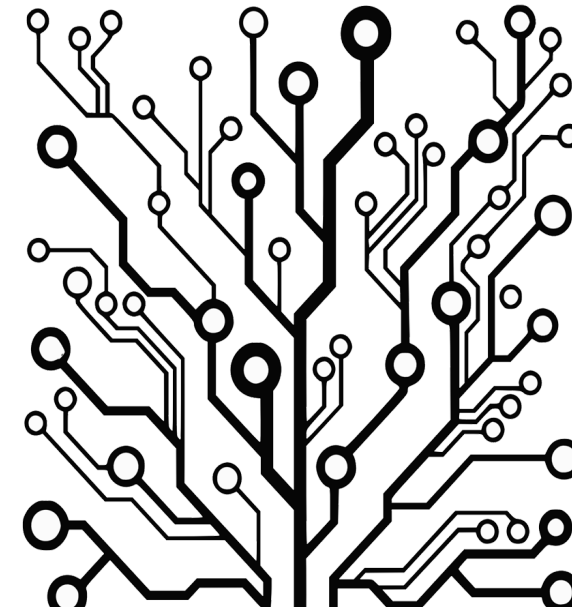


Corso di Laurea Magistrale in
Ingegneria Informatica

Elaborato del corso di *Applicazioni Telematiche*

Secure Messaging

Documentazione



Parricelli Francesco M63/720
Strofaldi Aldo M63/728
Pirozzi Luca M63/744

Anno Accademico 2017/2018

Indice

	Indice	2
1	DOCUMENTAZIONE	3
1.1	Documentazione Server Tomcat	3
1.1.1	Diagramma delle Classi	3
1.1.1.1	Package: access control	3
1.1.1.2	Package: authentication	4
1.1.1.3	Package: authorization	6
1.1.1.4	Package: bcrypt	6
1.1.1.5	Package: certificates	6
1.1.1.6	Package: DAO	7
1.1.1.7	Package: entity	8
1.1.1.8	Package: exception	8
1.1.1.9	Package: keystore	10
1.1.1.10	Package: keystore.rsaKeystore	10
1.1.1.11	Package: registration	10
1.1.1.12	Package: twosteps	10
1.1.1.13	Package: utility	12
1.1.1.14	Package: utility.database	12
1.1.1.15	Package: utility.mail	12
1.1.1.16	Package: utility.network	12
1.1.1.17	Package: webfilter	13
1.1.1.18	Package: websocket	13
1.1.1.19	Package: xml.registration	13

1 Documentazione

1.1 Documentazione Server Tomcat

In questa sezione vengono mostrati i diagrammi UML relativamente alla realizzazione del Server Tomcat, su cui sono installate le Servlet per la gestione delle politiche di login e registrazione, nonché l'interfacciamento con la base di dati e con il mail server.

1.1.1 Diagramma delle Classi

Di seguito viene riportato il diagramma delle classi di Secure Messaging: piuttosto che limitarci ad allegare un solo diagramma, troppo complesso e poco informativo, si riportano le classi per package e relazioni intra ed inter-package; in questo modo abbiamo suddiviso il diagramma delle classi originali in porzioni sufficientemente piccole da poter essere analizzate con profitto ma sufficientemente grandi da essere significative, che nell'insieme permette ad un programmatore la giusta documentazione per individuare struttura e dipendenza delle classi con l'adeguata granularità di informazioni. Per ragioni di semplicità sono state omesse le relazioni al package delle eccezioni, da cui praticamente tutti i package dipendono.

1.1.1.1 Package: **access control**

Contiene la classe che si occupa delle operazioni per prevenire o individuare le intrusioni in un account, aggiornare i conteggi dei login falliti e bloccare i tentativi di accesso.

