SWEeneyThreads

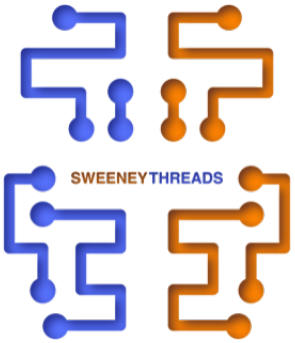
Actorbase

a NoSQL DB based on the Actor model

Specifica Tecnica

Redattori:
Bortolazzo Matteo
Nicoletti Luca

Approvazione:
Bonato Paolo
Verifica:
Padovan Tommaso
Tommasin Davide



Versione 1.0.5

->day

Indice

Diario delle modifiche

Versione	Data	Autore	Descrizione
1.0.5	2016-05-10	*Programmatore* Nicoletti Luca *Programmatore* Bortolazzo Matteo	Modificata la sezione Actorbase.driver.
1.0.4	2016-05-08	*Programmatore* Nicoletti Luca *Programmatore* Bortolazzo Matteo	Modificata la sezione Actorbase.client.
1.0.3	2016-05-06	*Programmatore* Nicoletti Luca *Programmatore* Bortolazzo Matteo	Modificata la sezione Actorbase.server.messages.
1.0.2	2016-05-04	*Programmatore* Nicoletti Luca *Programmatore* Bortolazzo Matteo	Modificata la sezione Actorbase.server.actors.
1.0.1	2016-05-01	*Programmatore* Nicoletti Luca *Programmatore* Bortolazzo Matteo	Iniziata la revisione, modificata la prima parte del Server.
1.0.0	2016-04-11	*Responsabile* Maino Elia	Documento approvato
0.3.0	2016-04-11	*Verificatore* Padovan Tommaso	Verificato il documento
0.2.1	2016-04-11	*Progettista* Nicoletti Luca	Riviste sezione da 4.8 a 4.17 e modificata sintassi delle classi correlate ad ogni classe. Sostituite vecchie immagini con immagini aggiornate (inserite aggregazioni tra le classi)
0.2.0	2016-04-11	*Verificatore* Bortolazzo Matteo	Verificato le tabelle inserite in v0.1.2 e il diario delle modifiche
0.1.3	2016-04-10	*Progettista* Nicoletti Luca	Separata tabella del diario della modifiche in file esterno.
0.1.2	2016-04-10	*Progettista* Biggeri Mattia	inserita tabella requisito-componente
0.1.1	2016-04-10	*Progettista* Nicoletti Luca	Sistemazione errori rilevati dalla verifica effettuata in versione 0.0.12
0.1.0	2016-04-10	*Verificatore* Bortolazzo Matteo	Verificato l'intero documento, segnalati vari errori in tutte le sezioni
0.0.11	2016-04-10	*Progettista* Maino Elia	Completata la descrizione di classe e interfacce del componente server
0.0.10	2016-04-10	*Progettista* Nicoletti Luca	Modificate e inserite le nuove immagini dei packages e delle classi per la sezione Server (4.2). Modificate le immagini nei diagrammi di sequenza (Sezione 6) e modificate le descrizioni di tutti in modo da uniformarle. Cambiate le immagini dei Design Pattern (Sezione 9.1)
0.0.9	2016-04-10	*Progettista* Padovan Tommaso	Incremento sezione Componenti e Classi: allineata sezione Driver allo standard, stesura sezione Client.
0.0.8	2016-04-09	*Progettisti* Bonato Paolo Biggeri Mattia	Stesura sezione tracciamento componenti-requisiti.

Versione	Data	Autore	Descrizione
0.0.7	2016-04-09	*Progettista* Nicoletti Luca	Inseriti tutti i diagrammi di sequenza riguardanti l'implementazione di richieste lato server, sostituita immagine dei Packages generale. Inserita una breve spiegazione di ogni diagramma di sequenza
0.0.6	2016-04-09	*Progettisti* Biggeri Mattia Padovan Tommaso Bonato Paolo	Stesura sezioni Driver, Diagrammi attività, Stime di fattibilità e bisogno di risorse; aggiunta immagine nella sezione 3.2.3, aggiunto Singleton nelle descrizioni dei Design Pattern, aggiornata leggenda.
0.0.5	2016-04-08	*Progettisti* Maino Elia Nicoletti Luca Bortolazzo Matteo	Stesura sezione riguardante le componenti dell'architettura lato server: diagrammi dei package e delle classi e descrizioni testuali
0.0.4	2016-04-06	*Progettisti* Biggeri Mattia Tommasin Davide	Aggiunta sezione in appendice suoi Design Pattern, contiene al momento la descrizione di: MVC, Event-driven, Command
0.0.3	2016-04-03	*Progettista* Bonato Paolo	Accorpate le sezioni "Componenti", "Package" e "Classi" in "Componenti e classi". Riadattata la sezione "Metodo e formalismo di specifica" alla nuova struttura. Inserite le immagini 1 e 2. Apportate le correzioni indicate.
0.0.2	2016-03-26	*Progettisti* Bonato Paolo Biggeri Mattia Padovan Tommaso Tommasin Davide Bortolazzo Matteo	Prima stesura di Architettura generale (sezione 3) e componenti (sezione 4)
0.0.1	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 illustra la progettazione attuale di Actorbase. Verranno presentati l'architettura, le componenti, le classi e i design pattern utilizzati.

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 v2.0.0*. 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/`
- **IntelliJ:**
  `http://www.jetbrains.com/idea/`

1.4.1 Normativi

- **Norme di progetto:** *Norme di progetto v2.0.0*
- **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:

- **Completamente Object-Oriented:** A differenza di Java, Scala è completamente orientato agli oggetti. Non c'è distinzione del tipo: oggetto - tipo primitivo, ogni valore è semplicemente un oggetto.
- **Staticamente tipato:** È un linguaggio tipato staticamente, questo permette di effettuare più facilmente i test. Inoltre Scala è in grado di stabilire il tipo di un oggetto per inferenza.
- **Può eseguire codice Java:** Scala può eseguire codice scritto in Java. È dunque possibile utilizzare classi e librerie scritte in Java all'interno di programmi scritti in Scala.
- **Concorrenza e distribuzione:** Ottimo supporto alla programmazione multi-threaded e distribuita, essenziale per la realizzazione di un prodotto responsive e scalabile.
- **Supporto alla definizione di DSL:** Scala supporta nativamente la definizione di DSL.
- **Supporto di Akka:** Il linguaggio supporta la libreria Akka che è richiesta dal capitolato.

Inoltre il Committente ha espresso esplicitamente la sua preferenza sull'utilizzo di Scala.

Figura 1: Scala - logo

2.2 Akka

L'utilizzo della libreria Akka oltre ad essere reso obbligatorio dal committente, fornisce un'eccellente base su cui sviluppare un sistema basato sul modello ad attori. Akka permette di costruire facilmente applicazioni message-driven che siano estremamente concorrenti, distribuite e resilienti. La natura distribuita e asincrona degli attori messi a disposizione da Akka soddisfa pienamente i bisogni del sistema da implementare.

Figura 2: Akka - logo

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.
Inizialmente si descriveranno le tre componenti fondamentali: Client, Server e Driver; poi le componenti più piccole al loro interno, specificando i package e le classi che li compongono.

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

L'architettura generale del sistema è di tipo client-server.

Il Server ha un'architettura di tipo event-driven basata sul modello ad attori e accetta connessioni in entrata tramite un socket TCP.

Il Client è un semplice programma che espone un'interfaccia a linea di comando per accettare comandi che vengono inoltrati al Server tramite il Driver.

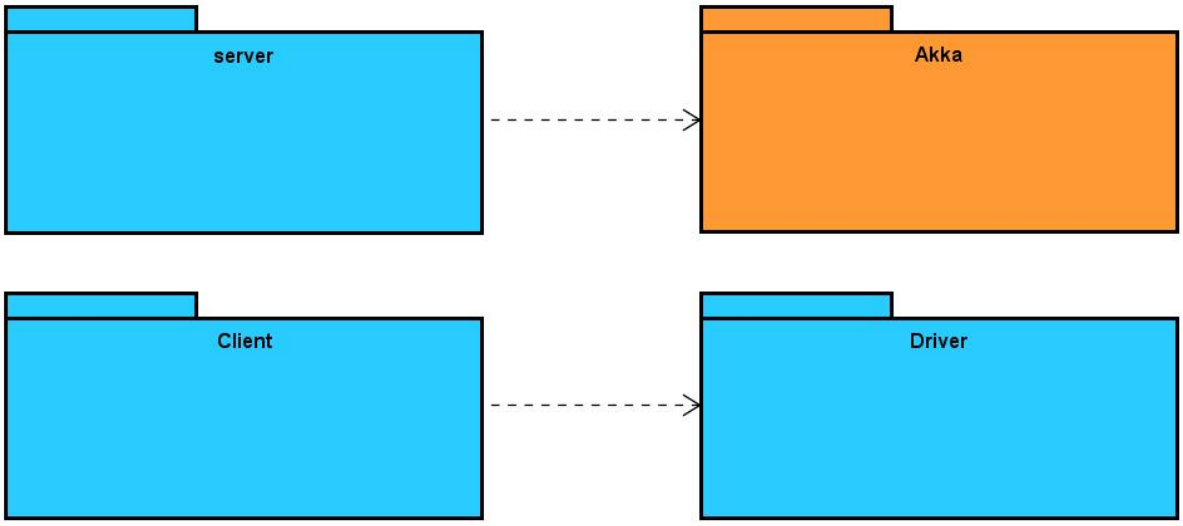


Figura 3: Architettura generale, vista Package

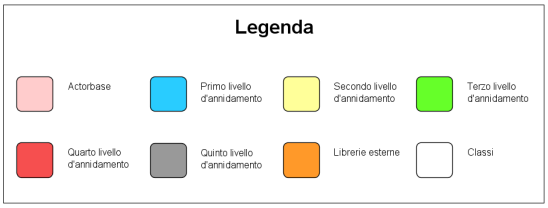


Figura 4: Legenda

3.2.1 Server

Il Server di *Actorbase* è composto da quattro package principali ed una classe: i package **utils**, **messages**, **actors** ed **enums** e la classe **Server**.

Il package **utils** contiene delle classi di supporto che vengono utilizzate dai vari attori presenti per effettuare operazioni di vario genere.
Il package **messages** contiene tutte le interfacce e le classi dei messaggi che gli attori possono mandarsi tra di loro.
Il package **actors** contiene tutte le classi che definisco gli attori del sistema.
Il package **enums** contiene le enumerazioni utilizzate per vari scopi.

La classe **Server** è l'entry point del programma e gestisce la configurazione iniziale all'avvio dello stesso.

3.2.2 Client

Il Client di *Actorbase* è composto da due sole classi: **Client** e **Welcome**, entrambe *Singleton*.

3.2.3 Driver

Il Driver è una libreria, invocando i metodi della quale è possibile effettuare richieste TCP sul socket su cui il Server è in ascolto.
Contiene due classi e un'interfaccia: l'interfaccia **Connection**, le classi **Driver** e **ConcreteConnection**. Quest'ultima è una realizzazione dell'interfaccia **Connection**.

4 Componenti e Classi

4.1 Actorbase

4.1.1 Descrizione

È il livello principale del sistema. L'interazione tra le componenti dei package **Server** e **Driver** avviene tramite una connessione di rete TCP di tipo client-server.

4.1.2 Package Figli

- Actorbase.server
- Actorbase.client
- Actorbase.driver

4.2 Actorbase.server

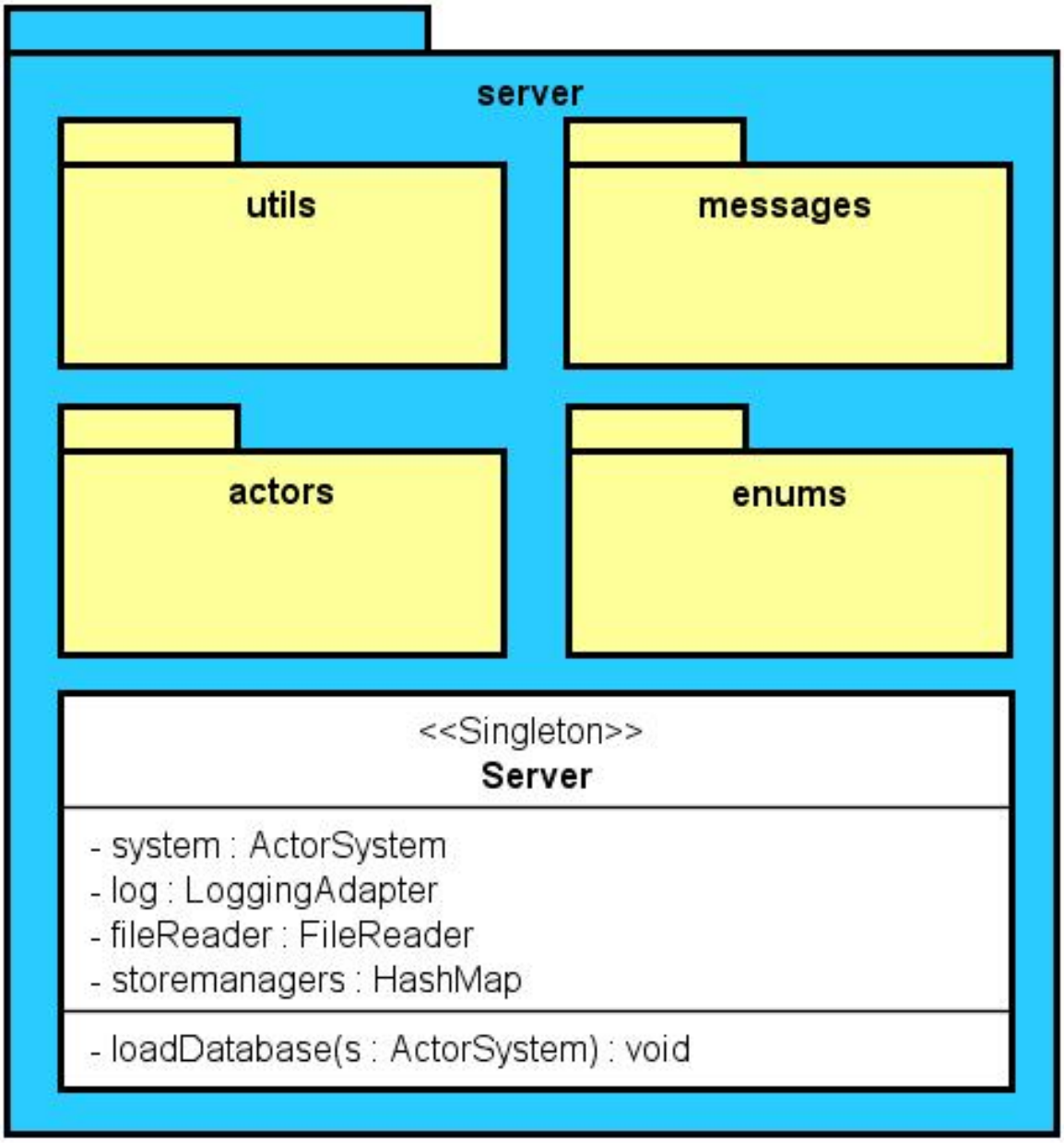


Figura 5: Componente Actorbase.server

4.2.1 Descrizione

Package per la componente lato server del sistema. È composto dai package **utils**, **messages**, **actors** ed **enums** e dalla classe **Server**.

4.2.2 Package Figli

- Actorbase.server.utils
- Actorbase.server.messages
- Actorbase.server.actors

- Actorbase.server.enums

4.2.3 Classi

- Actorbase.server.Server

4.3 Actorbase.server.Server

4.3.1 Descrizione

Classe principale della parte **Server** del programma. È di fatto l'entry point dello stesso, gestisce la configurazione iniziale e avvia il sistema. Utilizza il design pattern **Singleton**.

4.3.2 Utilizzo

Classe che fornisce un punto di accesso al programma, la sua esecuzione avvia il server sulla macchina in cui viene lanciata.

4.3.3 Relazione con altre classi

- **Actorbase.server.actors.Doorkeeper:** relazione uscente, creazione.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.utils.FileManager:** relazione uscente, creazione.

4.4 Actorbase.server.utils

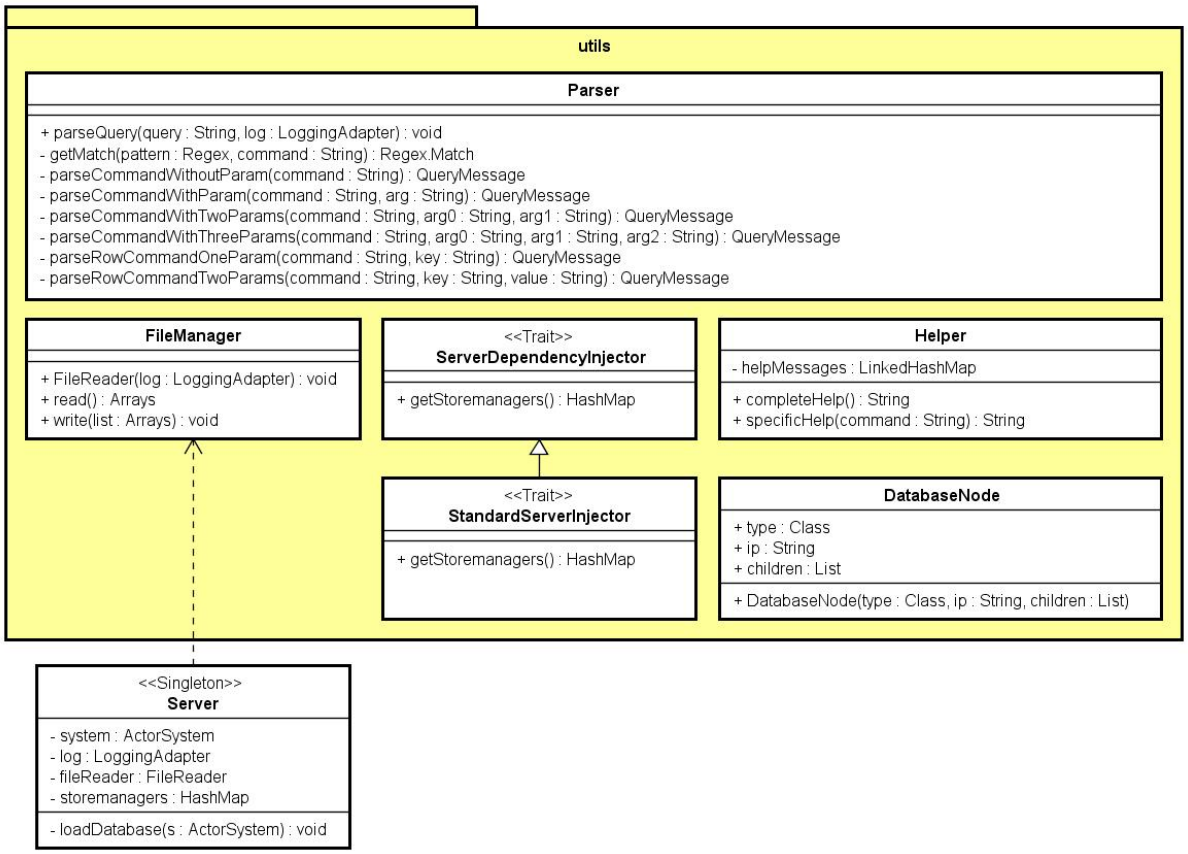


Figura 6: Componente Actorbase.server.utils

4.4.1 Descrizione

Package contenente le classi che effettuano operazioni varie a supporto delle varie componenti del server, e degli attori nello specifico.

4.4.2 Classi

- Actorbase.server.utils.Parser
- Actorbase.server.utils.FileManager
- Actorbase.server.utils.Helper
- Actorbase.server.utils.DatabaseNode

4.4.3 Trait

- Actorbase.server.utils.ServerDependencyInjector
- Actorbase.server.utils.StandardServerInjector

4.5 Actorbase.server.utils.ServerDependencyInjector

4.5.1 Descrizione

Trait che espone i metodi necessari per le interrogazioni al server da parte del **Main**.

4.5.2 Utilizzo

Viene utilizzato per delegare la chiamata ai metodi del Server ad un'interfaccia in modo da poter utilizzare server differenti in base alle necessità.

4.5.3 Relazione con altre classi

- **Actorbase.server.actors.Main:** relazione entrante, aggregazione.

4.5.4 Classi figlie

- Actorbase.server.utils.StandardServerInjector

4.6 Actorbase.server.utils.StandardServerInjector

4.6.1 Descrizione

Implementazione del trait **ServerDependencyInjector** che utilizza il server da noi creato.

4.6.2 Utilizzo

Viene utilizzato per delegare la chiamata ai metodi del Server ad un'interfaccia in modo da poter utilizzare server differenti in base alle necessità.

4.6.3 Relazione con altre classi

- **Actorbase.server.actors.Main:** relazione entrante, aggregazione.

4.7 Actorbase.server.utils.Parser

4.7.1 Descrizione

Classe utilizzata dal server per tradurre la stringa ricevuta da un client in un messaggio che ne rappresenta la richiesta per la comunicazione tra attori.

