid operation"**).

### 6.5.5 Comando update

Questo comando permette di aggiornare un item dalla mappa, ricercandolo tramite la chiave e inserendo il valore aggiornato.

```
actorbase> update '<key>' <value>
```

L'aggiornamento viene confermato in caso di successo con il messaggio **"Item updated"**. Nel caso non si disponga dei permessi di modifica al database, o l'item richiesto non esista, si riceve un messaggio di operazione non valida (**"Invalid operation"**).

## 6.6 Operazioni comandi amministratore

Un utente autenticato come amministratore può eseguire le seguenti operazioni:

- Lista utenti
- Aggiunta utente
- Rimozione utente
- Lista permessi utente
- Aggiunta permesso utente

- Rimozione permesso utente
- Impostazione numero massimo righe per Storekeeper
- Impostazione numero Ninja per Storekeeper
- Impostazione numero Warehouseman per Storekeeper
- Impostazione numero massimo Storekeeper per Storefinder
- Impostazione numero massimo Storefinder per Storemanager

#### 6.6.1 listuser command

Questo comando mostra l'intera lista di utenti nel server:

```
actorbase> listmap
```

#### 6.6.2 adduser command

Questo comando aggiunge un utente al server server:

```
actorbase> adduser username password
```

Se l'aggiunta è confermata il messaggio "User x added" viene mostrato, altrimenti un messaggio di errore contenente "User already exists" viene mostrato se un utente con quell'username è già presente.

#### 6.6.3 removeuser command

Questo comando rimuove un utente dal server:

```
actorbase> removeuser username
```

Se la rimozione è confermata il messaggio "User x removed" viene mostrato, altrimenti un messaggio di errore contenente "User doesn't exists" viene mostrato se non esiste un utente con quell'username.

#### 6.6.4 listpermission command

Questo comando mostra tutti i permessi di un utente:

```
actorbase> listpermission
```

#### 6.6.5 addpermission command

Questo comando permette di aggiungere un permesso per un database ad un utente :

```
actorbase> addpermission username database permissionType
```

Se l'aggiunta è confermata "Permission x added to user" viene visualizzato, altrimenti viene mostrato un messaggio contenente "User doesn't exists" se un utente con quell'username non esiste od uno contenente "Database doesn't exists" se un database con quel nome non esiste.

#### 6.6.6 removepermission command

Questo comando permette di rimuovere un permesso per un database ad un utente:

```
actorbase> removepermission username database
```

Se la rimozione è confermata "Permission x added to user" viene visualizzato, altrimenti viene mostrato un messaggio contenente "A permission for that database doesn't exists" se un permesso per quel database per l'utente non esiste.

#### **6.6.7 setmaxrows command**

Questo comando permette di impostare il numero massimo di righe per Storekeeper:

```
actorbase> setmaxrows number
```

#### **6.6.8 setninja command**

Questo comando permette di impostare il numero di Ninja per Storekeeper:

```
actorbase> setninja number
```

#### **6.6.9 setwarehouseman command**

Questo comando permette di impostare il numero di Warehouseman per Storekeeper:

```
actorbase> setwarehouseman number
```

#### **6.6.10 setmaxstorekeeper command**

Questo comando permette di impostare il numero massimo di Storekeeper per Storefinder:

```
actorbase> setmaxstorekeeper number
```

#### **6.6.11 setmaxstorefinder command**

Questo comando permette di impostare il numero massimo di Storefinder per Storemanager:

```
actorbase> setmaxstorefinder number
```

## Elenco delle figure

1	Logo di Actorbase . . . . .	3
2	Interfaccia di log del server . . . . .	5
3	Informazioni di benvenuto applicativo client . . . . .	6

## Elenco delle tabelle

1	Diario delle modifiche . . . . .	2
---	----------------------------------	---