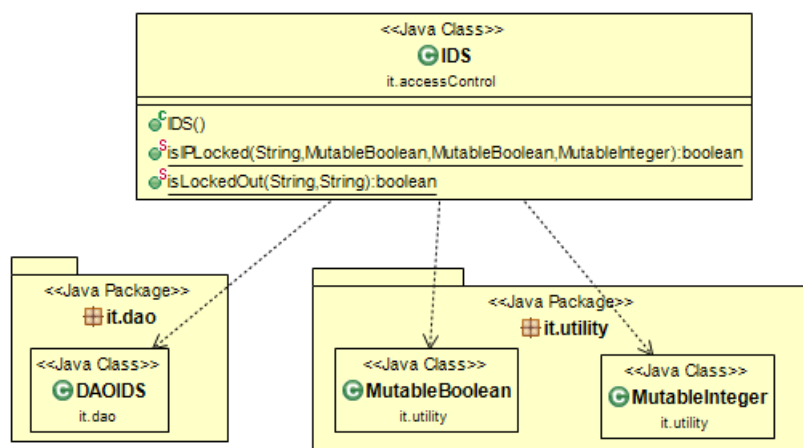


Figura 1 – Package Access Control

1.1.1.2 Package: **authentication**

Appartengono a questo package la servlet di autenticazione e la relativa logica (disaccoppiata dalla servlet per il principio della separazione degli interessi). Il package è visualizzato in figura 2.

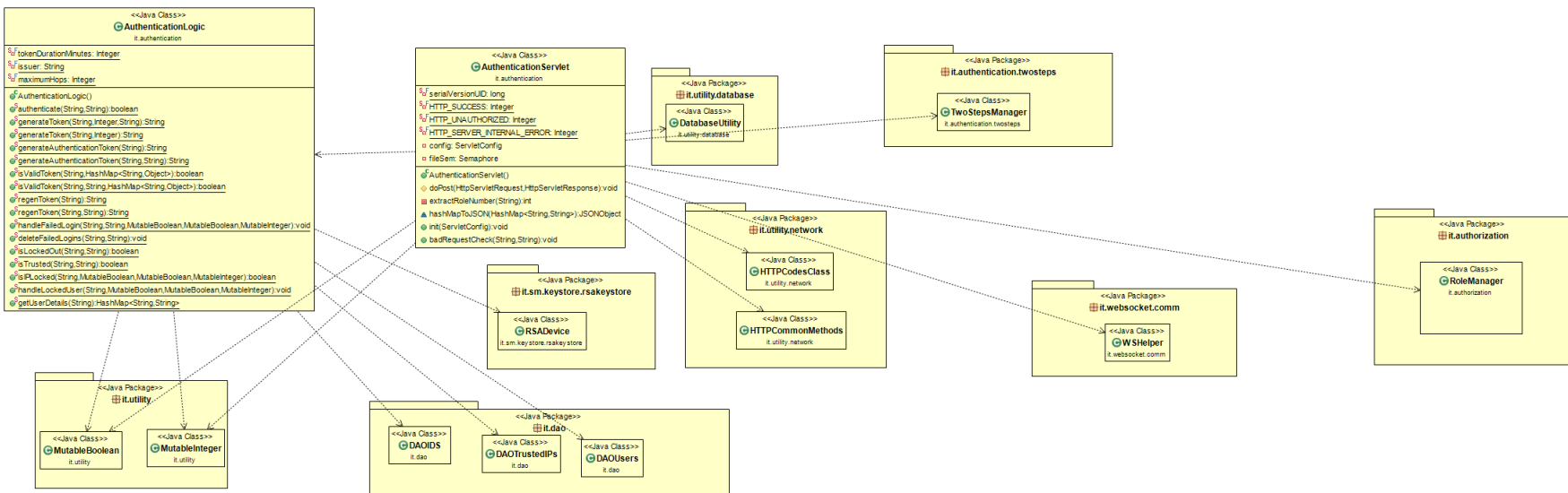


Figura 2 – Package Authentication

1.1.1.3 Package: **authorization**

Le classi di questo package si occupano di determinare se l'utente è autorizzato ad effettuare una determinata operazione. Il package è visualizzato in figura 3.