4.7.2 Utilizzo

Viene utilizzata dal Server quando esso riceve una richiesta da un Client.

4.7.3 Relazione con altre classi

- **Actorbase.server.actors.Usermanager:** relazione entrante, creazione.
- **Actorbase.server.messages.QueryMessages.QueryMessage:** relazione uscente, creazione.

4.8 Actorbase.server.utils.FileManager

4.8.1 Descrizione

Classe utilizzata dagli attori per gestire i file su disco.

4.8.2 Utilizzo

Viene utilizzata dalla maggior parte degli attori quando necessitano di leggere o scrivere su file.

4.8.3 Relazione con altre classi

- **Actorbase.server.Server:** relazione entrante, creazione.
- **Actorbase.server.actors.Warehouseman:** relazione entrante, creazione.
- **Actorbase.server.utils.DatabaseNode:** relazione uscente, creazione.

4.9 Actorbase.server.utils.Helper

4.9.1 Descrizione

Classe che mette a disposizione dei metodi per visualizzare la lista di comandi disponibili e la loro spiegazione.

4.9.2 Utilizzo

Viene utilizzata dall'attore Main per recuperare la lista di comandi con la loro spiegazione, è anche possibile recuperare la spiegazione di un singolo comando.

4.9.3 Relazione con altre classi

- **Actorbase.server.actors.Main:** relazione entrante, creazione.

4.10 Actorbase.server.utils.DatabaseNode

4.10.1 Descrizione

Classe utilizzata per costruire l'albero che rappresenta la struttura gerarchica delle componenti del database, è costituita da un singolo nodo e tutti i suoi discendenti.

4.10.2 Utilizzo

Viene utilizzata da tutti gli attori presenti nel database al momento della sua chiusura e apertura per salvare e ricostruire la struttura gerarchica.

4.10.3 Relazione con altre classi

- **Actorbase.server.Server:** relazione entrante, creazione.
- **Actorbase.server.actors.Main:** relazione entrante, creazione.
- **Actorbase.server.actors.Doorkeeper:** relazione entrante, creazione.
- **Actorbase.server.actors.Storefinder:** relazione entrante, creazione.
- **Actorbase.server.actors.Storekeeper:** relazione entrante, creazione.
- **Actorbase.server.actors.Warehouseman:** relazione entrante, creazione.

- **Actorbase.server.actors.Usermanager:** relazione entrante, creazione.
- **Actorbase.server.actors.Storemanager:** relazione entrante, creazione.
- **Actorbase.server.utils.FileManager:** relazione entrante, creazione.

4.11 Actorbase.server.actors

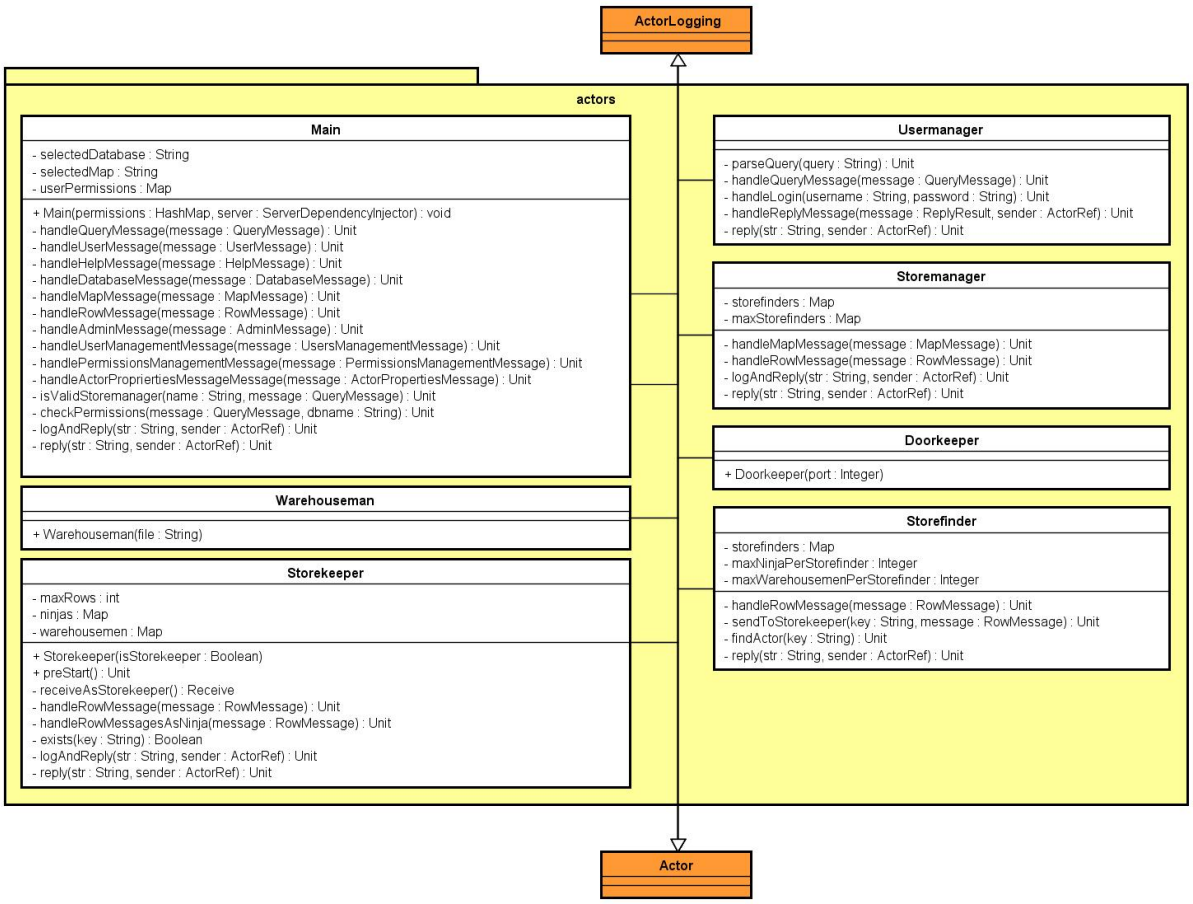


Figura 7: Componente Actorbase.server.actors

4.11.1 Descrizione

Package contenente tutti gli attori che compongono il database.

4.11.2 Classi

- Actorbase.server.actors.Main
- Actorbase.server.actors.Doorkeeper
- Actorbase.server.actors.Storekeeper
- Actorbase.server.actors.Storefinder
- Actorbase.server.actors.Warehouseman
- Actorbase.server.actors.Usermanager
- Actorbase.server.actors.Storemanager

4.12 Actorbase.server.actors.Doorkeeper

4.12.1 Descrizione

Questo attore apre il socket su cui i client potranno connettersi, crea un attore **Usermanager** per ogni nuova connessione ed inoltra ogni messaggio ricevuto dal client e all'**Usermanager** ad esso associato.

4.12.2 Utilizzo

Viene utilizzato per aprire un socket e resta in attesa di richieste di connessione da parte di client.

4.12.3 Relazione con altre classi

- **Actorbase.server.Server:** relazione entrante, creazione.
- **Actorbase.server.actors.Usermanager:** relazione uscente, creazione.
- **Actorbase.server.utils.ReplyResult:** relazione di utilizzo.

4.13 Actorbase.server.actors.Usermanager

4.13.1 Descrizione

Questo attore si preoccupa di trasformare le stringhe ricevute dal client in messaggi interpretabili da altri attori.

4.13.2 Utilizzo

Viene utilizzato per gestire una connessione da parte di un client, crea un attore **Main** e invia le richieste ricevute all'attore creato.

4.13.3 Relazione con altre classi

- **Actorbase.server.actors.Doorkeeper:** relazione entrante, creazione.
- **Actorbase.server.actors.Main:** relazione uscente, creazione.
- **Actorbase.server.utils.Parser:** relazione uscente, creazione.
- **Actorbase.server.utils.ReplyResult:** relazione di utilizzo.
- **Actorbase.server.messages.query.QueryMessage:** relazione di utilizzo.

4.14 Actorbase.server.actors.Main

4.14.1 Descrizione

Questo attore è responsabile del controllo dei permessi delle richieste ricevute. Gestisce tutti i messaggi, elaborandoli o inoltrandoli se non di sua competenza.

4.14.2 Utilizzo

Viene utilizzato per controllare i permessi delle richieste ricevute, gestisce i messaggi a livello di database e di help e inoltra tutti gli altri messaggi che non sono di sua competenza. Inoltre si preoccupa di mantenere in memoria le selezioni di database e mappa effettuate dall'utente.

4.14.3 Relazione con altre classi

- **Actorbase.server.Server:** relazione di utilizzo.
- **Actorbase.server.actors.Usermanager:** relazione entrante, creazione.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione uscente, creazione.
- **Actorbase.server.utils.Helper:** relazione uscente, creazione.
- **Actorbase.server.Permission:** relazione di utilizzo.
- **Actorbase.server.utils.ReplyResult:** relazione di utilizzo.
- **Actorbase.server.messages.query.QueryMessage:** relazione di utilizzo.
- **Actorbase.server.messages.query.ReplyMessage:** relazione di utilizzo.
- **Actorbase.server.messages.query.PermissionMessages.NoPermissionMessage:** relazione di utilizzo.
- **Actorbase.server.messages.query.PermissionMessages.AdminPermissionMessage:** relazione di utilizzo.
- **Actorbase.server.messages.internal.AskMapMessage:** relazione uscente, creazione e utilizzo.
- **Actorbase.server.messages.internal.BecomeStorekeeperMessage:** relazione di utilizzo.
- **Actorbase.server.messages.internal.SendMapMessage:** relazione di utilizzo.
- **Actorbase.server.messages.internal.LinkMessages.LinkMessage:** relazione di utilizzo.

4.15 Actorbase.server.actors.Storemanager

4.15.1 Descrizione

Questo attore rappresenta un database o una sua parte.

4.15.2 Utilizzo

Viene utilizzato per gestire tutti i messaggi a livello mappa, e inoltra tutti gli altri messaggi che non sono di sua competenza.

4.15.3 Relazione con altre classi

- **Actorbase.server.Server:** relazione entrante, creazione.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione uscente, creazione.
- **Actorbase.server.utils.ReplyResult:** relazione di utilizzo.
- **Actorbase.server.messages.query.user.UserMessage:** relazione di utilizzo.

4.16 Actorbase.server.actors.Storefinder

4.16.1 Descrizione

Questo attore rappresenta una mappa o una sua parte.

4.16.2 Utilizzo

Viene utilizzato per gestire gli **Storekeeper** a cui inoltra i messaggi.

4.16.3 Relazione con altre classi

- **Actorbase.server.actors.Storemanager:** relazione entrante, creazione.
- **Actorbase.server.actors.Storekeeper:** relazione uscente, creazione.
- **Actorbase.server.actors.Warehouseman:** relazione uscente, creazione.
- **Actorbase.server.utils.ReplyResult:** relazione di utilizzo.
- **Actorbase.server.messages.query.users.RowMessages.RowMessage:** relazione di utilizzo.
- **Actorbase.server.message.internal.BecomeStorekeeperMessage:** relazione uscente, creazione e utilizzo.
- **Actorbase.server.messages.internal.LinkMessages.LinkMessage:** relazione uscente, creazione e utilizzo.

4.17 Actorbase.server.actors.Storekeeper

4.17.1 Descrizione

Questo attore contiene una parte di mappa. Esso ha anche funzionalità di backup, qualora uno **Storekeeper** morisse, un altro potrebbe subentrare al suo posto.

4.17.2 Utilizzo

Viene utilizzato per memorizzare in memoria RAM le entry presenti in una mappa, gestisce i messaggi a livello di riga. Quando un attore della stessa classe a cui esso è associato si spegne, esso diventa la copia ufficiale della parte di mappa da esso rappresentata.

4.17.3 Relazione con altre classi

- **Actorbase.server.actors.Storefinder:** relazione entrante, creazione.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.
- **Actorbase.server.utils.ReplyResult:** relazione di utilizzo.
- **Actorbase.server.messages.query.users.RowMessages.RowMessage:** relazione di utilizzo.
- **Actorbase.server.message.internal.BecomeStorekeeperMessage:** relazione di utilizzo.
- **Actorbase.server.messages.internal.SendMapMessage:** relazione di utilizzo.
- **Actorbase.server.messages.internal.LinkMessages.LinkMessage:** relazione di utilizzo.

4.18 Actorbase.server.actors.Warehouseman

4.18.1 Descrizione

Questo attore permette l'interazione del database con il filesystem.

4.18.2 Utilizzo

Viene utilizzato per mantenere un backup del database su file presenti su disco.

4.18.3 Relazione con altre classi

- **Actorbase.server.actors.Storekeeper:** relazione di utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione entrante, creazione.
- **Actorbase.server.utils.ReplyResult:** relazione di utilizzo.
- **Actorbase.server.messages.query.users.RowMessages.RowMessage:** relazione di utilizzo.
- **Actorbase.server.messages.internal.SendMapMessage:** relazione uscente, creazione e utilizzo.

4.19 Actorbase.server.enums

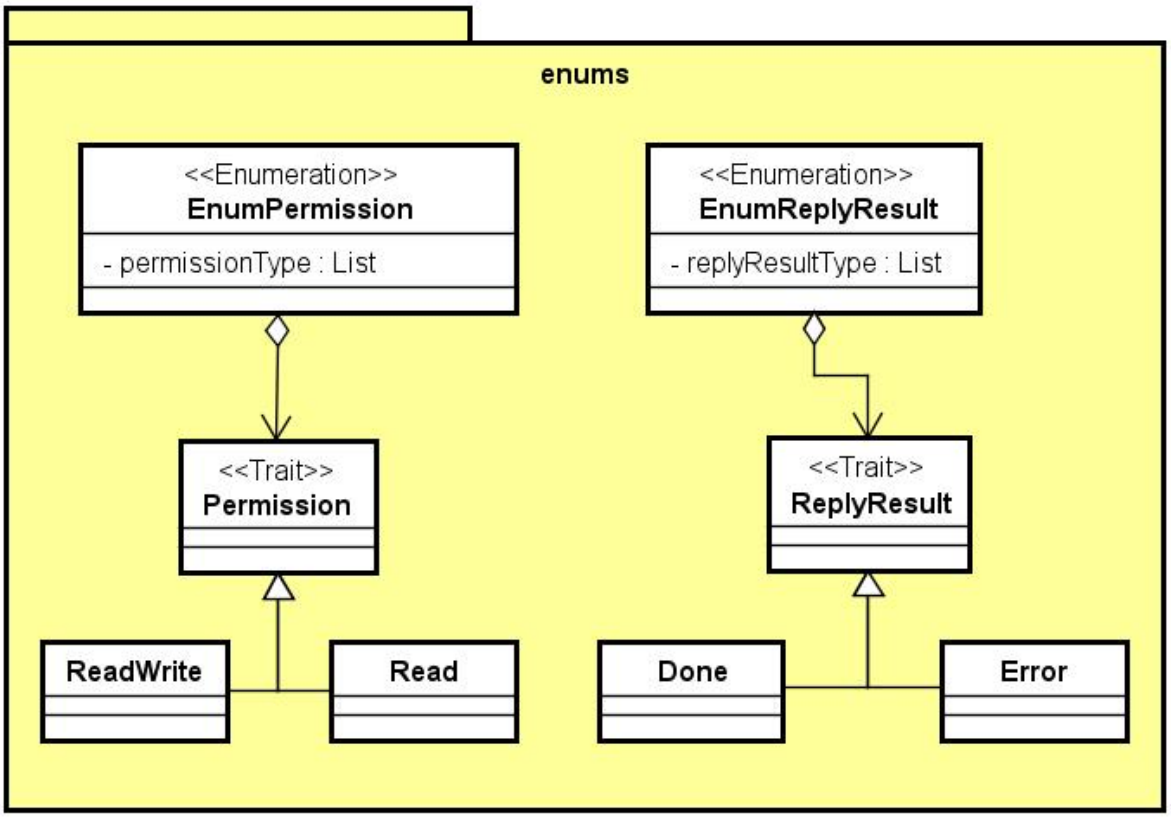


Figura 8: Componente Actorbase.server.enums

4.19.1 Descrizione

Package contenente le enumerazioni utilizzate dagli attori per distinguere i tipi di permessi e di risposte.

4.19.2 Classi

- Actorbase.server.enums.Read
- Actorbase.server.enums.ReadWrite
- Actorbase.server.enums.Done
- Actorbase.server.enums.Error
- Actorbase.server.enums.EnumPermission
- Actorbase.server.enums.EnumReplyResult

4.19.3 Trait

- Actorbase.server.utils.Permission
- Actorbase.server.utils.ReplyResult

4.20 Actorbase.server.enums.Permission

4.20.1 Descrizione

Trait per rappresentare i permessi che la richiesta di un utente può avere.

4.20.2 Utilizzo

Viene utilizzata dall'attore **Main** per effettuare i controlli dei permessi dei messaggi.

4.20.3 Relazione con altre classi

- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.enums.EnumPermission:** relazione entrante, aggregazione.

4.20.4 Classi figlie

- Actorbase.server.enums.Read.
- Actorbase.server.enums.ReadWrite.

4.21 Actorbase.server.enums.ReplyResult

4.21.1 Descrizione

Trait per rappresentare le risposte ed il loro stato (avvenuta con successo, errore).

4.21.2 Utilizzo

Viene utilizzata da tutti gli attori per indicare se la richiesta effettuata da un utente è andata a buon fine o no.

4.21.3 Relazione con altre classi

- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Doorkeeper:** relazione di utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione di utilizzo.
- **Actorbase.server.actors.Storekeeper:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.enums.EnumReplyResult:** relazione entrante, aggregazione.

4.21.4 Classi figlie

- Actorbase.server.enums.Error.
- Actorbase.server.enums.Done.

4.22 Actorbase.server.enums.Read

4.22.1 Descrizione

Questa classe rappresenta il permesso di lettura.

4.22.2 Utilizzo

Viene utilizzata dall'attore **Main** per effettuare i controlli dei permessi dei messaggi.

4.22.3 Relazione con altre classi

- **Actorbase.server.actors.Main:** relazione di utilizzo.

4.22.4 Trait implementati

- **Actorbase.server.enums.Permission**

4.23 Actorbase.server.enums.Write

4.23.1 Descrizione

Questa classe rappresenta il permesso di scrittura.

4.23.2 Utilizzo

Viene utilizzata dall'attore **Main** per effettuare i controlli dei permessi dei messaggi.

4.23.3 Relazione con altre classi

- **Actorbase.server.actors.Main:** relazione di utilizzo.

4.23.4 Trait implementati

- **Actorbase.server.enums.Permission**

4.24 Actorbase.server.enums.Done

4.24.1 Descrizione

Questa classe rappresenta l'avvenuto successo di una richiesta effettuata da un utente.

4.24.2 Utilizzo

Viene utilizzata da tutti gli attori come risposta in caso di richiesta effettuata con successo.

4.24.3 Relazione con altre classi

- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Doorkeeper:** relazione di utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione di utilizzo.
- **Actorbase.server.actors.Storekeeper:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.

4.24.4 Trait implementati

- **Actorbase.server.enums.ReplyResult**

4.25 Actorbase.server.enums.Error

4.25.1 Descrizione

Questa classe rappresenta il verificarsi di un errore durante l'esecuzione di una richiesta effettuata da un utente.

4.25.2 Utilizzo

Viene utilizzata da tutti gli attori come risposta in caso di errore durante l'esecuzione di una richiesta effettuata da un utente.

4.25.3 Relazione con altre classi

- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Doorkeeper:** relazione di utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione di utilizzo.
- **Actorbase.server.actors.Storekeeper:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.

4.25.4 Trait implementati

- **Actorbase.server.enums.ReplyResult**

4.26 Actorbase.server.enums.EnumPermission

4.26.1 Descrizione

Questa classe contiene la lista dei permessi che un utente può avere.

4.26.2 Utilizzo

Viene utilizzata dall'attore **Main** per effettuare i controlli dei permessi dei messaggi.

4.26.3 Relazione con altre classi

- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.enums.Permission:** relazione uscente, aggregazione.

4.27 Actorbase.server.enums.EnumReplyResult

4.27.1 Descrizione

Questa classe contiene la lista di tipi di risposta che una richiesta effettuata da un utente può avere.

4.27.2 Utilizzo

Viene utilizzata da tutti gli attori per accedere ai tipi di risposta che una richiesta effettuata da un utente può avere.

4.27.3 Relazione con altre classi

- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Doorkeeper:** relazione di utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione di utilizzo.
- **Actorbase.server.actors.Storekeeper:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.enums.ReplyResult:** relazione uscente, aggregazione.

