SWEeneyThreads

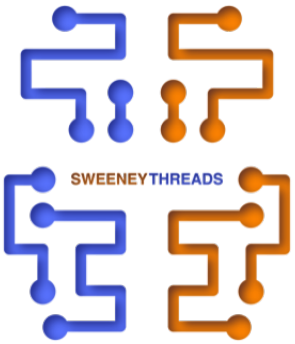
Actorbase

a NoSQL DB based on the Actor model

Specifica Tecnica

Redattori:
Bonato Paolo
Biggeri Mattia

Approvazione:
Verifica:



Versione 1.0.0

26 marzo 2016

Indice

Diario delle modifiche

Versione	Data	Autore	Descrizione
1.0.1	2016-03-26	*Progettisti* Bonato Paolo Biggeri Mattia Padovan Tommaso Tommasin Davide Bortolazzo Matteo	Prima stesura di Architettura generale (sezinoe 3) e componenti (sezione 4)
1.0.0	2016-03-24	*Analisti* Bonato Paolo Biggeri Mattia	Creazione scheletro documento, stesura introduzione, definizione di metodo e formalismo di specifica.

Tabella 1: Diario delle modifiche

1 Introduzione

1.1 Scopo del documento

Il documento definisce la progettazione ad alto livello del progetto Actorbase. Verrà presentata l'architettura generale, le componenti, le classi e i design pattern utilizzati per realizzare il prodotto.

1.2 Scopo del prodotto

Il progetto consiste nella 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.

1.3 Glossario

Al fine di evitare ambiguità di linguaggio e di massimizzare la comprensione dei documenti, il gruppo ha steso un documento interno che è il *Glossario v1.0.3*. In esso saranno definiti, in modo chiaro e conciso i termini che possono causare ambiguità o incomprensione del testo.

1.4 Riferimenti

- **Slide dell'insegnamento Ingegneria del software mod.A:**
  `http://www.math.unipd.it/~tullio/IS-1/2015/Dispense/E02.pdf`
- **Scala:**
  `http://www.scala-lang.org/`
- **Java:**
  `http://www.java.com/`
- **Akka:**
  `http://akka.io/`

1.4.1 Normativi

- **Norme di progetto:** *Norme di progetto v1.1.2*
- **Capitolato d'appalto Actorbase (C1):**
  `http://www.math.unipd.it/~tullio/IS-1/2015/Progetto/C1p.pdf`

2 Tecnologie utilizzate

2.1 Scala

Le possibili scelte dettate dal capitolato sono Java e Scala. Si è scelto di utilizzare Scala perché offre i seguenti vantaggi:

- **Concorrenza e distribuzione:** Ottimo supporto alla programmazione multi-threaded e distribuita, essenziale per la realizzazione di un prodotto responsive e scalabile.
- **Supporto di Akka:** Il linguaggio supporta la libreria Akka che è richiesta dal capitolato.
- **Preferenza del Committente:** Il Committente ha espresso la sua preferenza sull'utilizzo di Scala.

2.2 Akka

L'utilizzo della libreria Akka è reso obbligatorio dal capitolato ed è la base del modello ad attori che costituisce il progetto.

3 Descrizione dell'architettura

3.1 Metodo e formalismo di specifica

Nell'esposizione dell'architettura del prodotto si procederà con un approccio di tipo top-down, ovvero dal generale al particolare. Inizialmente si descriverà la distinzione tra le componenti Client e Server con le rispettive dipendenze. Questa suddivisione verrà mantenuta nelle sezioni a seguire. Per ogni package saranno descritti brevemente il tipo, l'obiettivo e la funzione e saranno specificati eventuali figli, classi ed interazioni con altri package. Ogni classe sarà dotata di una breve descrizione e ne saranno specificate le responsabilità, le classi ereditate, le sottoclassi e le relazioni con altre classi. Successivamente saranno mostrati e descritti i diagrammi delle attività che coinvolgono l'utente. Infine si illustreranno degli esempi di utilizzo dei design pattern nell'architettura del sistema.

3.2 Architettura generale

Il sistema ha un architettura generale di tipo client-server. Il server ha un architettura di tipo event-driven basata sul modello ad attori ed espone delle API tramite socket TCP. Il client ha un architettura di tipo Model-View-Controller con interfaccia da linea di comando e comunica con il server grazie ad un driver tramite connessione TCP.