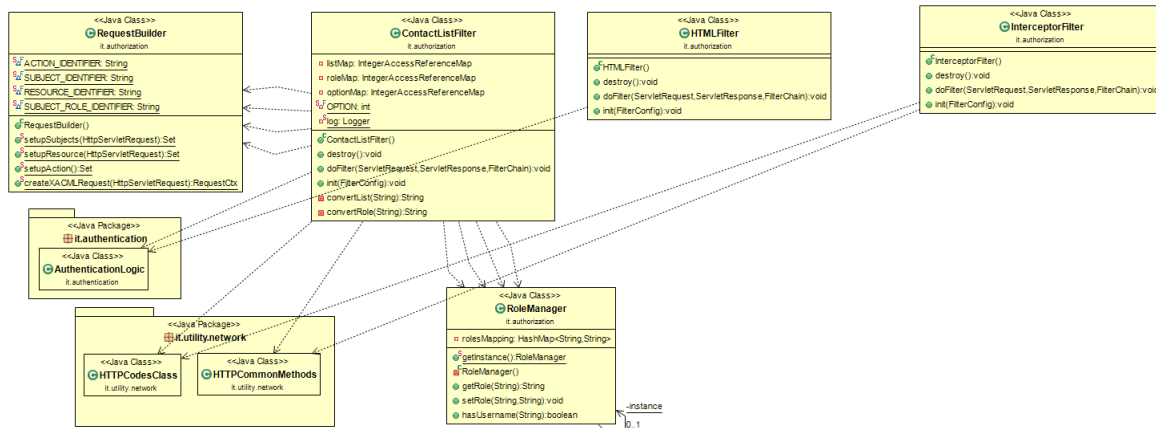


Figura 3 – Package Authorization

1.1.1.4 Package: **bcrypt**

Package importato da una libreria OTS che implementa hashing/salting delle password con algoritmo bcrypt (utilizzato, fra gli altri, da OpenBSD). Il package è visualizzato in figura 4.

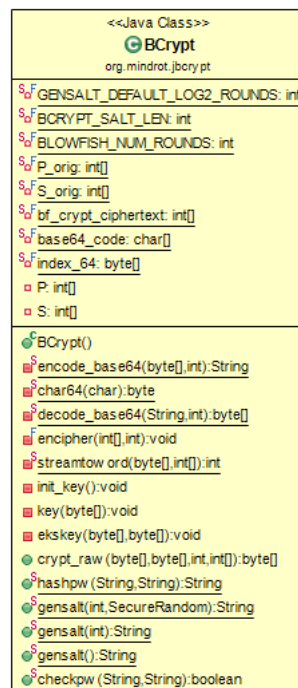


Figura 4 – Package Bcrypt

1.1.1.5 Package: **certificates**

Contiene la servlet a cui rivolgersi per recuperare i certificati associati ad utenti, tecnici ed amministratori. Il package è visualizzato in figura 5.

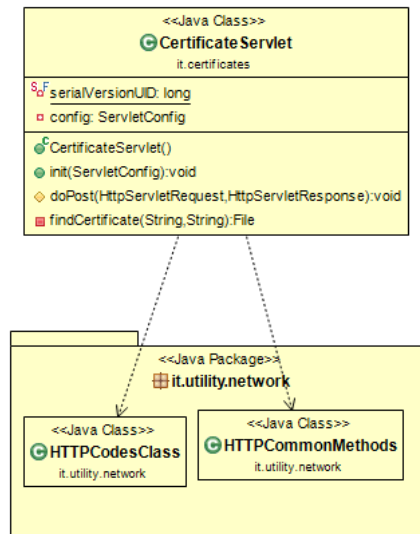


Figura 5 – Package Certificates

1.1.1.6 Package: DAO

Questo package rappresenta il ponte tra i dati del database e l'applicazione. Sono le classi a cui le altre si rivolgono per operare su dati che altrimenti sarebbero da prelevare direttamente dal database.

Il package è visualizzato in figura 6.