4.28 Actorbase.server.messages

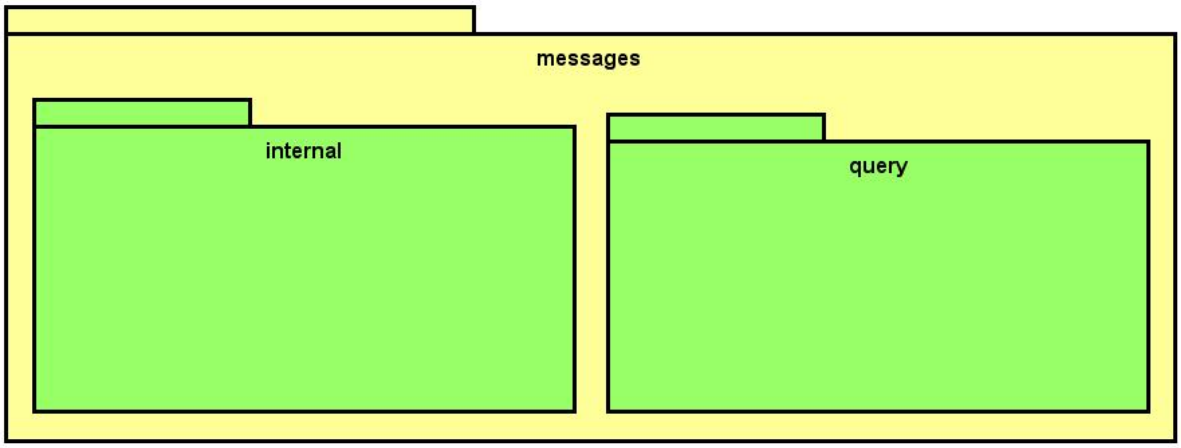


Figura 9: Componente Actorbase.server.messages

4.28.1 Descrizione

Package contenente tutti i messaggi che gli attori si scambiano tra di loro. È composto dai package **internal** e **query**.

4.28.2 Package Figli

- Actorbase.server.messages.internal.
- Actorbase.server.messages.query.

4.29 Actorbase.server.messages.internal

Figura 10: Componente Actorbase.server.messages.internal

4.29.1 Descrizione

Package contenente i messaggi che gli attori si scambiano non riguardanti richieste effettuate dagli utenti.

4.29.2 Package Figli

- Actorbase.server.messages.internal.LinkMessages

4.29.3 Classi

- Actorbase.server.messages.internal.AskMapMessage
- Actorbase.server.messages.internal.BecomeStorekeeperMessage
- Actorbase.server.messages.internal.SendMapMessage

4.30 Actorbase.server.messages.internal.AskMapMessage

4.30.1 Descrizione

Questa classe rappresenta un messaggio per verificare la presenza di una mappa in un **Storemanager**.

4.30.2 Utilizzo

Viene inviato da un **Main** ad uno **Storemanager**.

4.30.3 Relazione con altre classi

- **Actorbase.server.actors.Main:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.

4.31 Actorbase.server.messages.internal.BecomeAStorekeeperMsg

4.31.1 Descrizione

Questa classe rappresenta un messaggio per far cambiare l'implementazione di ricezione di messaggi di uno **Storekeeper** che precedentemente si comportava come un **Ninja**.

4.31.2 Utilizzo

Viene inviato da uno **Storefinder** ad uno **Storekeeper**.

4.31.3 Relazione con altre classi

- **Actorbase.server.actors.Storefinder:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.actors.Storekeeper:** relazione di utilizzo.

4.32 Actorbase.server.messages.internal.SendMapMessage

4.32.1 Descrizione

Questa classe rappresenta un messaggio contenente una mappa.

4.32.2 Utilizzo

Viene inviato da un **Warehouseman** ad uno **Storekeeper**.

4.32.3 Relazione con altre classi

- **Actorbase.server.actors.Warehouseman:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.actors.Storekeeper:** relazione di utilizzo.

4.33 Actorbase.server.messages.internal.LinkMessages

4.33.1 Descrizione

Package contenente i messaggi che gli attori si scambiano non riguardanti richieste effettuate dagli utenti.

4.33.2 Classi

- Actorbase.server.messages.internal.LinkMessages.AddNinjaMessage
- Actorbase.server.messages.internal.LinkMessages.AddWarehousemanMessage
- Actorbase.server.messages.internal.LinkMessages.RemoveNinjaMessage
- Actorbase.server.messages.internal.LinkMessages.RemoveWarehousemanMessage

4.33.3 Trait

- Actorbase.server.messages.internal.LinkMessages.LinkMessage

4.34 Actorbase.server.messages.internal.LinkMessages.LinkMessage

4.34.1 Descrizione

Trait per la richiesta di aggiunta o rimozione di un attore da parte di un altro attore.

4.34.2 Utilizzo

Viene inviato da uno **Storefinder** ad uno **Storekeeper**.

4.34.3 Relazione con altre classi

- **Actorbase.server.actors.Storefinder:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.actors.Storekeeper:** relazione di utilizzo.

4.34.4 Classi figlie

- Actorbase.server.messages.internal.LinkMessages.AddNinjaMessage
- Actorbase.server.messages.internal.LinkMessages.AddWarehousemanMessage
- Actorbase.server.messages.internal.LinkMessages.RemoveNinjaMessage
- Actorbase.server.messages.internal.LinkMessages.RemoveWarehousemanMessage

4.35 Actorbase.server.messages.internal.LinkMessages.AddNinjaMessage

4.35.1 Descrizione

Questa classe rappresenta la richiesta di aggiunta di un **Ninja**, ovvero uno **Storekeeper** di backup, da parte di uno **Storefinder** ad uno **Storekeeper**.

4.35.2 Utilizzo

Viene inviato da uno **Storefinder** ad uno **Storekeeper**.

4.35.3 Relazione con altre classi

- **Actorbase.server.actors.Storefinder:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.actors.Storekeeper:** relazione di utilizzo.

4.35.4 Trait implementati

- **Actorbase.server.messages.internal.LinkMessages.LinkMessage**

4.36 Actorbase.server.messages.internal.LinkMessages.AddWarehousemanMessage

4.36.1 Descrizione

Questa classe rappresenta la richiesta di aggiunta di un **Warehouseman** da parte di uno **Storefinder** ad uno **Storekeeper**.

4.36.2 Utilizzo

Viene inviato da uno **Storefinder** ad uno **Storekeeper**.

4.36.3 Relazione con altre classi

- **Actorbase.server.actors.Storefinder:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.actors.Storekeeper:** relazione di utilizzo.

4.36.4 Trait implementati

- **Actorbase.server.messages.internal.LinkMessages.LinkMessage**

4.37 Actorbase.server.messages.internal.LinkMessages.RemoveNinjaMessage

4.37.1 Descrizione

Questa classe rappresenta la richiesta di rimozione di un **Ninja**, ovvero uno **Storekeeper** di backup, da parte di uno **Storefinder** ad uno **Storekeeper**.

4.37.2 Utilizzo

Viene inviato da uno **Storefinder** ad uno **Storekeeper**.

4.37.3 Relazione con altre classi

- **Actorbase.server.actors.Storefinder:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.actors.Storekeeper:** relazione di utilizzo.

4.37.4 Trait implementati

- **Actorbase.server.messages.internal.LinkMessages.LinkMessage**

4.38 Actorbase.server.messages.internal.LinkMessages.RemoveWarehousemanMessage

4.38.1 Descrizione

Questa classe rappresenta la richiesta di rimozione di un **Warehouseman** da parte di uno **Storefinder** ad uno **Storekeeper**.

4.38.2 Utilizzo

Viene inviato da uno **Storefinder** ad uno **Storekeeper**.

4.38.3 Relazione con altre classi

- **Actorbase.server.actors.Storefinder:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.actors.Storekeeper:** relazione di utilizzo.

4.38.4 Trait implementati

- **Actorbase.server.messages.internal.LinkMessages.LinkMessage**

4.39 Actorbase.server.messages.query

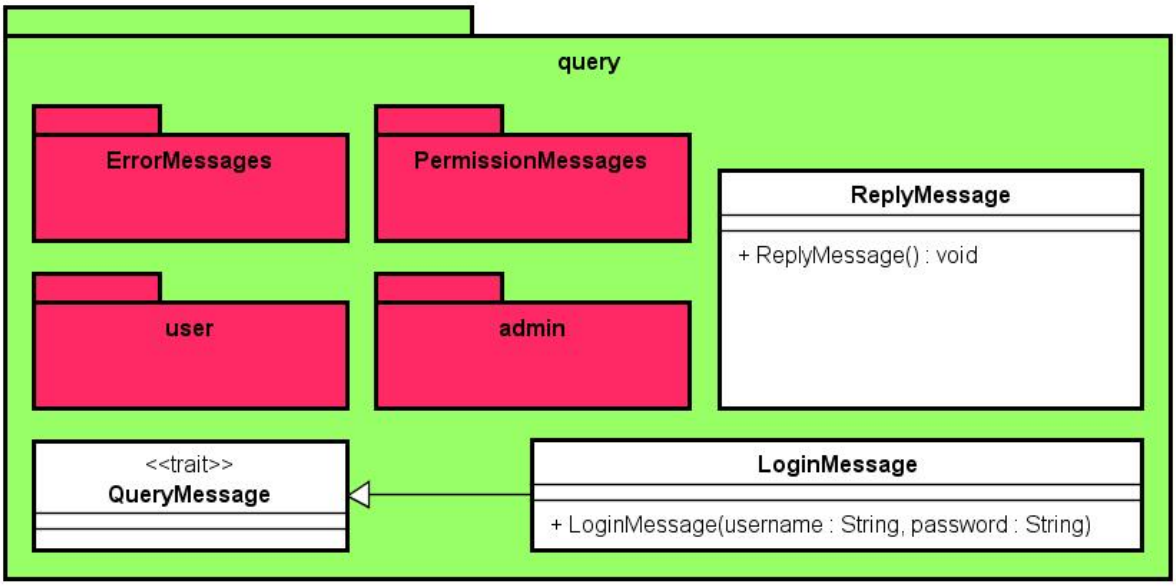


Figura 11: Componente Actorbase.server.messages.query

4.39.1 Descrizione

Package contenente i messaggi che rappresentano le possibili richieste che gli utenti possono effettuare. Questi messaggi vengono scambiati tra gli attori presenti nel database.

4.39.2 Package Figli

- Actorbase.server.messages.query.ErrorMessages
- Actorbase.server.messages.query.PermissionMessages
- Actorbase.server.messages.query.user
- Actorbase.server.messages.query.admin

4.39.3 Classi

- Actorbase.server.messages.query.LoginMessage
- Actorbase.server.messages.query.ReplyMessage

4.39.4 Trait

- Actorbase.server.messages.query.QueryMessage

4.40 Actorbase.server.messages.query.QueryMessage

4.40.1 Descrizione

Classe che rappresenta un messaggio contenente una richiesta effettuata da un utente.

4.40.2 Utilizzo

Viene utilizzata da tutti gli attori, in particolar modo segue diversi percorsi in base alla sottoclasse concreta a cui appartiene. Viene creato da un **Parser** sotto richiesta di un **Usermanager** ed inviato ad un **Main**. Da qui può fermarsi in qualsiasi punto della sequenza: **Storemanager** -> **Storefinder** -> **Storekeeper** -> **Warehouseman**.

4.40.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione di utilizzo.
- **Actorbase.server.actors.Storekeeper:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.
- **Actorbase.server.messages.query.ReplyMessage:** relazione entrante, aggregazione.

4.40.4 Classi figlie

- Actorbase.server.messages.query.LoginMessage
- Actorbase.server.messages.query.ErrorMessages.InvalidQueryMessage
- Actorbase.server.messages.query.admin.AdminMessage
- Actorbase.server.messages.query.user.UserMessage

4.41 Actorbase.server.messages.query.LoginMessage

4.41.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di login effettuata da parte di un utente.

4.41.2 Utilizzo

Nel percorso esposto in Actorbase.server.messages.query.QueryMessage si ferma all'attore **Usermanager**.

4.41.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.

4.41.4 Trait implementati

- Actorbase.server.messages.query.QueryMessage

4.42 Actorbase.server.messages.query.ReplyMEssage

4.42.1 Descrizione

Classe che rappresenta un messaggio di risposta ad una richiesta effettuata da parte di un utente.

4.42.2 Utilizzo

Effettua il percorso esposto in Actorbase.server.messages.query.QueryMessage in senso opposto.

4.42.3 Relazioni con altre classi

- **Actorbase.server.actors.Usermanager:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.actors.Main:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.actors.Storekeeper:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione entrante, creazione e utilizzo.

4.43 Actorbase.server.messages.query.ErrorMessages

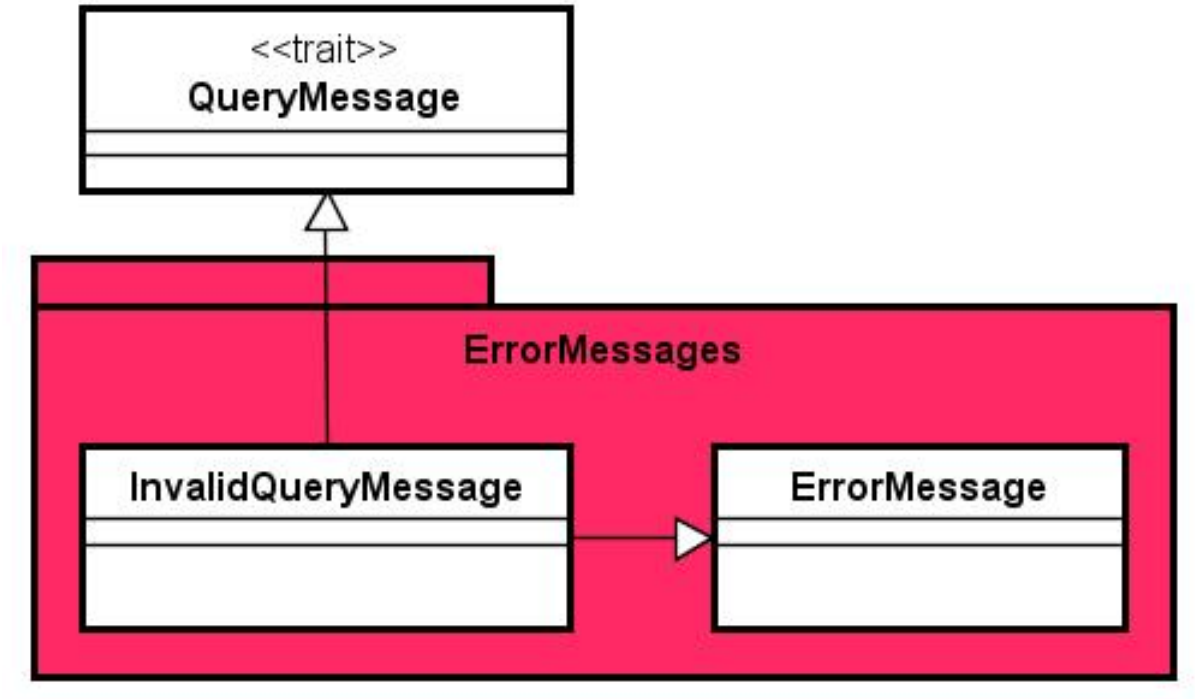


Figura 12: Componente Actorbase.server.messages.query.ErrorMessages

4.43.1 Descrizione

Package contenente i messaggi di errore che vengono utilizzati qualora si verifichi un errore durante la realizzazione di una richiesta effettuata da un utente.

4.43.2 Classi

- Actorbase.server.messages.query.ErrorMessages.InvalidQueryMessage

4.43.3 Trait

- Actorbase.server.messages.query.ErrorMessages.ErrorMessage

4.44 Actorbase.server.messages.query.ErrorMessages.ErrorMessage

4.44.1 Descrizione

Trait per segnalare che la richiesta effettuata non è riconosciuta dal server.

4.44.2 Utilizzo

Viene inviato da uno **Usermanager**.

4.44.3 Relazione con altre classi

- **Actorbase.server.actors.Usermanager:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.utils.Parser:** relazione entrante, creazione.

4.44.4 Classi figlie

- Actorbase.server.messages.query.ErrorMessages.InvalidQueryMessage

4.45 Actorbase.server.messages.query.ErrorMessages.InvalidQueryMessage

4.45.1 Descrizione

Questa classe rappresenta un messaggio per segnalare che la richiesta effettuata non è riconosciuta dal server.

4.45.2 Utilizzo

Viene inviato da uno **Usermanager**.

4.45.3 Relazione con altre classi

- **Actorbase.server.actors.Usermanager:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.utils.Parser:** relazione entrante, creazione.

4.45.4 Trait implementati

- **Actorbase.server.messages.query.QueryMessage**
- **Actorbase.server.messages.query.ErrorMessages.ErrorMessage**

4.46 Actorbase.server.messages.query.PermissionMessages

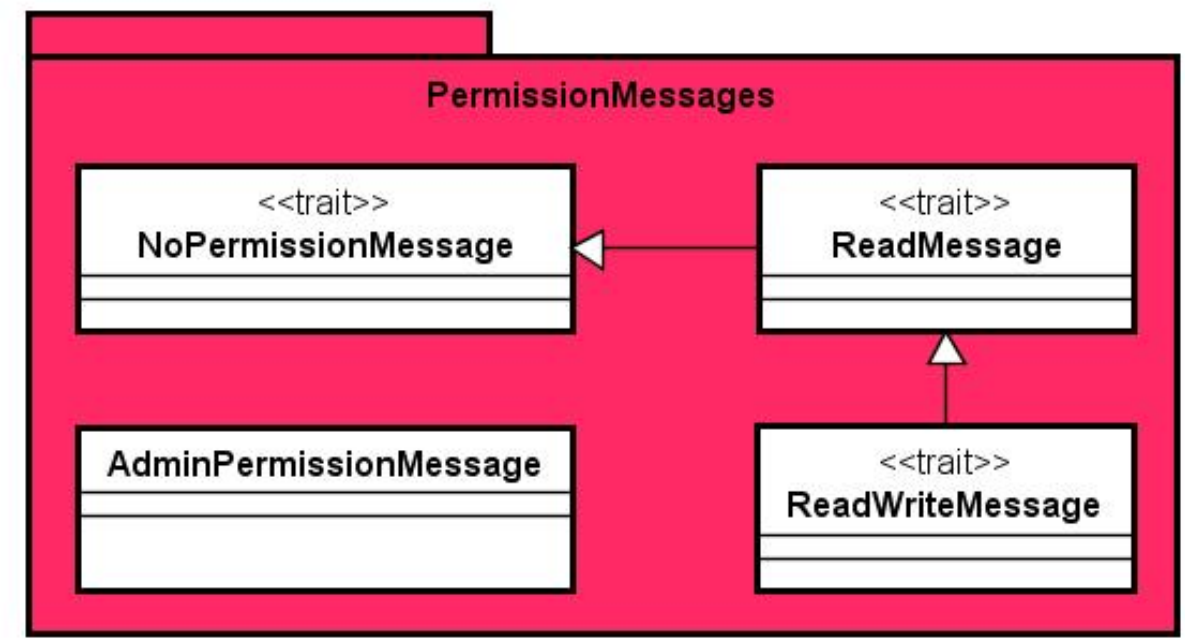


Figura 13: Componente Actorbase.server.messages.query.PermissionMessages

4.46.1 Descrizione

Package contenente i messaggi che rappresentano i vari livelli di permessi necessari agli utenti al fine di poter effettuare una richiesta.

4.46.2 Trait

- Actorbase.server.messages.query.PermissionMessages.AdminPermissionMessage
- Actorbase.server.messages.query.PermissionMessages.NoPermissionMessage

- Actorbase.server.messages.query.PermissionMessages.ReadMessage
- Actorbase.server.messages.query.PermissionMessages.ReadWrite

4.47 Actorbase.server.messages.query.PermissionMessages.AdminPermissionMessage

4.47.1 Descrizione

Trait che rappresenta un comando che richiede i permessi di amministrazione per la sua esecuzione.

4.47.2 Utilizzo

Ha lo stesso utilizzo di Actorbase.server.messages.query.QueryMessage.

4.47.3 Classi figlie

- Actorbase.server.messages.query.admin.AdminMessage

4.48 Actorbase.server.messages.query.PermissionMessages.NoPermissionMessage

4.48.1 Descrizione

Trait che rappresenta un comando che non richiede alcun permesso per la sua esecuzione.

4.48.2 Utilizzo

Ha lo stesso utilizzo di Actorbase.server.messages.query.QueryMessage.

4.48.3 Classi figlie

- Actorbase.server.messages.query.user.DatabaseMessages.CreateDatabaseMessage
- Actorbase.server.messages.query.user.HelpMessages.CompleteHelpMessage
- Actorbase.server.messages.query.user.HelpMessages.SpecificHelpMessage

4.49 Actorbase.server.messages.query.PermissionMessages.ReadMessage

4.49.1 Descrizione

Trait che rappresenta un comando che richiede almeno i permessi di lettura per la sua esecuzione.

4.49.2 Utilizzo

Ha lo stesso utilizzo di Actorbase.server.messages.query.QueryMessage.

4.49.3 Classi figlie

- Actorbase.server.messages.query.user.DatabaseMessages.SelectDatabaseMessage
- Actorbase.server.messages.query.user.DatabaseMessages.ListDatabaseMessage
- Actorbase.server.messages.query.user.MapMessages.SelectMapMessage
- Actorbase.server.messages.query.user.MapMessages.ListMapMessage
- Actorbase.server.messages.query.user.RowMessages.FindRowMessage
- Actorbase.server.messages.query.user.RowMessages.ListKeysMessage

4.50 Actorbase.server.messages.query.PermissionMessages.ReadWriteMessage

4.50.1 Descrizione

Trait che rappresenta un comando che richiede i permessi di scrittura per la sua esecuzione.