4 Componenti

4.1 Server

Le componenti principali del server sono gli attori:

- **Storekeeper:** Mantengono fisicamente in memoria le mappe chiave/valore
- **Ninja:** Sono associati ad uno Storekeeper. Risiedono in nodi differenti da questi ultimi, ma in continuo aggiornamento con essi. Permettono di mantenere la consistenza del database anche in caso di caduta di uno Storekeeper; in questo caso un Ninja verrà eletto nuovo Storekeeper.
- **Storefinder:** Si occupano di ricevere richieste dall'esterno e instradarle ai rispettivi Storefinder, virtualmente ogni Storefinder definisce un indice sulla chiave della mappa. Il loro numero è veriabile e può essere configurato. Uno degli Storefinder è definito **main** e funge da punto di accesso al database.
- **Warehousemen:** Si interfaciano con gli Storekeeper e trascrivono persistentemente le rispettive mappe su disco.
- **Manager:** Ricevono le richieste di gestione degli attori di tipo Storekeeper, ad esempio sono responsabili del numero massimo di coppie chiave/valore contenute in un'istanza di Storekeeper.

4.2 Client

Il client ha struttura MVC.

- **Model:** Il Model è la componente che si occupa di comunicare con il server usando i metodi del driver. E notifica la view quando avviene un cambiamento nel suo stato.
- **View:** La View è la componente che interagisce con l'utente mediante interfaccia a linea di comando. L'utente può usare il DSL per interrogare il Model. La View esegue delle *state query* sul model per avere le informazioni aggiornate.
- **Controller:** Il Controller è la componenente che esegue il parsing dei comandi del DSL inseriti nella View e li notifica al Model.

4.3 Driver

Il Driver è una libreria, invocando i metodi della quale è possibile effettuare richieste TCP verso le API esposte dal Server.

5 Package

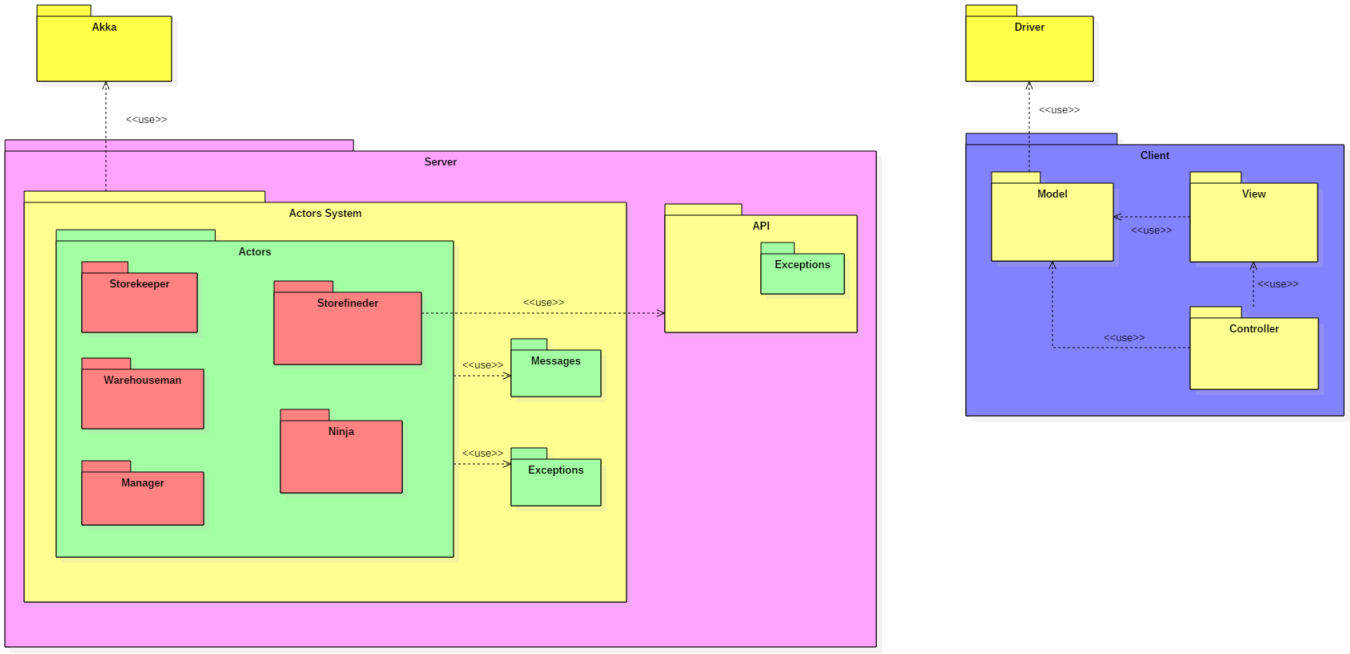


Figura 1: Architettura generale, vista Package

5.1 Lato server

5.1.1 package server 1

5.2 Lato client

5.2.1 package client 1

6 Classi

6.1 Lato server

6.2 classe server 1

6.3 Lato client

6.4 classe client 1

7 Diagrammi delle attività

8 Design pattern

9 Stime di fattibilità e di bisogno di risorse

10 Tracciamento

10.1 Tracciamento componenti-requisiti

10.2 Tracciamento requisiti-componenti

11 Appendice

Elenco delle figure

Elenco delle tabelle