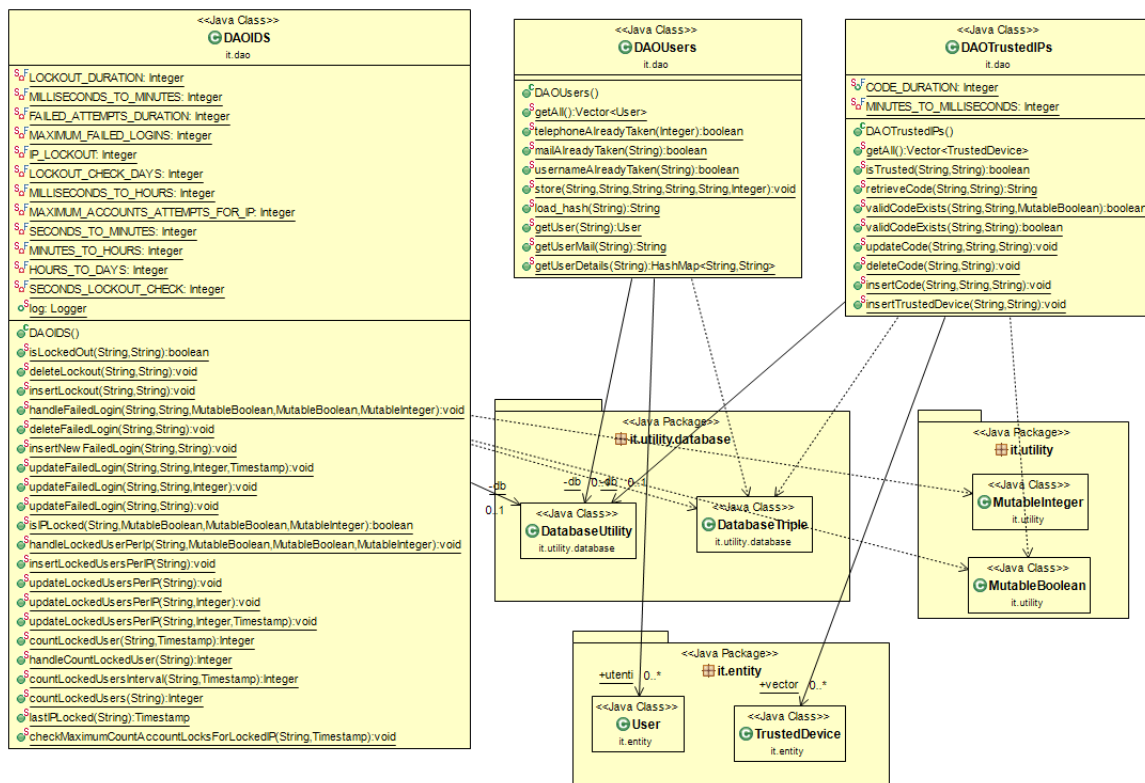


Figura 6 – Package DAO

1.1.1.7 Package: **entity**

Costituiscono una rappresentazione Object-oriented che effettua il mapping rispetto alla rappresentazione del db.

Il package è visualizzato in figura 7.