4.50.2 Utilizzo

Ha lo stesso utilizzo di Actorbase.server.messages.query.QueryMessage.

4.50.3 Classi figlie

- Actorbase.server.messages.query.user.DatabaseMessages.DeleteDatabaseMessage
- Actorbase.server.messages.query.user.MapMessages.DeleteMapMessage
- Actorbase.server.messages.query.user.MapMessages.CreateMapMessage
- Actorbase.server.messages.query.user.RowMessages.InsertRowMessage
- Actorbase.server.messages.query.user.RowMessages.UpdateRowMessage
- Actorbase.server.messages.query.user.RowMessages.RemoveRowMessage

4.51 Actorbase.server.messages.query.admin

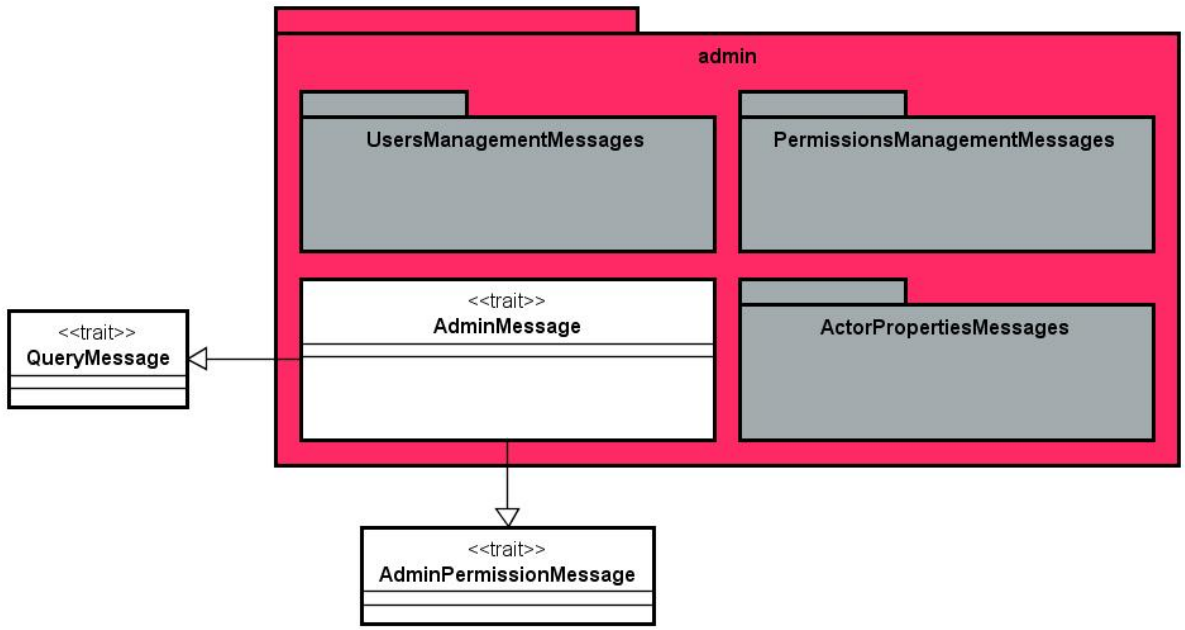


Figura 14: Componente Actorbase.server.messages.query.admin

4.51.1 Descrizione

Package contenente i messaggi che rappresentano i comandi amministrativi.

4.51.2 Package Figli

- Actorbase.server.messages.query.admin.ActorPropertiesMessages
- Actorbase.server.messages.query.admin.PermissionsManagementMessages
- Actorbase.server.messages.query.admin.UserManagementMessages

4.51.3 Trait

- Actorbase.server.messages.query.admin.AdminMessage

4.52 Actorbase.server.messages.query.admin.AdminMessage

4.52.1 Descrizione

Trait che rappresenta tutti i comandi per effettuare operazioni di tipo amministrativo.

4.52.2 Utilizzo

Viene utilizzato per poter distinguere i comandi di tipo amministrativo da altri comandi.

4.52.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

4.52.4 Classi figlie

- Actorbase.server.messages.query.admin.ActorPropertiesMessage
- Actorbase.server.messages.query.admin.PermissionsManagementMessage
- Actorbase.server.messages.query.admin.UsersManagementMessage

4.53 Actorbase.server.messages.query.admin.ActorPropertiesMessages

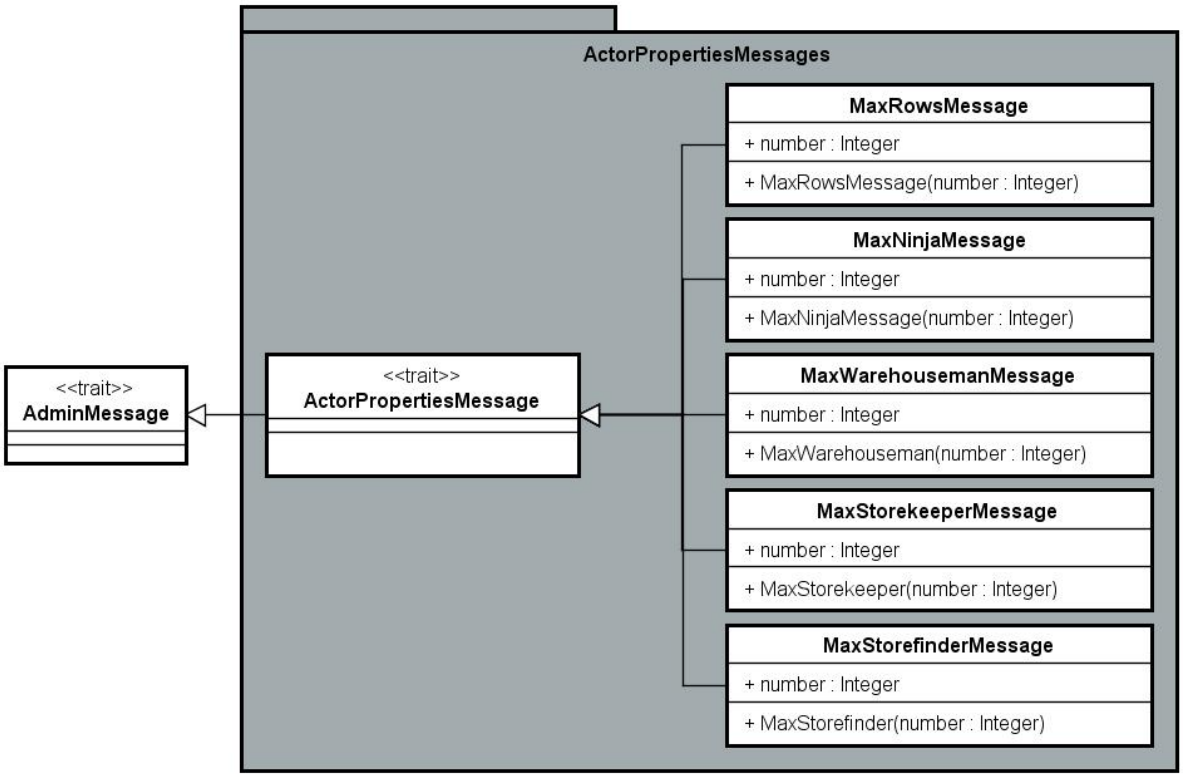


Figura 15: Componente Actorbase.server.messages.query.admin.ActorPropertiesMessages

4.53.1 Descrizione

Package contenente i messaggi per modificare le proprietà degli attori che compongono il sistema.

4.53.2 Classi

- Actorbase.server.messages.query.admin.ActorPropertiesMessages.MaxRowMessage
- Actorbase.server.messages.query.admin.ActorPropertiesMessages.MaxNinjaMessage
- Actorbase.server.messages.query.admin.ActorPropertiesMessages.MaxWarehousemanMessage
- Actorbase.server.messages.query.admin.ActorPropertiesMessages.MaxStorekeeperMessage
- Actorbase.server.messages.query.admin.ActorPropertiesMessages.MaxStorefinderMessage

4.53.3 Trait

- Actorbase.server.messages.query.admin.ActorPropertiesMessages.ActorPropertiesMessage

4.54 Actorbase.server.messages.query.admin.ActorPropertiesMessages.ActorPropertiesM

4.54.1 Descrizione

Trait per identificare i messaggi per modificare le proprietà degli attori che compongono il sistema.

4.54.2 Utilizzo

Viene utilizzato per poter distinguere i comandi per modificare le proprietà degli attori che compongono il sistema.

4.54.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.actors.Storekeeper:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione entrante, creazione e utilizzo.

4.54.4 Classi figlie

- Actorbase.server.messages.query.admin.ActorPropertiesMessages.MaxRowMessage
- Actorbase.server.messages.query.admin.ActorPropertiesMessages.MaxNinjaMessage
- Actorbase.server.messages.query.admin.ActorPropertiesMessages.MaxWarehousemanMessage
- Actorbase.server.messages.query.admin.ActorPropertiesMessages.MaxStorekeeperMessage
- Actorbase.server.messages.query.admin.ActorPropertiesMessages.MaxStorefinderMessage

4.55 Actorbase.server.messages.query.admin.ActorPropertiesMessages.MaxRowMessage

4.55.1 Descrizione

Questa classe rappresenta un messaggio per modificare il numero di righe massimo di una mappa che uno **Storekeeper** può gestire.

4.55.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage fermandosi all'attore **Storekeeper**.

4.55.3 Trait implementati

- Actorbase.server.messages.query.admin.ActorPropertiesMessages.ActorPropertiesMessage

4.56 Actorbase.server.messages.query.admin.ActorPropertiesMessages.MaxNinjaMessag

4.56.1 Descrizione

Questa classe rappresenta un messaggio per modificare il numero di attori **Ninja** assegnati ad ogni **Storekeeper**.

4.56.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage fermandosi all'attore **Storefinder**.

4.56.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione entrante, creazione e utilizzo.

4.56.4 Trait implementati

- Actorbase.server.messages.query.admin.ActorPropertiesMessage

4.57 Actorbase.server.messages.query.admin.ActorPropertiesMessages.MaxWarehousem

4.57.1 Descrizione

Questa classe rappresenta un messaggio per modificare il numero di attori **Warehouseman** assegnati ad ogni **Storekeeper**.

4.57.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage fermandosi all'attore **Storefinder**.

4.57.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione entrante, creazione e utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione entrante, creazione e utilizzo.

4.57.4 Trait implementati

- Actorbase.server.messages.query.admin.ActorPropertiesMessage

4.58 Actorbase.server.messages.query.admin.ActorPropertiesMessages.MaxStorekeeperN

4.58.1 Descrizione

Questa classe rappresenta un messaggio per modificare il numero massimo di attori **Storekeeper** gestiti da ogni **Storefinder**.

4.58.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage fermandosi all'attore **Storemanager**.

4.58.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione entrante, creazione e utilizzo.

4.58.4 Trait implementati

- Actorbase.server.messages.query.admin.ActorPropertiesMessage

4.59 Actorbase.server.messages.query.admin.ActorPropertiesMessages.MaxStorefinderM

4.59.1 Descrizione

Questa classe rappresenta un messaggio per modificare il numero massimo di attori **Storefinder** gestiti da ogni **Usermanager**.

4.59.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage fermandosi all'attore **Main**.

4.59.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

4.59.4 Trait implementati

- Actorbase.server.messages.query.admin.ActorPropertiesMessage

4.60 Actorbase.server.messages.query.admin.PermissionsManagementMessages

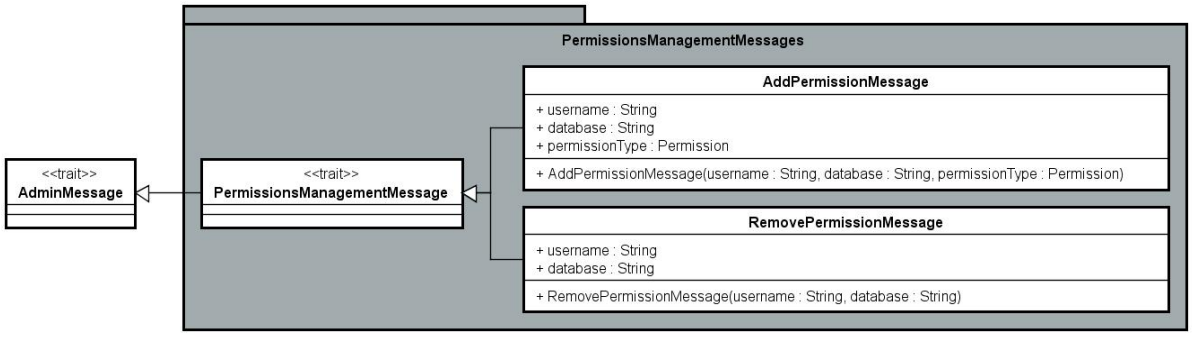


Figura 16: Componente Actorbase.server.messages.query.admin.PermissionsManagementMessages

4.60.1 Descrizione

Pacchetto contenente i messaggi che permettono di gestire i permessi assegnati agli utenti.

4.60.2 Classi

- Actorbase.server.messages.query.admin.PermissionsManagementMessages.AddPermissionMessage
- Actorbase.server.messages.query.admin.PermissionsManagementMessages.RemovePermissionMessage
- Actorbase.server.messages.query.admin.PermissionsManagementMessages.ListPermissionMessage

4.60.3 Trait

- Actorbase.server.messages.query.admin.PermissionsManagementMessages.PermissionsManagementMessage

4.61 Actorbase.server.messages.query.admin.PermissionsManagementMessages.Permissi

4.61.1 Descrizione

Trait che rappresenta i messaggi per gestire i permessi assegnati agli utenti.

4.61.2 Utilizzo

Viene utilizzato per distinguere i messaggi per gestire i permessi assegnati agli utenti.

4.61.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

4.61.4 Classi figlie

- Actorbase.server.messages.query.admin.PermissionsManagementMessages.AddPermissionMessage
- Actorbase.server.messages.query.admin.PermissionsManagementMessages.RemovePermissionMessage
- Actorbase.server.messages.query.admin.PermissionsManagementMessages.ListPermissionMessage

4.62 Actorbase.server.messages.query.admin.PermissionsManagementMessages.AddPerr

4.62.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di aggiunta di permessi ad uno specifico utente.

4.62.2 Utilizzo

Viene utilizzata per aggiungere permessi ad uno specifico utente.

4.62.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

4.62.4 Trait implementati

- Actorbase.server.messages.query.admin.PermissionsManagementMessages.PermissionsManagementMessage

4.63 Actorbase.server.messages.query.admin.PermissionsManagementMessages.Removel

4.63.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di rimozione di permessi ad uno specifico utente.

4.63.2 Utilizzo

Viene utilizzata per rimuovere permessi ad uno specifico utente.

4.63.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

4.63.4 Trait implementati

- Actorbase.server.messages.query.admin.PermissionsManagementMessages.PermissionsManagementMessage

4.64 Actorbase.server.messages.query.admin.PermissionsManagementMessages.ListPerm

4.64.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di visualizzazione di permessi di uno specifico utente.

4.64.2 Utilizzo

Viene utilizzata per mostrare i permessi per ogni database di uno specifico utente.

4.64.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

4.64.4 Trait implementati

- Actorbase.server.messages.query.admin.PermissionsManagementMessages.PermissionsManagementMessage

4.65 Actorbase.server.messages.query.admin.UserManagementMessages

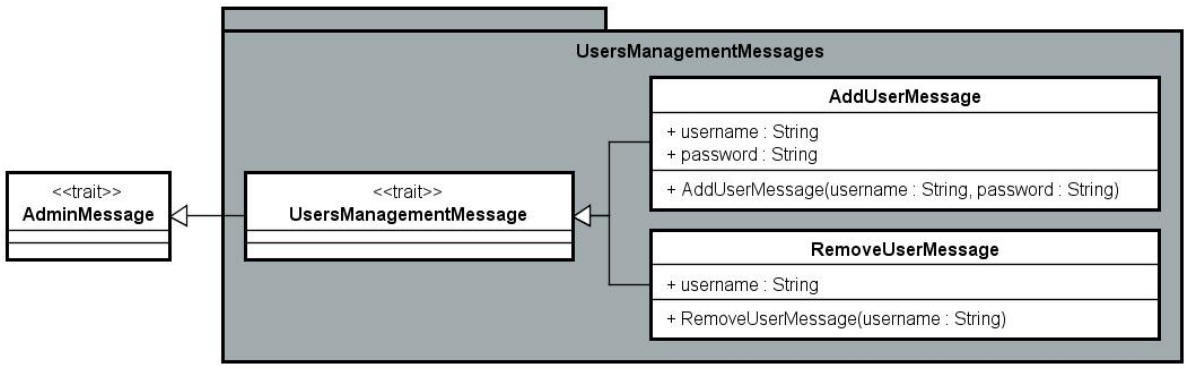


Figura 17: Componente Actorbase.server.messages.query.admin.UserManagementMessages

4.65.1 Descrizione

Pacchetto contenente i messaggi per gestire gli utenti nel database.

4.65.2 Classi

- Actorbase.server.messages.query.admin.UserManagementMessages.AddUserMessage
- Actorbase.server.messages.query.admin.UserManagementMessages.RemoveUserMessage
- Actorbase.server.messages.query.admin.UserManagementMessages.ListUserMessage

4.65.3 Trait

- Actorbase.server.messages.query.admin.UserManagementMessages.UserManagementMessage

4.66 Actorbase.server.messages.query.admin.UserManagementMessages.UserManageme

4.66.1 Descrizione

Trait che rappresenta i messaggi per gestire gli utenti nel database.

4.66.2 Utilizzo

Viene utilizzato per distinguere i messaggi per gestire gli utenti nel database.

4.66.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

4.66.4 Classi figlie

- Actorbase.server.messages.query.admin.UserManagementMessages.AddUserMessage
- Actorbase.server.messages.query.admin.UserManagementMessages.RemoveUserMessage
- Actorbase.server.messages.query.admin.UserManagementMessages.ListUserMessage

4.67 Actorbase.server.messages.query.admin.UserManagementMessages.AddUserMessag

4.67.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di aggiunta di un utente.

4.67.2 Utilizzo

Viene utilizzata per aggiungere un utente al sistema.

4.67.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

4.67.4 Trait implementati

- Actorbase.server.messages.query.admin.UserManagementMessages.UserManagementMessage

4.68 Actorbase.server.messages.query.admin.UserManagementMessages.RemoveUserMe

4.68.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di rimozione di un utente.

4.68.2 Utilizzo

Viene utilizzata per rimuovere un utente dal sistema.

4.68.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

4.68.4 Trait implementati

- Actorbase.server.messages.query.admin.UserManagementMessages.UserManagementMessage

4.69 Actorbase.server.messages.query.admin.UserManagementMessages.ListUserMessag

4.69.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di visualizzazione di tutti gli utenti presenti.

4.69.2 Utilizzo

Viene utilizzata per visualizzare tutti gli utenti presenti nel sistema.

4.69.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

4.69.4 Trait implementati

- Actorbase.server.messages.query.admin.UserManagementMessages.UserManagementMessage

4.70 Actorbase.server.messages.query.user

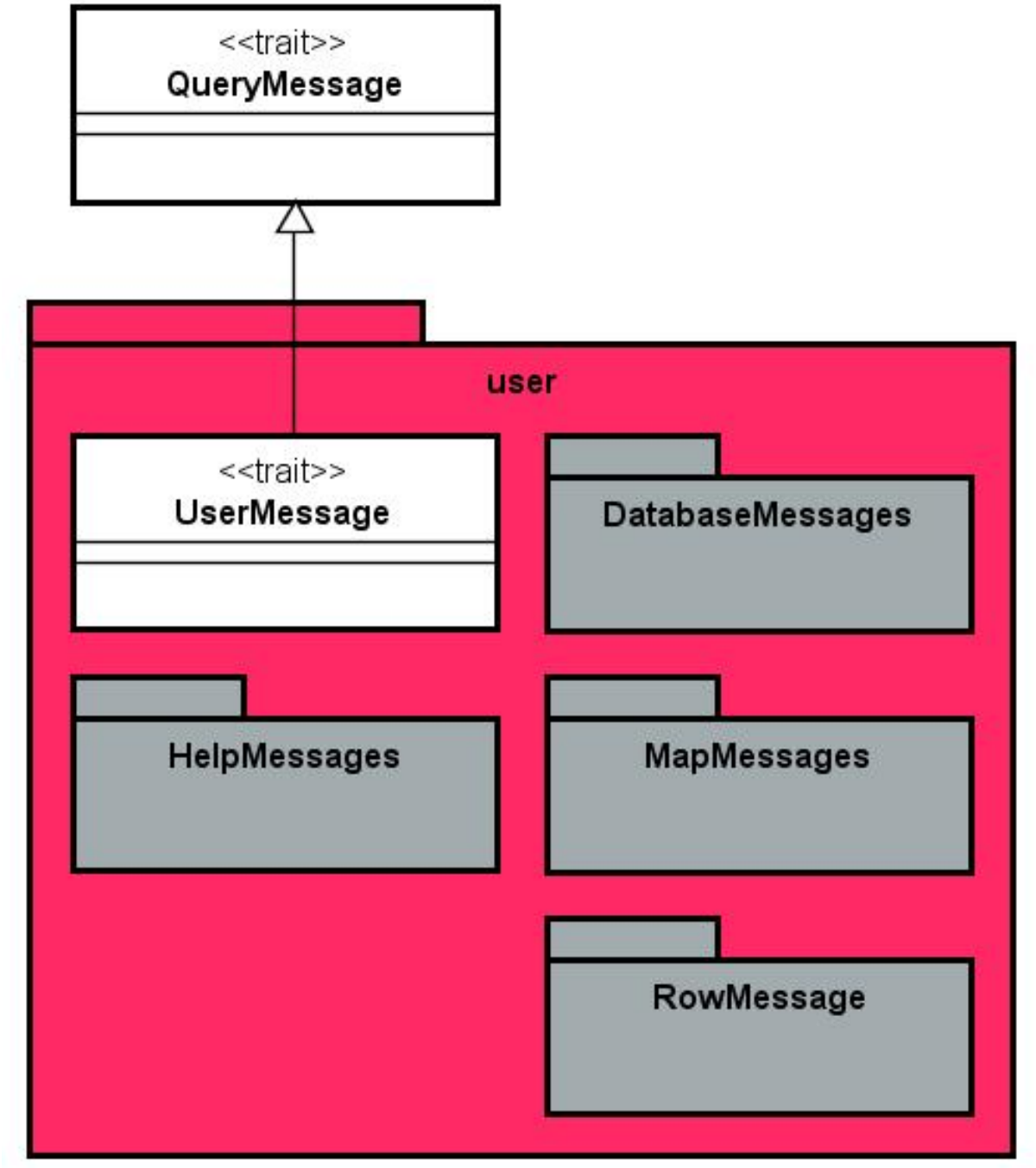


Figura 18: Componente Actorbase.server.messages.query.user

4.70.1 Descrizione

Pacchetto contenente tutti i messaggi che rappresentano richieste non amministrative.

4.70.2 Package Figli

- Actorbase.server.messages.query.user.RowMessages
- Actorbase.server.messages.query.user.MapMessages

- Actorbase.server.messages.query.user.DatabaseMessages
- Actorbase.server.messages.query.user.HelpMessages

4.70.3 Trait

- Actorbase.server.messages.query.user.UserMessage

4.71 Actorbase.server.messages.query.user.UserMessage

4.71.1 Descrizione

Trait che rappresenta richieste non amministrative.

4.71.2 Utilizzo

Viene utilizzato per distinguere i messaggi non amministrativi.

4.71.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione di utilizzo.
- **Actorbase.server.actors.Storekeeper:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

4.71.4 Classi figlie

- Actorbase.server.messages.query.user.DatabaseMessages.DatabaseMessage
- Actorbase.server.messages.query.user.MapMessages.MapMessage
- Actorbase.server.messages.query.user.RowMessages.RowMessage
- Actorbase.server.messages.query.user.HelpMessages.HelpMessage

4.72 Actorbase.server.messages.query.user.RowMessages

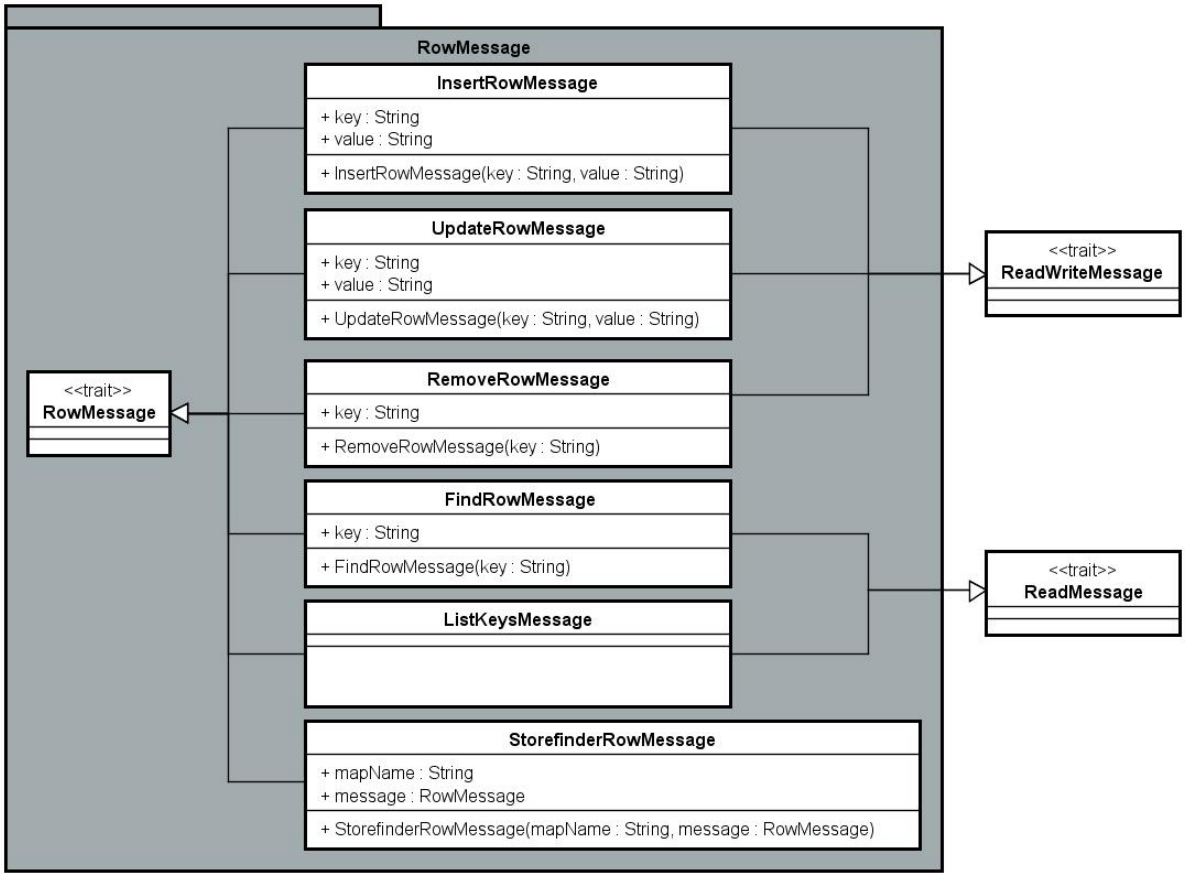


Figura 19: Componente Actorbase.server.messages.query.user.RowMessages

4.72.1 Descrizione

Pacchetto contenente i messaggi che rappresentano le query a livello di riga.

4.72.2 Classi

- Actorbase.server.messages.query.user.RowMessages.InsertRowMessage
- Actorbase.server.messages.query.user.RowMessages.UpdateRowMessage
- Actorbase.server.messages.query.user.RowMessages.RemoveRowMessage
- Actorbase.server.messages.query.user.RowMessages.FindRowMessage
- Actorbase.server.messages.query.user.RowMessages.ListKeysMessage

4.72.3 Trait

- Actorbase.server.messages.query.user.RowMessages.RowMessage

4.73 Actorbase.server.messages.query.user.RowMessages.RowMessage

4.73.1 Descrizione

Trait che rappresenta i messaggi contenenti richieste a livello di riga.

4.73.2 Utilizzo

Viene utilizzato per poter riconoscere i messaggi contenenti richieste a livello di riga.

4.73.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione di utilizzo.
- **Actorbase.server.actors.Storekeeper:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

4.73.4 Classi figlie

- Actorbase.server.messages.query.user.RowMessages.InsertRowMessage
- Actorbase.server.messages.query.user.RowMessages.UpdateRowMessage
- Actorbase.server.messages.query.user.RowMessages.RemoveRowMessage
- Actorbase.server.messages.query.user.RowMessages.FindRowMessage
- Actorbase.server.messages.query.user.RowMessages.ListKeysMessage

4.74 Actorbase.server.messages.query.user.RowMessages.InsertRowMessage

4.74.1 Descrizione

Classe che rappresenta un messaggio contente la richiesta di inserimento di una riga in una mappa.

4.74.2 Utilizzo

Effettua l'intero percorso descritto in Actorbase.server.messages.query.QueryMessage.

4.74.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione di utilizzo.
- **Actorbase.server.actors.Storekeeper:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

4.74.4 Trait implementati

- Actorbase.server.messages.query.user.RowMessages.RowMessage

4.75 Actorbase.server.messages.query.user.RowMessages.UpdateRowMessage

4.75.1 Descrizione

Classe che rappresenta un messaggio contente la richiesta di modifica di una riga in una mappa.

4.75.2 Utilizzo

Effettua l'intero percorso descritto in Actorbase.server.messages.query.QueryMessage.

4.75.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione di utilizzo.
- **Actorbase.server.actors.Storekeeper:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

4.75.4 Trait implementati

- Actorbase.server.messages.query.user.RowMessages.RowMessage

4.76 Actorbase.server.messages.query.user.RowMessages.RemoveRowMessage

4.76.1 Descrizione

Classe che rappresenta un messaggio contente la richiesta di rimozione di una riga in una mappa.

4.76.2 Utilizzo

Effettua l'intero percorso descritto in Actorbase.server.messages.query.QueryMessage.

4.76.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione di utilizzo.
- **Actorbase.server.actors.Storekeeper:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

4.76.4 Trait implementati

- Actorbase.server.messages.query.user.RowMessages.RowMessage

4.77 Actorbase.server.messages.query.user.RowMessages.FindRowMessage

4.77.1 Descrizione

Classe che rappresenta un messaggio contente la richiesta di ricercare una riga in una mappa.

4.77.2 Utilizzo

Effettua l'intero percorso descritto in Actorbase.server.messages.query.QueryMessage.

4.77.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione di utilizzo.
- **Actorbase.server.actors.Storekeeper:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

4.77.4 Trait implementati

- Actorbase.server.messages.query.user.RowMessages.RowMessage

4.78 Actorbase.server.messages.query.user.RowMessages.ListKeysMessage

4.78.1 Descrizione

Classe che rappresenta un messaggio contente la richiesta di visualizzazione di tutte le chiavi presenti in una mappa.

4.78.2 Utilizzo

Effettua l'intero percorso descritto in Actorbase.server.messages.query.QueryMessage.

4.78.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione di utilizzo.
- **Actorbase.server.actors.Storekeeper:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

4.78.4 Trait implementati

- Actorbase.server.messages.query.user.RowMessages.RowMessage

4.79 Actorbase.server.messages.query.user.MapMessages

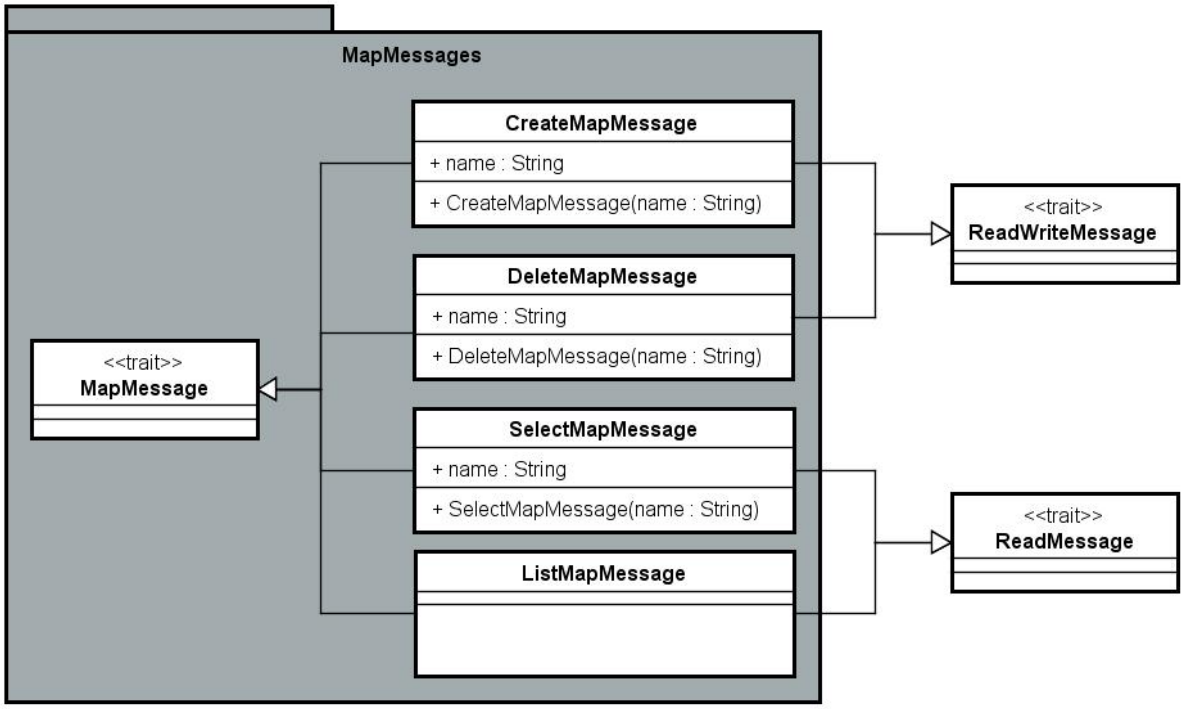


Figura 20: Componente Actorbase.server.messages.query.user.MapMessages

4.79.1 Descrizione

Pacchetto contenente i messaggi che rappresentano le query a livello di mappa.

4.79.2 Classi

- Actorbase.server.messages.query.user.MapMessages.CreateMapMessage
- Actorbase.server.messages.query.user.MapMessages.DeleteMapMessage
- Actorbase.server.messages.query.user.MapMessages.SelectMapMessage
- Actorbase.server.messages.query.user.MapMessages.ListMapMessage

4.79.3 Trait

- Actorbase.server.messages.query.user.MapMessages.MapMessage

4.80 Actorbase.server.messages.query.user.MapMessages.MapMessage

4.80.1 Descrizione

Trait che rappresenta i messaggi contenenti richieste a livello di mappa.

4.80.2 Utilizzo

Viene utilizzato per poter riconoscere i messaggi contenenti richieste a livello di mappa.

4.80.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

4.80.4 Classi figlie

- Actorbase.server.messages.query.user.MapMessages.CreateMapMessage
- Actorbase.server.messages.query.user.MapMessages.DeleteMapMessage
- Actorbase.server.messages.query.user.MapMessages.SelectMapMessage
- Actorbase.server.messages.query.user.MapMessages.ListMapMessage

4.81 Actorbase.server.messages.query.user.MapMessages.CreateMapMessage

4.81.1 Descrizione

Classe che rappresenta un messaggio contente la richiesta di creazione di una mappa in un database.

4.81.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage saltando l'attore **Storekeeper**.

4.81.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

4.81.4 Trait implementati

- Actorbase.server.messages.query.user.MapMessages.MapMessage

4.82 Actorbase.server.messages.query.user.MapMessages.DeleteMapMessage

4.82.1 Descrizione

Classe che rappresenta un messaggio contente la richiesta di rimozione di una mappa in un database.

4.82.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage saltando l'attore **Storekeeper**.

4.82.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

4.82.4 Trait implementati

- Actorbase.server.messages.query.user.MapMessages.MapMessage

4.83 Actorbase.server.messages.query.user.MapMessages.SelectMapMessage

4.83.1 Descrizione

Classe che rappresenta un messaggio contente la richiesta di selezione di una mappa in un database.

4.83.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage saltando l'attore **Storekeeper**.

4.83.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

4.83.4 Trait implementati

- Actorbase.server.messages.query.user.MapMessages.MapMessage

4.84 Actorbase.server.messages.query.user.MapMessages.ListMapMessage

4.84.1 Descrizione

Classe che rappresenta un messaggio contente la richiesta di visualizzazione di tutte le mappe presenti in un database.

4.84.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage saltando l'attore **Storekeeper**.

4.84.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Storemanager:** relazione di utilizzo.
- **Actorbase.server.actors.Storefinder:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

4.84.4 Trait implementati

- Actorbase.server.messages.query.user.MapMessages.MapMessage

4.85 Actorbase.server.messages.query.user.DatabaseMessages

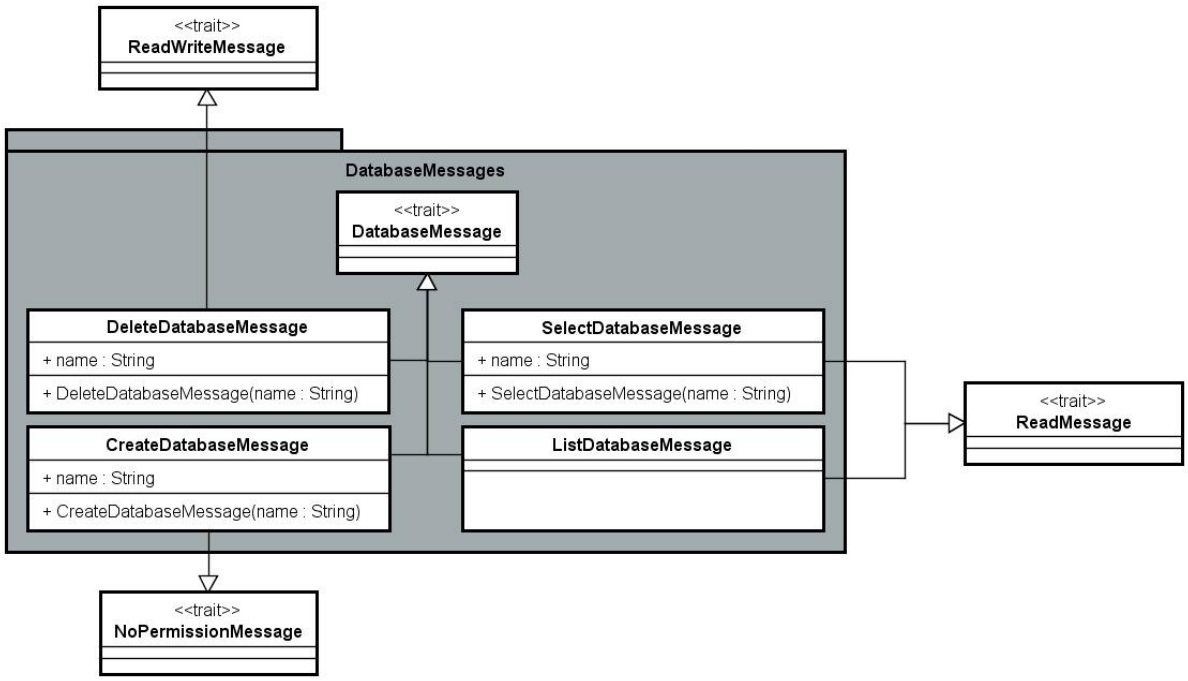


Figura 21: Componente Actorbase.server.messages.query.user.DatabaseMessages

4.85.1 Descrizione

Pacchetto contenente i messaggi che rappresentano le query a livello di database.

4.85.2 Classi

- Actorbase.server.messages.query.user.DatabaseMessages.CreateDatabaseMessage
- Actorbase.server.messages.query.user.DatabaseMessages.DeleteDatabaseMessage
- Actorbase.server.messages.query.user.DatabaseMessages.SelectDatabaseMessage
- Actorbase.server.messages.query.user.DatabaseMessages.ListDatabaseMessage

4.85.3 Trait

- Actorbase.server.messages.query.user.DatabaseMessages.DatabaseMessage

4.86 Actorbase.server.messages.query.user.DatabaseMessages.DatabaseMessage

4.86.1 Descrizione

Trait che rappresenta i messaggi contenenti richieste a livello di database.

4.86.2 Utilizzo

Viene utilizzato per poter distinguere i messaggi contenenti richieste a livello di database.

4.86.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

4.86.4 Classi figlie

- Actorbase.server.messages.query.user.DatabaseMessages.CreateDatabaseMessage
- Actorbase.server.messages.query.user.DatabaseMessages.DeleteDatabaseMessage
- Actorbase.server.messages.query.user.DatabaseMessages.SelectDatabaseMessage
- Actorbase.server.messages.query.user.DatabaseMessages.ListDatabaseMessage

4.87 Actorbase.server.messages.query.user.DatabaseMessages.CreateDatabaseMessage

4.87.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di creazione di un database.

4.87.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage fermandosi all'attore **Main**. Viene inoltre inoltrato ai **Warehouseman**.

4.87.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

4.87.4 Trait implementati

- Actorbase.server.messages.query.user.DatabaseMessages.DatabaseMessage

4.88 Actorbase.server.messages.query.user.DatabaseMessages.DeleteDatabaseMessage

4.88.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di rimozione di un database.

4.88.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage fermandosi all'attore **Main**. Viene inoltre inoltrato ai **Warehouseman**.

4.88.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

4.88.4 Trait implementati

- Actorbase.server.messages.query.user.DatabaseMessages.DatabaseMessage

4.89 Actorbase.server.messages.query.user.DatabaseMessages.SelectDatabaseMessage

4.89.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di selezione di un database.