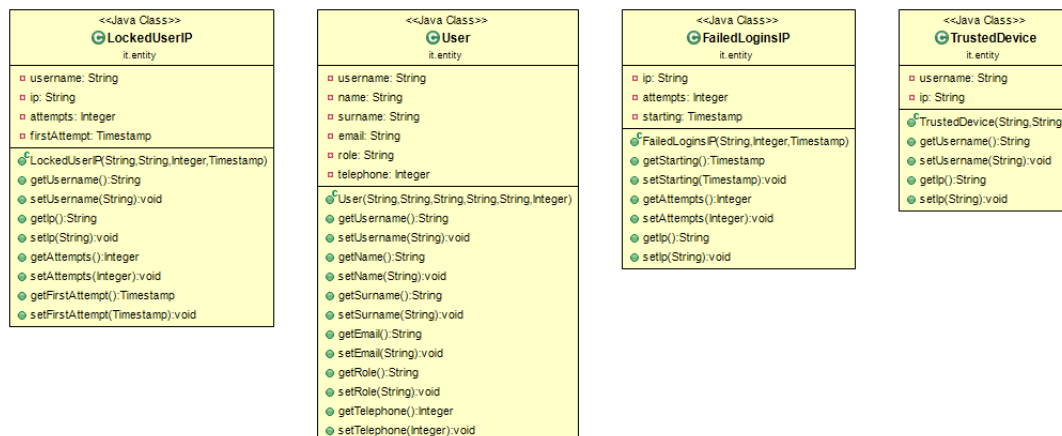


Figura 7 – Package Entity

1.1.1.8 Package: **exception**

Raccoglie tutte le eccezioni che possono essere lanciate. A sua volta è diviso in sotto-package che non vengono rappresentati per semplicità.

Il package è visualizzato in figura 8, a pagina successiva.

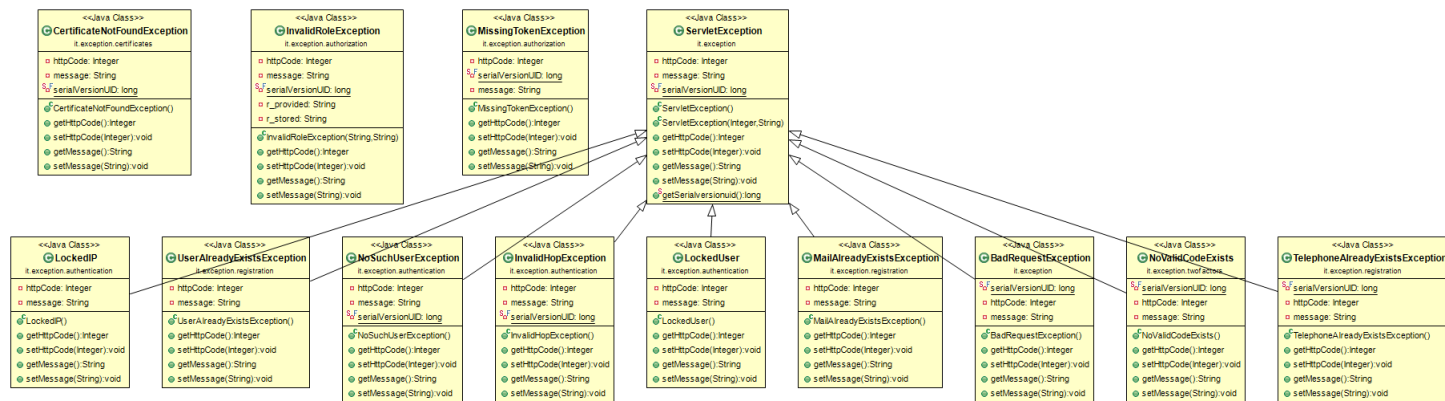


Figura 8 – Package Exception

1.1.1.9 Package: **keystore**

Contiene la semplice interfaccia che ci aspetteremmo da un'entità crittografica: la capacità di criptare e decriptare bytes. Implementeranno l'interfaccia tutte quelle classi che si occuperanno di crittografare dati (implementando gli specifici algoritmi di cui MyKeystore è indipendente).

Il package è visualizzato in figura 9.

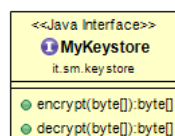


Figura 9 – Package Keystore

1.1.1.10 Package: **keystore.rsaKeystore**

Package innestato nel precedente che rappresenta una crittografia specifica, in questo caso di tipo RSA. Simula le caratteristiche di una smart-card (di cui avremmo bisogno per una crittografia sicura) in software, e usa le chiavi private e pubblica per codificare e decodificare: la privata viene usata per codificare i dati da mandare al client, la pubblica quelli che solo il server può leggere, come la chiave AES utilizzata per codificare i parametri di accesso al db e al mail server.

Il package è visualizzato in figura 10.