4.89.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage fermandosi all'attore **Main**. Viene inoltre inoltrato ai **Warehouseman**.

4.89.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

4.89.4 Trait implementati

- Actorbase.server.messages.query.user.DatabaseMessages.DatabaseMessage

4.90 Actorbase.server.messages.query.user.DatabaseMessages.ListDatabaseMessage

4.90.1 Descrizione

Classe che rappresenta un messaggio contenente la richiesta di visualizzazione dell'intera lista di database presenti.

4.90.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage fermandosi all'attore **Main**.

4.90.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.
- **Actorbase.server.actors.Warehouseman:** relazione di utilizzo.

4.90.4 Trait implementati

- Actorbase.server.messages.query.user.DatabaseMessages.DatabaseMessage

4.91 Actorbase.server.messages.query.user.HelpMessages

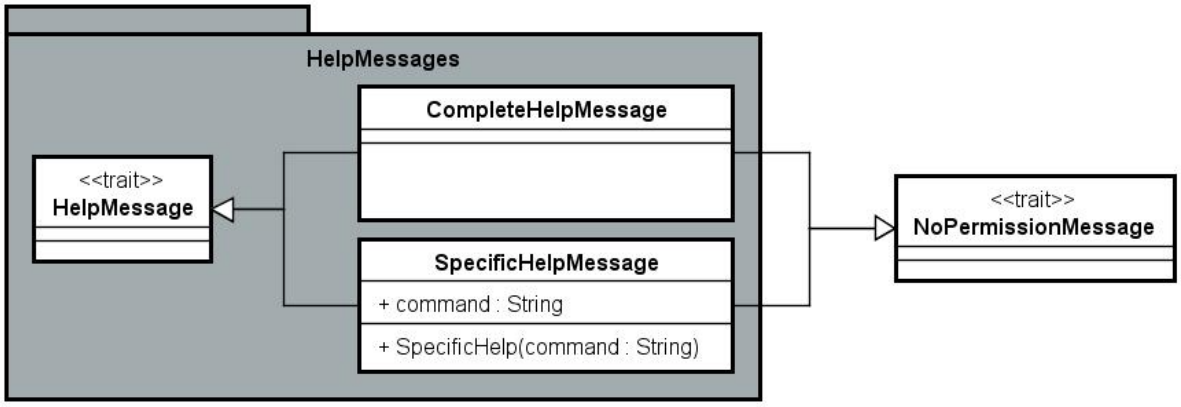


Figura 22: Componente Actorbase.server.messages.query.user.HelpMessages

4.91.1 Descrizione

Pacchetto contenente i messaggi che rappresentano le query di aiuto.

4.91.2 Classi

- Actorbase.server.messages.query.user.HelpMessages.CompleteHelp
- Actorbase.server.messages.query.user.HelpMessages.SpecificHelp

4.91.3 Trait

- Actorbase.server.messages.query.user.HelpMessages.HelpMessage

4.92 Actorbase.server.messages.query.user.HelpMessages.HelpMessage

4.92.1 Descrizione

Trait che rappresenta i messaggi contenenti richieste perla visualizzazione di aiuti.

4.92.2 Utilizzo

Viene utilizzata per poter distinguere i messaggi contenenti richieste per la visualizzazione di aiuti.

4.92.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

4.92.4 Classi figlie

- Actorbase.server.messages.query.user.HelpMessages.CompleteHelp
- Actorbase.server.messages.query.user.HelpMessages.SpecificHelp

4.93 Actorbase.server.messages.query.user.HelpMessages.CompleteHelp

4.93.1 Descrizione

Classe che rappresenta i messaggi contenenti richieste per la visualizzazione di tutti i comandi disponibili.

4.93.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage fermandosi all'attore **Main**.

4.93.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

4.93.4 Trait implementati

- Actorbase.server.messages.query.user.HelpMessages.HelpMessage

4.94 Actorbase.server.messages.query.user.HelpMessages.SpecificHelp

4.94.1 Descrizione

Classe che rappresenta i messaggi contenenti richieste per la visualizzazione di aiuto riguardante un comando specifico.

4.94.2 Utilizzo

Effettua il percorso descritto in Actorbase.server.messages.query.QueryMessage fermandosi all'attore **Main**.

4.94.3 Relazioni con altre classi

- **Actorbase.server.utils.Parser:** relazione entrante di creazione.
- **Actorbase.server.actors.Usermanager:** relazione di utilizzo.
- **Actorbase.server.actors.Main:** relazione di utilizzo.

4.94.4 Trait implementati

- Actorbase.server.messages.query.user.HelpMessages.HelpMessage

4.95 Actorbase.client

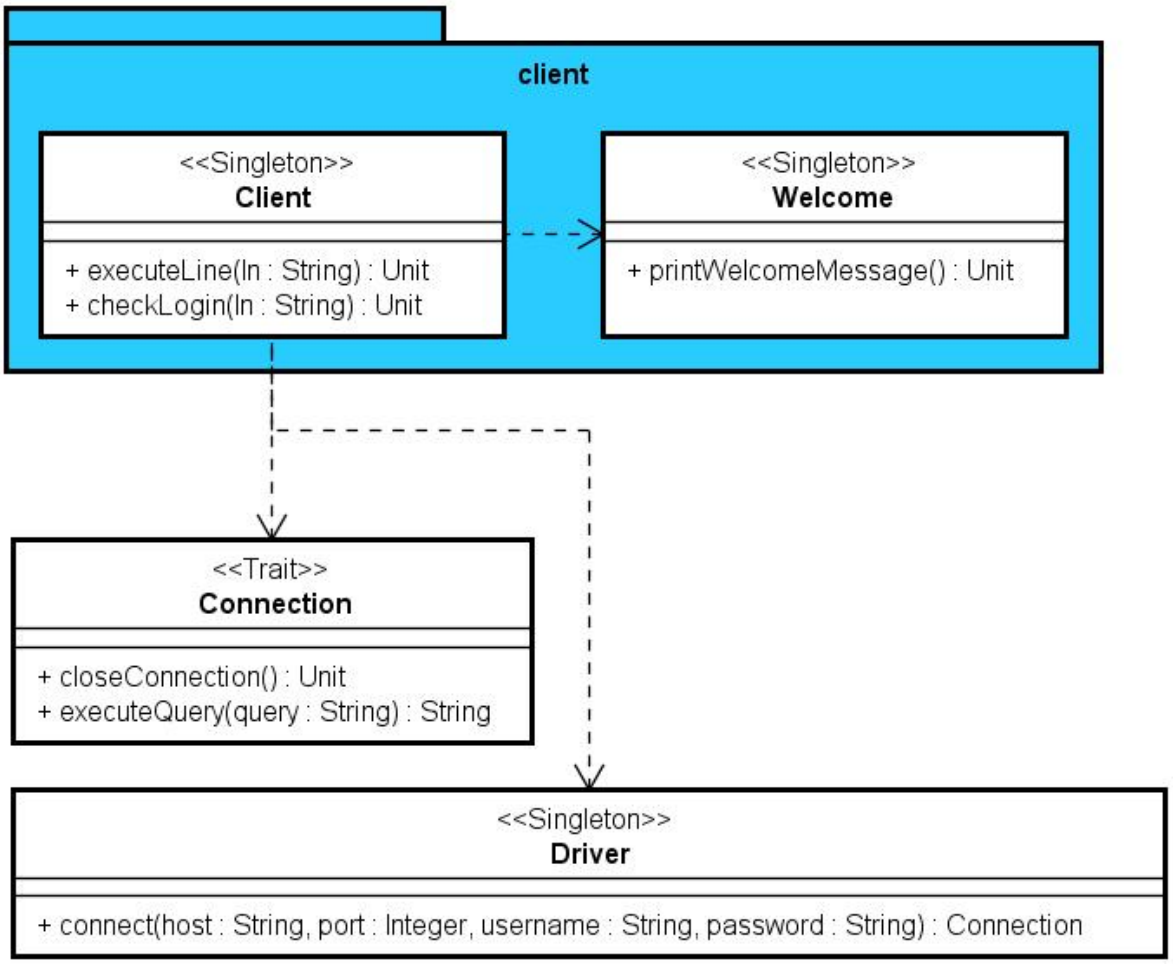


Figura 23: Componente Actorbase.client

4.95.1 Descrizione

Package per la componente lato client del sistema. È composto due classi che implementano il design pattern **Singleton**.

4.95.2 Classi

- Actorbase.client.Client
- Actorbase.client.Welcome

4.96 Actorbase.client.Client

4.96.1 Descrizione

Questa classe fornisce un'interfaccia a linea di comando e permette all'utente di inserire i comandi che verranno inviati al server. Questa classe implementa il design pattern **Singleton**.

4.96.2 Utilizzo

Viene utilizzata per distinguere i comandi di connessione, disconnessione e altri comandi possibili. Inoltre utilizza il driver per comunicare con il server.

4.96.3 Relazioni con altre classi

- **Actorbase.client.Welcome:** relazione di utilizzo.
- **Actorbase.driver.Driver:** relazione di utilizzo.
- **Actorbase.driver.Connection:** relazione di utilizzo.

4.97 Actorbase.client.Welcome

4.97.1 Descrizione

Classe si supporto per stampare un messaggio di benvenuto sulla consola del client.

4.97.2 Utilizzo

Viene utilizzata dal client per stampare un messaggio di benvenuto all'avvio dell'interfaccia a linea di comando.

4.97.3 Relazioni con altre classi

- **Actorbase.client.Client**

4.98 Actorbase.driver

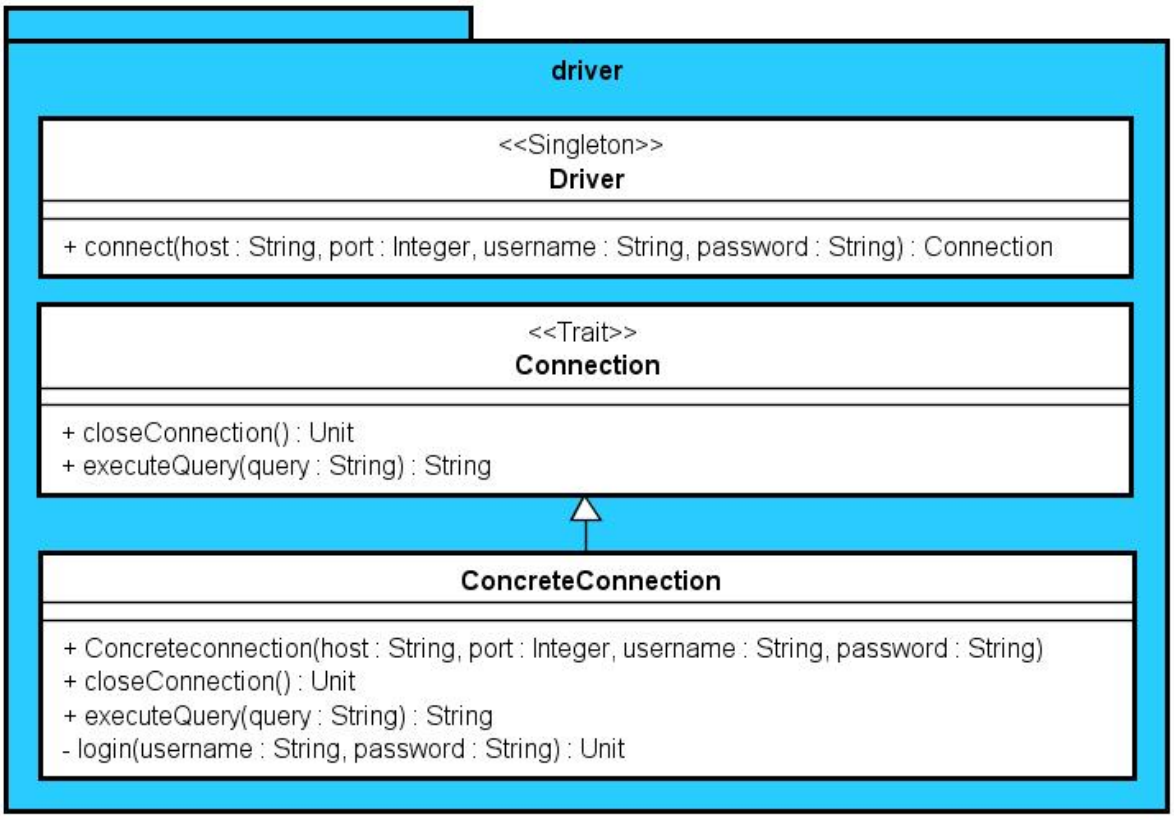


Figura 24: Componente Actorbase.driver

4.98.1 Descrizione

Package per la componente driver del sistema. È composto una classe che implementa il design pattern **Singleton**, un trait e una sua implementazione.

4.98.2 Classi

- Actorbase.driver.Driver
- Actorbase.driver.ConcreteConnection

4.98.3 Trait

- Actorbase.driver.Connection

4.99 Actorbase.driver.Connection

4.99.1 Descrizione

Trait utilizzato per effettuare il login su un determinato server.

4.99.2 Utilizzo

Costruisce una query nel modo corretto ricevendo un comando in formato stringa dal client.

4.99.3 Relazioni con altre classi

- **Actorbase.driver.Driver:** relazione entrante, creazione.
- **Actorbase.driver.Client:** relazione di utilizzo.

4.99.4 Classi figlie

- Actorbase.driver.ConcreteConnection

4.100 Actorbase.driver.ConcreteConnection

4.100.1 Descrizione

Classe utilizzata per effettuare il login su un determinato server. Apre una connessione verso il server specificato.

4.100.2 Utilizzo

Costruisce una query nel modo corretto ricevendo un comando in formato stringa dal client.

4.100.3 Relazioni con altre classi

- **Actorbase.driver.Driver:** relazione entrante, creazione.
- **Actorbase.driver.Client:** relazione di utilizzo.

4.100.4 Trait implementati

- Actorbase.driver.Connection

4.101 Actorbase.driver.Driver

4.101.1 Descrizione

Classe che implementa il design pattern **Singleton**,

4.101.2 Utilizzo

Viene utilizzato per creare una connessione che viene restituita al client.

4.101.3 Relazioni con altre classi

- **Actorbase.client.Client:** relazione di utilizzo.
- **Actorbase.driver.ConcreteConnection:** relazione uscente, creazione.

5 Diagrammi delle attività

Segue il diagramma delle attività che mostra l'interazione dell'utente con il client.

5.0.1 Diagramma attività principale

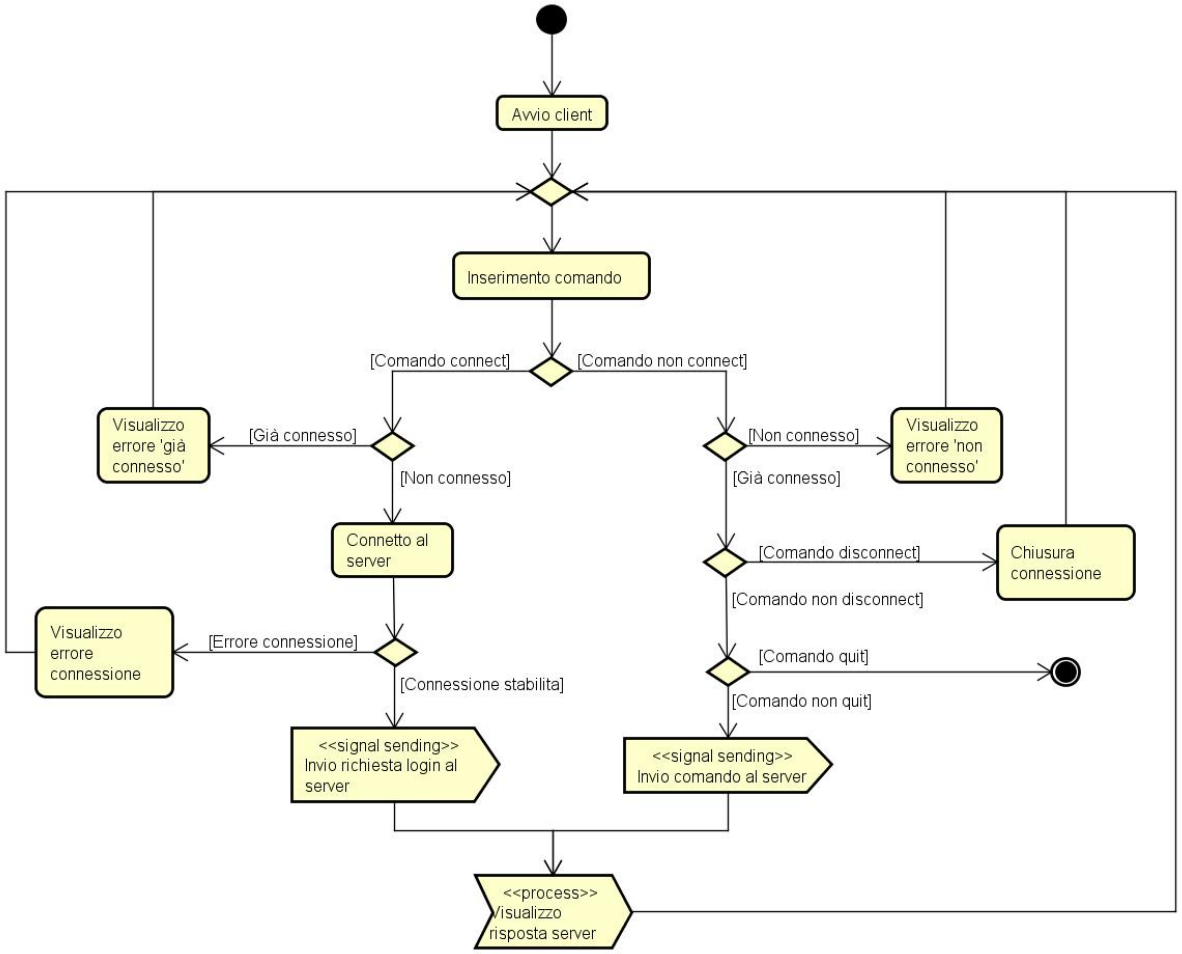


Figura 25: Diagramma attività principale

Dopo aver avviato il client l'utente inserire dei comandi. Se il comando è di connect ed il client non è connesso quest'ultimo si connette ed invia un comando di login al server, altrimenti stampa un messaggio di errore. Se il comando non è di connect ed il client è connesso quest'ultimo controlla che non sia ne un comando di disconnessione ne uno di chiusura ed invia il comando al server.

6 Diagrammi di sequenza

In questa sezione verranno illustrati e descritti i principali diagrammi di sequenza realizzati. Questi ultimi illustrano le azioni compiute dal server per l'avvio e la gestire le richieste utente.

Per esplicitare l'invio di un messaggio tra due attori, nei diagrammi viene chiamato il metodo *receive()* dell'attore che riceve il messaggio da parte di chi lo invia. Questo è dovuto al fatto che tutte le classi ereditano dall'interfaccia *Actor* messa a disposizione da *Akka* senza effettuare l'override del metodo per inviare i messaggi; mentre sovrascrivono il metodo di ricezione per gestire i messaggi in entrata.

Trattandosi di diagrammi di sequenza non molto specifici, alcuni attori effettuano operazioni senza chiamare dei metodi specifici già dichiarati.

6.1 Avvio

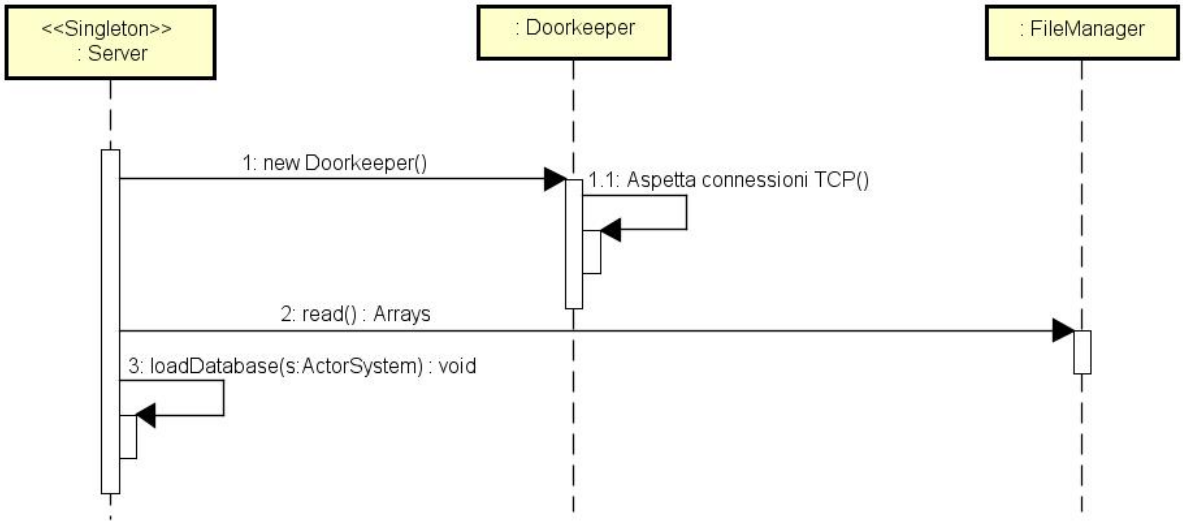


Figura 26: Diagramma di sequenza - avvio del server.

Nel diagramma precedente è possibile visualizzare quali sono le operazioni che vengono eseguite per avviare il server. Esso crea i vari Doorkeeper, i punti di accesso dall'esterno, che resteranno in ascolto di connessioni in entrata. Poi legge le configurazioni salvate su disco e carica i database.

6.2 Nuova connessione

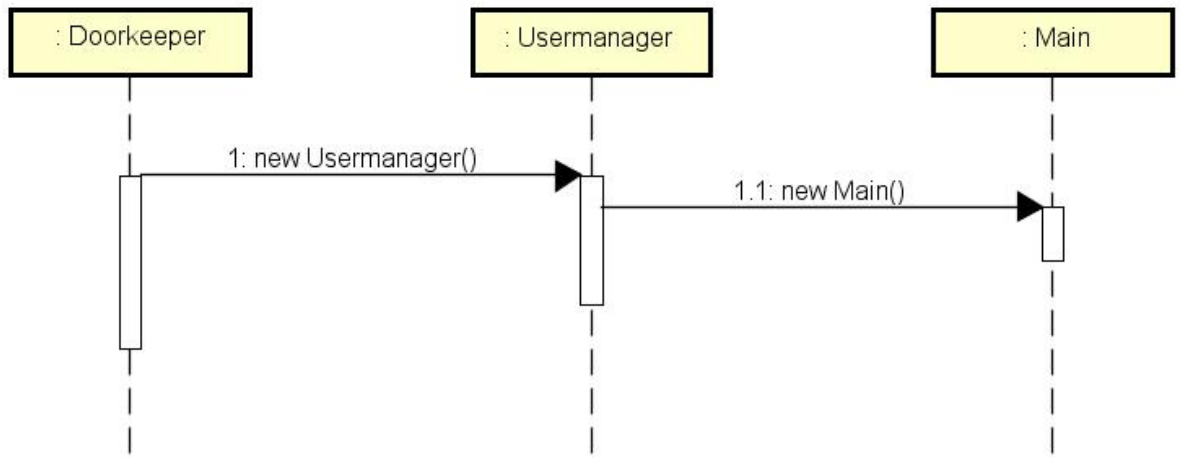


Figura 27: Diagramma di sequenza - nuova connessione al server.

Nel diagramma precedente è possibile visualizzare quali sono le operazioni che vengono eseguite quando un client si collega al socket TCP di un Doorkeeper. Per ogni nuova connessione l'attore Doorkeeper crea un attore Usermanager il quale crea un attore Main. Usermanager gestisce ogni richiesta proveniente dalla connessione a lui associata, nello specifico trasforma i comandi in forma testuale in messaggi da inoltrare al Main.

6.3 Comando utente

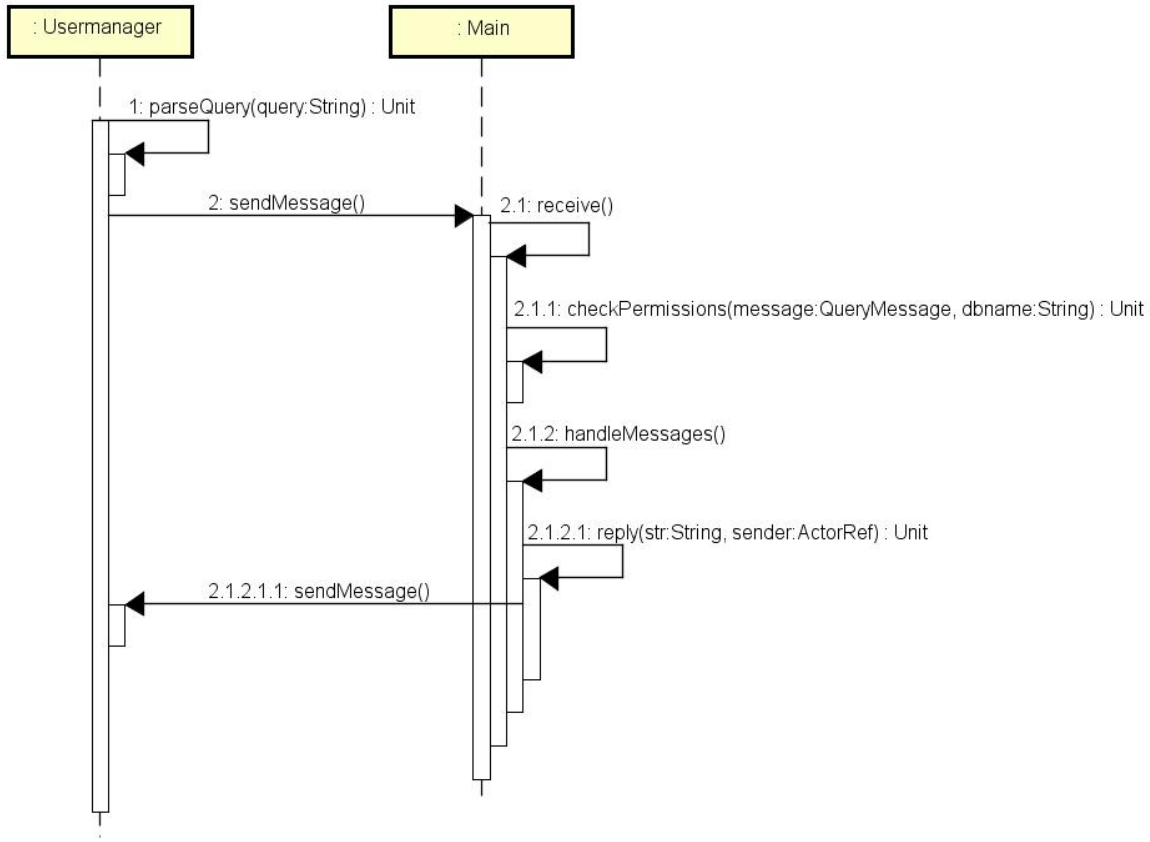


Figura 28: Diagramma di sequenza - gestione comando utente.

Nel diagramma precedente è possibile visualizzare quali sono le operazioni che vengono eseguite dagli attori Usermanager e Main alla ricezione di una richiesta dall'utente. Usermanager, dopo aver convertito i byte ricevuti in una stringa, crea un messaggio che rappresenta la richiesta dall'utente ed lo invia al Main. Il Main controlla che l'utente abbia i permessi per il tipo di richiesta, in caso affermativo gestisce, anche indirettamente, il messaggio e risponde.

6.4 Richiesta di creazione mappa

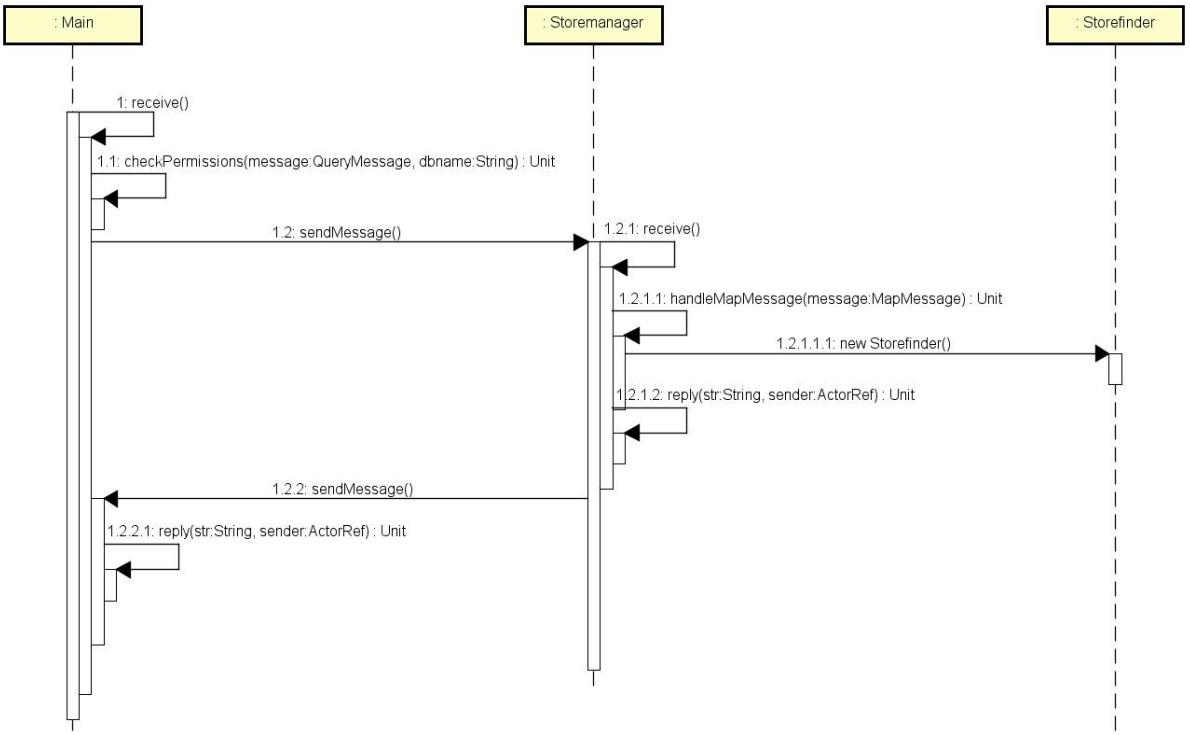


Figura 29: Diagramma di sequenza - gestione di richiesta di creazione mappa.

Nel diagramma precedente è possibile visualizzare quali sono le operazioni che vengono eseguite quando il Main riceve una richiesta di creazione di una mappa. Il Main alla ricezione di un messaggio, dopo aver controllato i permessi, ne controlla il tipo. Se è un messaggio di tipo mappa lo inoltra allo Storemanager che rappresenta il database precedentemente selezionato. Lo Storemanager crea un nuovo Storefinder che rappresenterà la mappa e risponde con l'esito che arriverà fino al client.

6.5 Richiesta di find

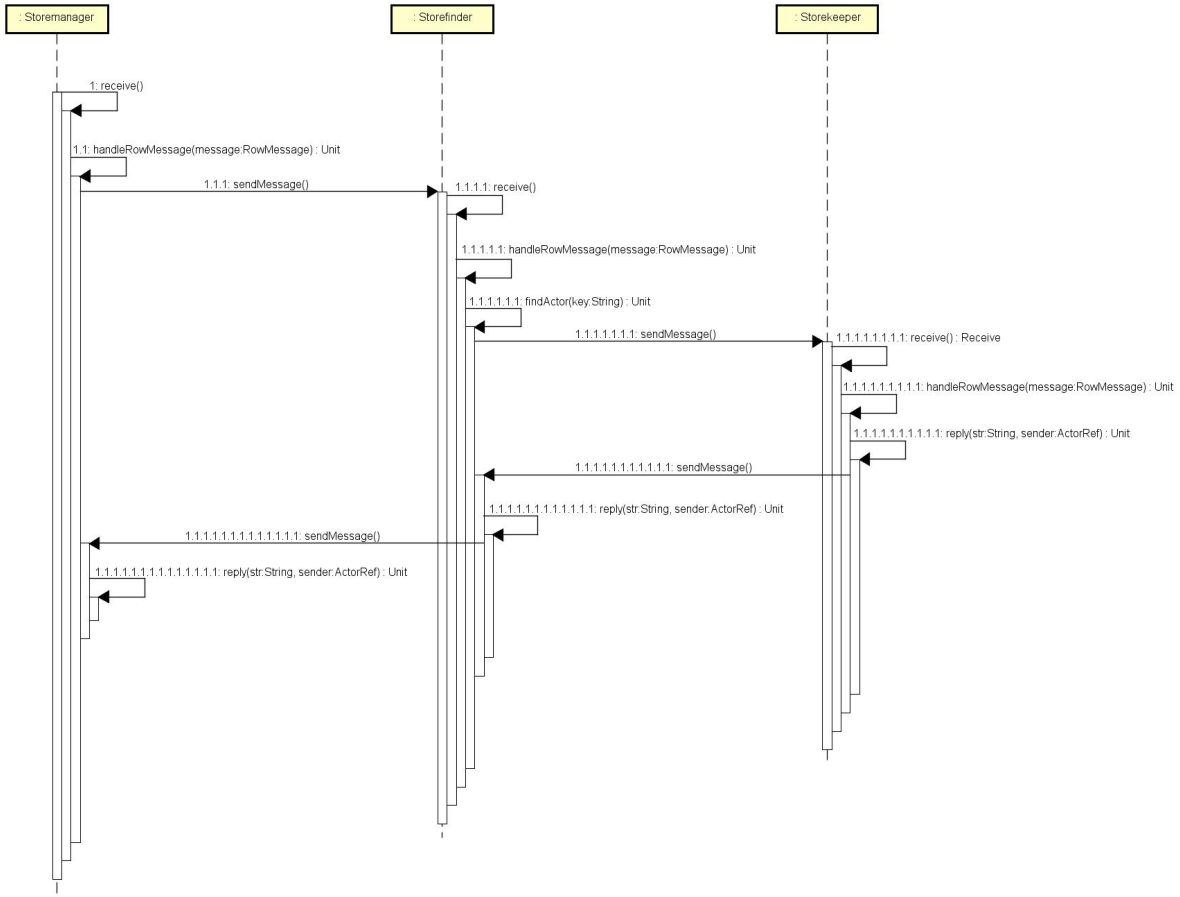


Figura 30: Diagramma di sequenza - gestione di richiesta di Find.

Nel diagramma precedente è possibile visualizzare quali sono le operazioni che vengono eseguite per gestire una richiesta di find. Il messaggio rappresentante un comando find viene inoltrato fino allo Storekeeper responsabile dell'area di mappa richiesta. Lo Storekeeper risponderà con il valore delle chiave o con un messaggio di errore. La risposta verrà poi inoltrata all'indietro fino al client.

7 Stime di fattibilità e di bisogno di risorse

L'architettura definita fino a questo punto è sufficiente per fornire una stima della fattibilità del prodotto e delle risorse richieste per la realizzazione.
Il gruppo inizialmente non aveva conoscenze sufficienti per stimare in modo appropriato la complessità dell'implementazione di un Database basato sulla logica ad attori. Grazie al livello di dettaglio raggiunto sono stati fugati molti dei dubbi e delle incertezze a riguardo, confermando le previsioni sull'esito positivo del progetto.
Sono state inoltre individuate con chiarezza le risorse tecnologiche che verranno utilizzate:

- Akka: libreria per modello ad attori.
- IntelliJ: framework per la stesura del codice.
- JVM: piattaforma per il funzionamento di Scala.

Il gruppo in contemporanea si è dedicato allo studio delle nuove tecnologie raggiungendo un buon livello di conoscenza. L'insieme di queste risorse potrà garantire la realizzazione di tutte le componenti dell'architettura.

8 Tracciamento

8.1 Tracciamento componenti-requisiti

Componente	Requisiti
Actorbase.Server	R[1][N][F] e figli
Actorbase.Server.API	R[1.3][N][F] e figli
	R[1.4.1.3][D][F]
	R[1.4.4.3][N][F]
	R[1.4.4.3][N][F]
	R[1.4.5.3][N][F]
	R[1.4.4.4][N][F]
	R[1.4.6.4][D][F]
	R[1.4.6.5][D][F]
	R[1.4.7.3][N][F]
	R[1.4.7.4][N][F]
	R[1.5.1.2][D][F]
	R[1.5.2.3][N][F]
	R[1.5.2.4][N][F]
	R[1.5.3.3][N][F]
	R[1.5.3.4][N][F]
	R[1.5.4.4][O][F]
	R[1.5.4.5][O][F]
	R[1.5.5.3][N][F]
	R[1.5.5.4][N][F]
	R[1.6.1.2][D][F]
	R[1.6.2.3][N][F]
	R[1.6.2.4][N][F]
	R[1.6.3.3][N][F]
	R[1.6.3.4][N][F]
	R[1.6.4.3][N][F]
	R[1.6.4.4][N][F]
	R[1.6.6.3][N][F]
	R[1.6.6.4][N][F]
Actorbase.Server.Core	R[1.4][N][F] e figli
	R[1.5][N][F] e figli
	R[1.6][N][F] e figli
Actorbase.Server.Core.Actors	R[1.4][N][F] e figli
	R[1.5][N][F] e figli
	R[1.6][N][F] e figli

Componente	Requisiti
Actorbase.Server.Core.Actors.DataManagement	R[1.4.4.2.3][N][F]
	R[1.4.4.2.4][N][F]
	R[1.4.4.2.5][N][F]
	R[1.4.5.2.3][N][F]
	R[1.4.5.2.4][N][F]
	R[1.4.5.2.5][N][F]
	R[1.4.6.3.3][N][F]
	R[1.4.6.3.4][N][F]
	R[1.4.6.3.5][N][F]
	R[1.5.2.2.3][N][F]
	R[1.5.2.2.4][N][F]
	R[1.5.2.2.5][O][F]
	R[1.5.3.2.3][N][F]
	R[1.5.3.2.4][N][F]
	R[1.5.3.2.5][O][F]
	R[1.5.4.3.3][N][F]
	R[1.5.4.3.4][N][F]
	R[1.5.4.3.5][O][F]
	R[1.6.2.1][N][F]
	R[1.6.2.2.3][N][F]
	R[1.6.2.2.4][N][F]
	R[1.6.2.3][N][F]
	R[1.6.3.2.3][N][F]
	R[1.6.3.2.4][N][F]
	R[1.6.3.2.5][O][F]
	R[1.6.4.1][N][F]
	R[1.6.3.2.3][N][F]
	R[1.6.3.2.4][N][F]
	R[1.6.3.2.5][O][F]
	R[1.6.6.1][N][F]
	R[1.6.6.2.3][N][F]
	R[1.6.6.2.4][N][F]
	R[1.6.6.2.5][O][F]
Actorbase.Server.Core.Actors.Manager	R[1.4.4.2.6][N][F]
	R[1.4.5.2.6][N][F]
	R[1.4.6.3.6][N][F]
	R[1.5.2.2.6][O][F]
	R[1.5.3.2.6][O][F]
	R[1.5.4.3.6][O][F]
	R[1.6.3.2.6][O][F]
	R[1.6.6.2.6][O][F]

Componente	Requisiti
Actorbase.Server.Core.Actors.StoreFinder	R[1.4.1][D][F] e figli R[1.4.4.1][D][F] R[1.4.4.1.1][D][F] R[1.4.4.2.1][N][F] R[1.4.4.2.2][N][F] R[1.4.5.1][D][F] R[1.4.5.1.1][D][F] R[1.4.5.1.2][N][F] R[1.4.5.2.2][N][F] R[1.4.6.1][D][F] R[1.4.6.1.1][D][F] R[1.4.6.2][D][F] R[1.4.6.2.1][D][F] R[1.4.6.3.1][N][F] R[1.4.6.3.2][N][F] R[1.4.7.1][D][F] R[1.4.7.1.1][D][F] R[1.4.7.2][N][F] R[1.4.7.2.1][N][F] R[1.5.1.1][D][F] R[1.5.2.1][D][F] R[1.5.2.1.1][D][F] R[1.5.2.2.1][N][F] R[1.5.2.2.2][N][F] R[1.5.3.1][D][F] R[1.5.3.1.1][D][F] R[1.5.3.2.1][N][F] R[1.5.3.2.2][N][F] R[1.5.4.1][D][F] R[1.5.4.1.1][D][F] R[1.5.4.2][D][F] R[1.5.4.2.1][D][F] R[1.5.4.3.1][N][F] R[1.5.4.3.2][N][F] R[1.5.5.1][D][F] R[1.5.5.1.1][D][F] R[1.5.5.2][N][F] R[1.5.5.2.1][N][F] R[1.6.1.1][D][F] R[1.6.2.2.1][N][F] R[1.6.2.2.2][N][F] R[1.6.3.2.1][N][F] R[1.6.3.2.2][N][F] R[1.6.4.2.1][N][F] R[1.6.4.2.2][N][F] R[1.6.6.2.1][N][F] R[1.6.6.2.2][N][F]
Actorbase.Server.Core.Messages	R[1.4][N][F] e figli R[1.5][N][F] e figli R[1.6][N][F] e figli
Actorbase.Driver	R[3][N][F] e figli
Actorbase.Client	R[2][N][F] e figli