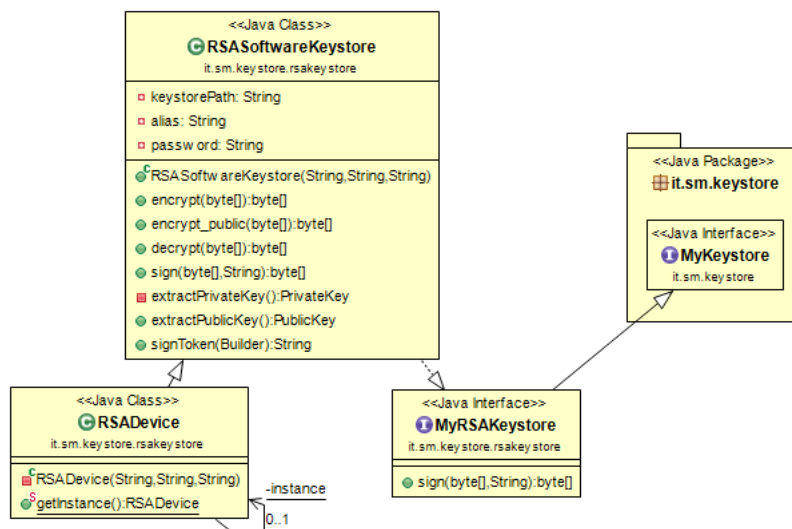


Figura 10 – Package keystore.rsaKeystore

1.1.1.11 Package: **registration**

Appartengono a questo package la servlet e la logica di registrazione.

Il package è visualizzato in figura 11.

1.1.1.12 Package: **twosteps**

Di questo package fanno parte le classi che si occupano dell'autenticazione in due passi; la classe TwoStepsServlet rappresenta la servlet a cui rivolgersi per l'invio del codice, ed effettua chiamate a TwoStepsLogic per ciò che concerne la logica delle operazioni, mentre TwoStepsManager si occupa dell'invio della mail nel caso in cui fosse richiesta.

Il package è visualizzato in figura 12.

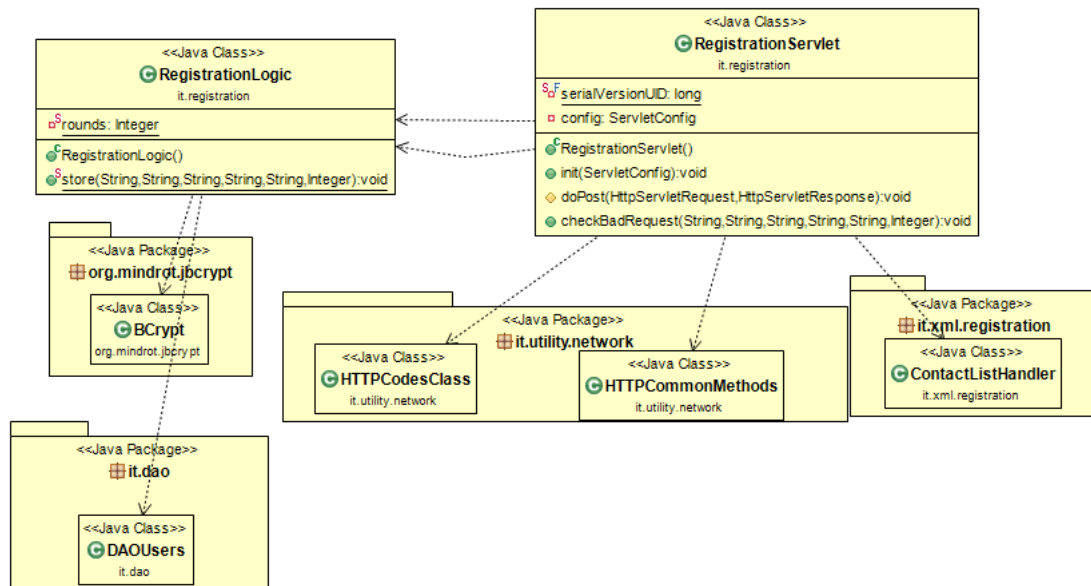


Figura 11 – Package Registration

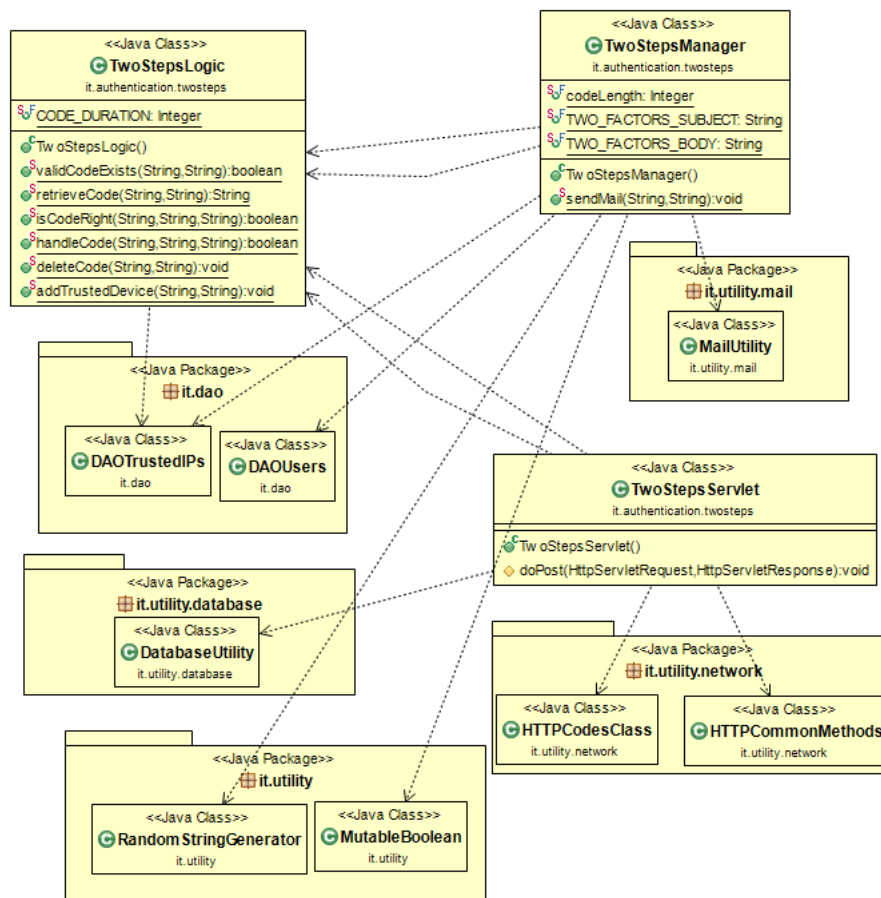


Figura 12 – Package Twosteps

1.1.1.13 Package: **utility**

Contiene diverse classi di utilità, utilizzate per risolvere in modo semplice alcuni semplici compiti, come la generazione del codice per l'autenticazione a due step (RandomStringGenerator).

Il package è visualizzato in figura 13.

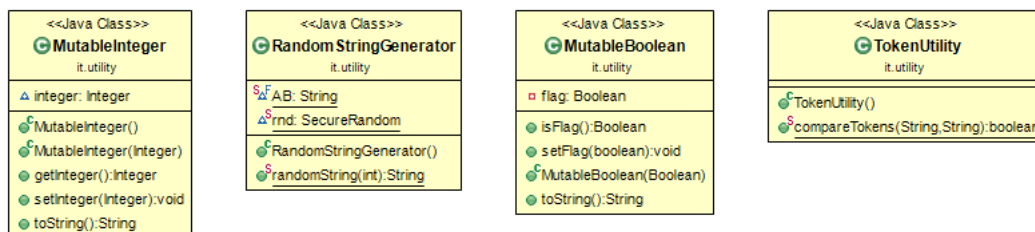


Figura 13 – Package Utility

1.1.1.14 Package: **utility.database**

A questo package appartengono delle classi di utilità create per rendere più semplice alcune operazioni di routine sul database.