Componente	Requisiti
Actorbase.Client.Model	R[2.1.3][N][F] R[2.2.3][N][F] R[2.3.4.2][D][F] R[2.4.3][D][F] R[2.7.3][N][F] R[2.8.3][N][F] R[2.9.3][N][F] R[2.10.3][N][F] R[2.11.3][D][F] R[2.12.3][N][F] R[2.13.3][N][F] R[2.14.3][D][F] R[2.15.3][N][F] R[2.18.3][D][F] R[2.19.3][N][F] R[2.20.3][N][F] R[2.21.3][N][F] R[2.23.3][N][F]
Actorbase.Client.View	R[2.1.5][N][F] R[2.1.6][N][F] R[2.2.4][N][F] R[2.3.2.2][D][F] R[2.3.4.3][D][F] R[2.4.4][D][F] R[2.7.4][N][F] R[2.7.5][N][F] R[2.8.4][N][F] R[2.8.5][N][F] e figli R[2.9.4][N][F] R[2.9.5][N][F] e figli R[2.10.4][N][F] R[2.10.5][N][F] e figli R[2.11.4][D][F] R[2.12.4][N][F] R[2.12.5][N][F] e figli R[2.13.4][N][F] R[2.13.5][N][F] R[2.14.4][D][F] R[2.14.5][D][F] e figli R[2.15.4][N][F] R[2.15.5][N][F] con figli R[2.18.4][D][F] R[2.19.4][N][F] R[2.20.4][N][F] R[2.20.5][N][F] e figli R[2.21.4][N][F] R[2.21.5][N][F] R[2.23.4][N][F] R[2.23.5][N][F] e figli

Componente	Requisiti
Actorbase.Client.Controller	R[2.1.1][N][F]
	R[2.1.2][N][F] e figli
	R[2.2.1][N][F]
	R[2.2.2][N][F]
	R[2.3.1][D][F]
	R[2.3.2.1][D][F]
	R[2.3.3][D][F]
	R[2.3.4.1][D][F]
	R[2.4.1][D][F]
	R[2.4.2][D][F]
	R[2.7.1][N][F]
	R[2.7.2][N][F] e figli
	R[2.8.1][N][F]
	R[2.8.2][N][F] e figli
	R[2.9.1][N][F]
	R[2.9.2][N][F] e figli
	R[2.10.1][N][F]
	R[2.10.2][N][F] e figli
	R[2.11.1][D][F]
	R[2.11.2][D][F]
	R[2.12.1][N][F]
	R[2.12.2][N][F] e figli
	R[2.13.1][N][F]
	R[2.13.2][N][F] e figli
	R[2.14.1][D][F]
	R[2.14.2][D][F] e figli
	R[2.15.1][N][F]
	R[2.15.2][N][F] e figli
	R[2.18.1][D][F]
	R[2.18.2][D][F]
	R[2.19.1][N][F]
	R[2.19.2][N][F] e figli
	R[2.20.1][N][F]
	R[2.20.2][N][F] e figli
	R[2.21.1][N][F]
	R[2.21.2][N][F] e figli
	R[2.23.1][N][F]
	R[2.23.2][N][F] e figli

Tabella 2: Componenti-Requisiti

8.2 Tracciamento requisiti-componenti

Requisito	Componenti
R[1][N][F] e figli	Actorbase.Server
R[1.3][N][F] e figli	Actorbase.Server.API
R[1.4][N][F] e figli	Actorbase.Server.Core
	Actorbase.Server.Core.Actors
	Actorbase.Server.Core.Messages
R[1.4.1][D][F] e figli	Actorbase.Server.Core.Actors.StoreFinder
R[1.4.1.3][D][F]	Actorbase.Server.API
R[1.4.4.1][D][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.4.4.1.1][D][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.4.4.2.1][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.4.4.2.2][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.4.4.2.3][N][F]	Actorbase.Server.Core.Actors.DataManagement

Requisito	Componenti
R[1.4.4.2.4][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.4.4.2.5][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.4.4.2.6][N][F]	Actorbase.Server.Core.Actors.Manager
R[1.4.4.3][N][F]	Actorbase.Server.API
R[1.4.4.4][N][F]	Actorbase.Server.API
R[1.4.5.1][D][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.4.5.1.1][D][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.4.5.1.2][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.4.5.2.2][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.4.5.2.3][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.4.5.2.4][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.4.5.2.5][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.4.5.2.6][N][F]	Actorbase.Server.Core.Actors.Manager
R[1.4.5.3][N][F]	Actorbase.Server.API
R[1.4.6.1][D][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.4.6.1.1][D][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.4.6.2][D][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.4.6.2.1][D][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.4.6.3.1][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.4.6.3.2][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.4.6.3.3][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.4.6.3.4][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.4.6.3.5][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.4.6.3.6][N][F]	Actorbase.Server.Core.Actors.Manager
R[1.4.6.4][D][F]	Actorbase.Server.API
R[1.4.6.5][D][F]	Actorbase.Server.API
R[1.4.7.1][D][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.4.7.1.1][D][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.4.7.2][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.4.7.2.1][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.4.7.3][N][F]	Actorbase.Server.API
R[1.4.7.4][N][F]	Actorbase.Server.API
R[1.5][N][F] e figli	Actorbase.Server.Core Actorbase.Server.Core.Actors Actorbase.Server.Core.Messages
R[1.5.1.1][D][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.5.1.2][D][F]	Actorbase.Server.API
R[1.5.2.1][D][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.5.2.1.1][D][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.5.2.2.1][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.5.2.2.2][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.5.2.2.3][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.5.2.2.4][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.5.2.2.5][O][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.5.2.2.6][O][F]	Actorbase.Server.Core.Actors.Manager
R[1.5.2.3][N][F]	Actorbase.Server.API
R[1.5.2.4][N][F]	Actorbase.Server.API
R[1.5.3.1][D][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.5.3.1.1][D][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.5.3.2.1][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.5.3.2.2][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.5.3.2.3][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.5.3.2.4][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.5.3.2.5][O][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.5.3.2.6][O][F]	Actorbase.Server.Core.Actors.Manager

Requisito	Componenti
R[1.5.3.3][N][F]	Actorbase.Server.API
R[1.5.3.4][N][F]	Actorbase.Server.API
R[1.5.4.1][D][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.5.4.1.1][D][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.5.4.2][D][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.5.4.2.1][D][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.5.4.3.1][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.5.4.3.2][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.5.4.3.3][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.5.4.3.4][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.5.4.3.5][O][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.5.4.3.6][O][F]	Actorbase.Server.Core.Actors.Manager
R[1.5.4.4][O][F]	Actorbase.Server.API
R[1.5.4.5][O][F]	Actorbase.Server.API
R[1.5.5.1][D][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.5.5.1.1][D][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.5.5.2][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.5.5.2.1][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.5.5.3][N][F]	Actorbase.Server.API
R[1.5.5.4][N][F]	Actorbase.Server.API
R[1.6][N][F] e figli	Actorbase.Server.Core Actorbase.Server.Core.Actors Actorbase.Server.Core.Messages
R[1.6.1.1][D][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.6.1.2][D][F]	Actorbase.Server.API
R[1.6.2.1][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.6.2.2.1][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.6.2.2.2][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.6.2.2.3][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.6.2.2.4][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.6.2.3][N][F]	Actorbase.Server.API Actorbase.Server.Core.Actors.DataManagement
R[1.6.2.4][N][F]	Actorbase.Server.API
R[1.6.3.2.1][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.6.3.2.2][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.6.3.2.3][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.6.3.2.4][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.6.3.2.5][O][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.6.3.2.6][O][F]	Actorbase.Server.Core.Actors.Manager
R[1.6.3.3][N][F]	Actorbase.Server.API
R[1.6.3.4][N][F]	Actorbase.Server.API
R[1.6.4.1][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.6.4.2.1][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.6.4.2.2][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.6.4.3][N][F]	Actorbase.Server.API
R[1.6.4.4][N][F]	Actorbase.Server.API
R[1.6.6.1][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.6.6.2.1][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.6.6.2.2][N][F]	Actorbase.Server.Core.Actors.StoreFinder
R[1.6.6.2.3][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.6.6.2.4][N][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.6.6.2.5][O][F]	Actorbase.Server.Core.Actors.DataManagement
R[1.6.6.2.6][O][F]	Actorbase.Server.Core.Actors.Manager
R[1.6.6.3][N][F]	Actorbase.Server.API
R[1.6.6.4][N][F]	Actorbase.Server.API

Requisito	Componenti
R[2][N][F] e figli	Actorbase.Client
R[2.1.1]	
R[2.1.2][N][F] e	
R[2.1.3][N][F]	Actorbase.Client.Model
R[2.1.5][N][F]	Actorbase.Client.View Actorbase.Client.Controller
R[2.1.6][N][F]	Actorbase.Client.View Actorbase.Client.Controller
R[2.2.1]	Actorbase.Client.Controller
R[2.2.2]	Actorbase.Client.Controller
R[2.2.3][N][F]	Actorbase.Client.Model
R[2.2.4][N][F]	Actorbase.Client.View Actorbase.Client.Controller
R[2.3.1]	Actorbase.Client.Controller
R[2.3.2.1]	Actorbase.Client.Controller
R[2.3.2.2][D][F]	Actorbase.Client.View Actorbase.Client.Controller
R[2.3.3]	Actorbase.Client.Controller
R[2.3.4.1]	Actorbase.Client.Controller
R[2.3.4.2][D][F]	Actorbase.Client.Model
R[2.3.4.3][D][F]	Actorbase.Client.View Actorbase.Client.Controller
R[2.4.1]	Actorbase.Client.Controller
R[2.4.2]	Actorbase.Client.Controller
R[2.4.3][D][F]	Actorbase.Client.Model
R[2.4.4][D][F]	Actorbase.Client.View Actorbase.Client.Controller
R[2.7.1]	Actorbase.Client.Controller
R[2.7.2][N][F] e figli	Actorbase.Client.Controller
R[2.7.3][N][F]	Actorbase.Client.Model
R[2.7.4][N][F]	Actorbase.Client.View Actorbase.Client.Controller
R[2.7.5][N][F]	Actorbase.Client.View Actorbase.Client.Controller
R[2.8.1]	Actorbase.Client.Controller
R[2.8.2][N][F] e figli	Actorbase.Client.Controller
R[2.8.3][N][F]	Actorbase.Client.Model
R[2.8.4][N][F]	Actorbase.Client.View Actorbase.Client.Controller
R[2.8.5][N][F] e figli	Actorbase.Client.View Actorbase.Client.Controller
R[2.9.1]	Actorbase.Client.Controller
R[2.9.2][N][F] e figli	Actorbase.Client.Controller
R[2.9.3][N][F]	Actorbase.Client.Model
R[2.9.4][N][F]	Actorbase.Client.View Actorbase.Client.Controller
R[2.9.5][N][F] e figli	Actorbase.Client.View Actorbase.Client.Controller
R[2.10.1]	Actorbase.Client.Controller
R[2.10.2][N][F] e figli	Actorbase.Client.Controller
R[2.10.3][N][F]	Actorbase.Client.Model
R[2.10.4][N][F]	Actorbase.Client.View Actorbase.Client.Controller
R[2.10.5][N][F] e figli	Actorbase.Client.View Actorbase.Client.Controller

Requisito	Componenti
R[2.11.1]	Actorbase.Client.Controller
R[2.11.2]	Actorbase.Client.Controller
R[2.11.3][D][F]	Actorbase.Client.Model
R[2.11.4][D][F]	Actorbase.Client.View Actorbase.Client.Controller
R[2.12.1]	Actorbase.Client.Controller
R[2.12.2][N][F] e figli	Actorbase.Client.Controller
R[2.12.3][N][F]	Actorbase.Client.Model
R[2.12.4][N][F]	Actorbase.Client.View Actorbase.Client.Controller
R[2.12.5][N][F] e figli	Actorbase.Client.View Actorbase.Client.Controller
R[2.13.1]	Actorbase.Client.Controller
R[2.13.2][N][F] e figli	Actorbase.Client.Controller
R[2.13.3][N][F]	Actorbase.Client.Model
R[2.13.4][N][F]	Actorbase.Client.View Actorbase.Client.Controller
R[2.13.5][N][F]	Actorbase.Client.View Actorbase.Client.Controller
R[2.14.1]	Actorbase.Client.Controller
R[2.14.2][D][F] e figli	Actorbase.Client.Controller
R[2.14.3][D][F]	Actorbase.Client.Model
R[2.14.4][D][F]	Actorbase.Client.View Actorbase.Client.Controller
R[2.14.5][D][F] e figli	Actorbase.Client.View Actorbase.Client.Controller
R[2.15.1]	Actorbase.Client.Controller
R[2.15.2][N][F] e figli	Actorbase.Client.Controller
R[2.15.3][N][F]	Actorbase.Client.Model
R[2.15.4][N][F]	Actorbase.Client.View Actorbase.Client.Controller
R[2.15.5][N][F] con figli	Actorbase.Client.View Actorbase.Client.Controller
R[2.18.1]	Actorbase.Client.Controller
R[2.18.2]	Actorbase.Client.Controller
R[2.18.3][D][F]	Actorbase.Client.Model
R[2.18.4][D][F]	Actorbase.Client.View Actorbase.Client.Controller
R[2.19.1]	Actorbase.Client.Controller
R[2.19.2][N][F] e figli	Actorbase.Client.Controller
R[2.19.3][N][F]	Actorbase.Client.Model
R[2.19.4][N][F]	Actorbase.Client.View Actorbase.Client.Controller
R[2.20.1]	Actorbase.Client.Controller
R[2.20.2][N][F] e figli	Actorbase.Client.Controller
R[2.20.3][N][F]	Actorbase.Client.Model
R[2.20.4][N][F]	Actorbase.Client.View Actorbase.Client.Controller
R[2.20.5][N][F] e figli	Actorbase.Client.View Actorbase.Client.Controller
R[2.21.1]	Actorbase.Client.Controller
R[2.21.2][N][F] e figli	Actorbase.Client.Controller
R[2.21.3][N][F]	Actorbase.Client.Model
R[2.21.4][N][F]	Actorbase.Client.View Actorbase.Client.Controller

Requisito	Componenti
R[2.21.5][N][F]	Actorbase.Client.View Actorbase.Client.Controller
R[2.23.1]	Actorbase.Client.Controller
R[2.23.2][N][F] e figli	Actorbase.Client.Controller
R[2.23.3][N][F]	Actorbase.Client.Model
R[2.23.4][N][F]	Actorbase.Client.View Actorbase.Client.Controller
R[2.23.5][N][F] e figli	Actorbase.Client.View Actorbase.Client.Controller
R[3][N][F] e figli	Actorbase.Driver

Tabella 3: Requisito-componente

9 Appendice

9.1 Descrizione Design Pattern

Segue, per ogni Design Pattern utilizzato, la descrizione dello scopo, motivazione e applicabilità.

9.1.1 Event-driven

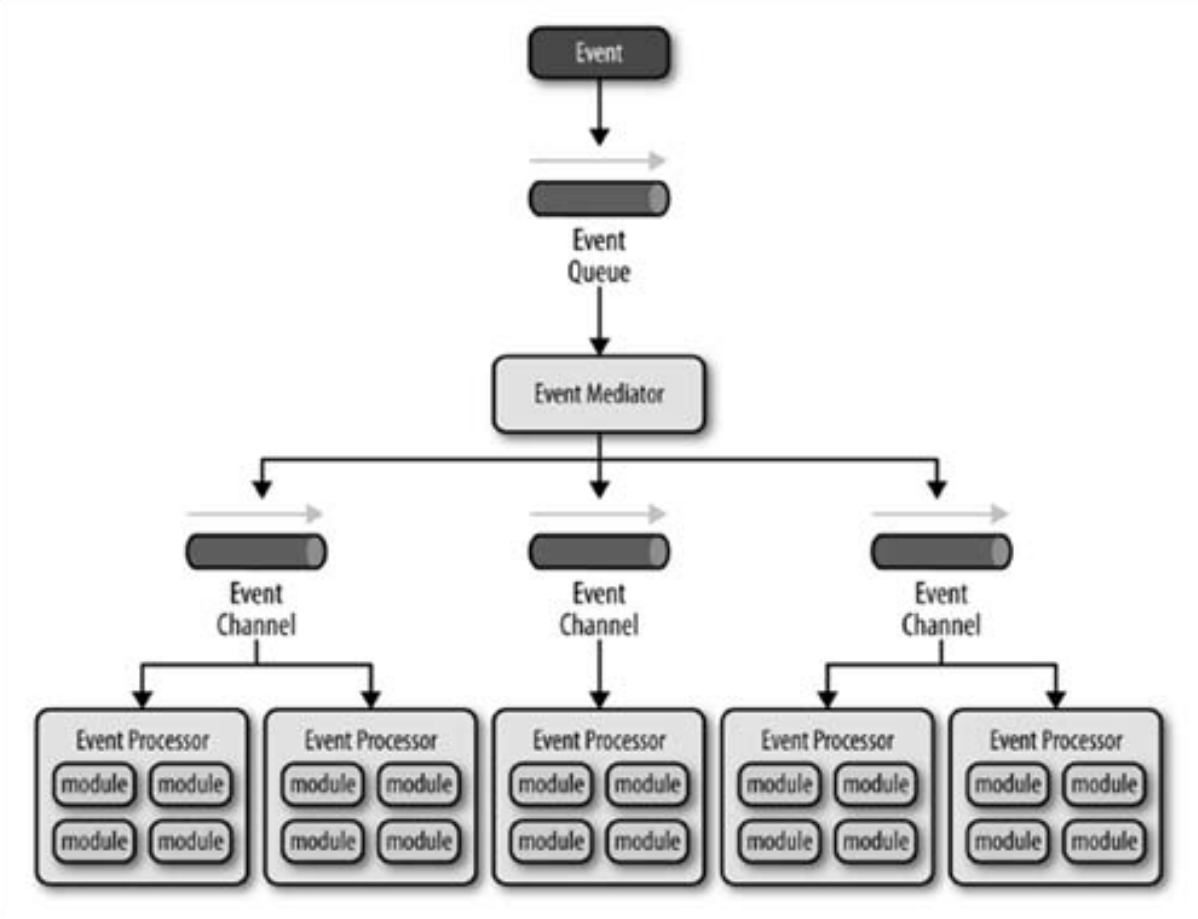


Figura 31: Diagramma del Design Pattern Event-driven

- **Scopo:** Produrre applicazioni molto scalabili e processare eventi asincroni disaccoppiati.
- **Motivazione:** Gestire le richieste che vengono volte all' applicativo tramite eventi processati in modo asincrono.
- **Applicabilità:** Gestione di eventi attraverso l'utilizzo di un mediatore e elaboratori di eventi

9.1.2 Singleton

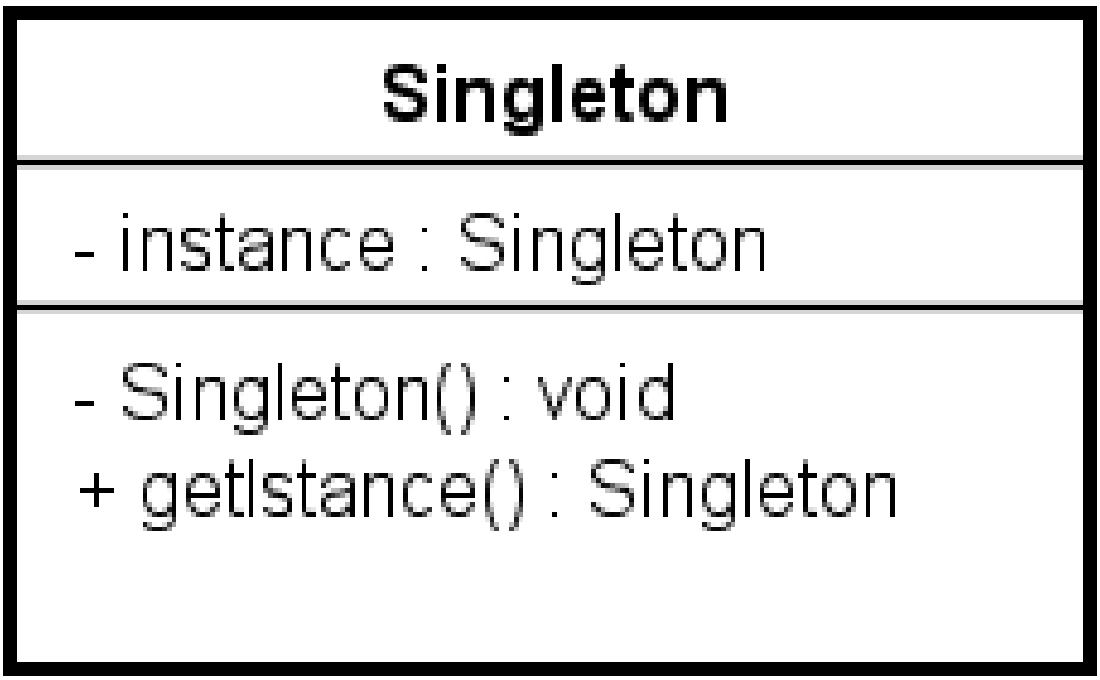


Figura 32: Diagramma del Design Pattern Command

- **Scopo:** Assicurare che una classe abbia una sola istanza con un unico punto di accesso globale.
- **Motivazione:** È necessario assicurare che esista una sola istanza di alcune classi. Una classe Singleton ha la responsabilità sulle proprie istanze, in modo che nessuna altra istanza possa essere creata, e fornisce un punto di accesso unico.
- **Applicabilità:**
  - Deve esistere una ed una sola istanza di una classe in tutta l'applicazione, accessibile dai client in modo noto.
  - L'istanza deve essere estendibile con ereditarietà, consentendo ai client di non modificare il proprio codice.

Elenco delle figure

Elenco delle tabelle