Il package è visualizzato in figura 14.

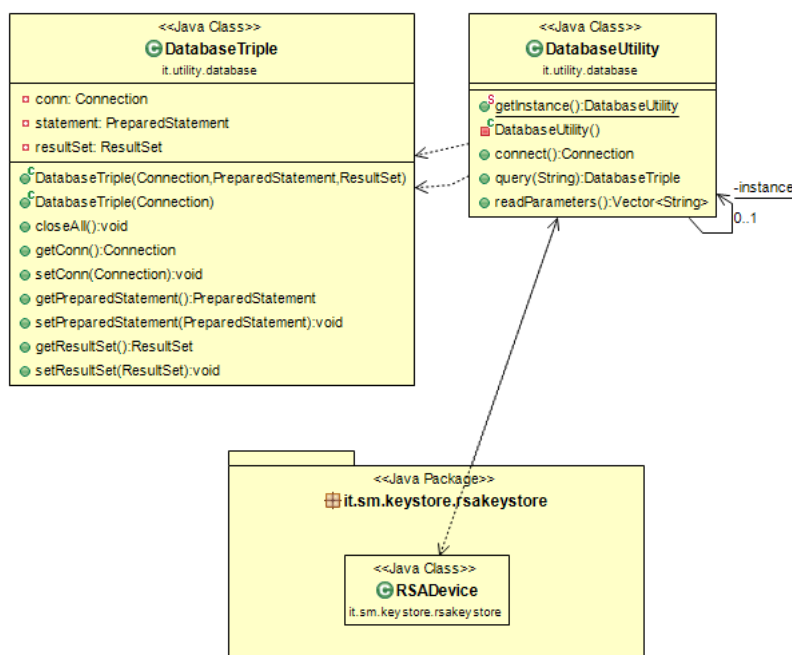


Figura 14 – Package utility.database

1.1.1.15 Package: **utility.mail**

Contiene la classe per l'interfacciamento al mail server.

Il package è visualizzato in figura 15.

1.1.1.16 Package: **utility.network**

Contiene la classe per l'interfacciamento al mail server.

Il package è visualizzato in figura 16.

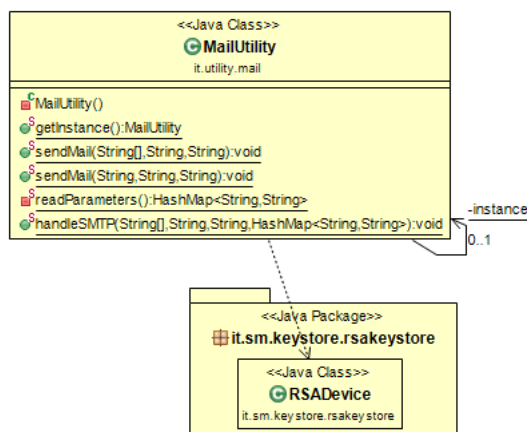


Figura 15 – Package utility.mail

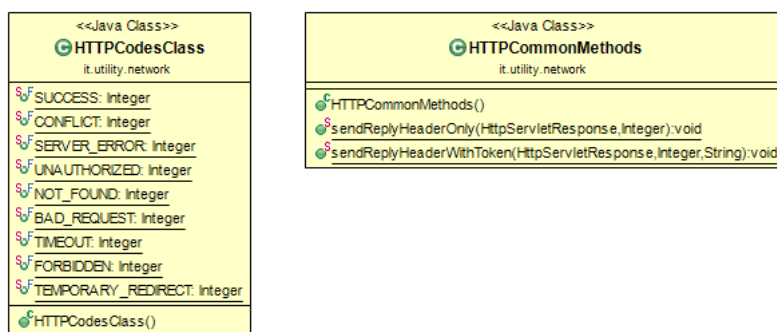


Figura 16 – Package utility.network

1.1.1.17 Package: **webfilter**

Le classi di questo package risolvono diversi problemi come il Cross Origin Resource Sharing. Il package è visualizzato in figura 17.

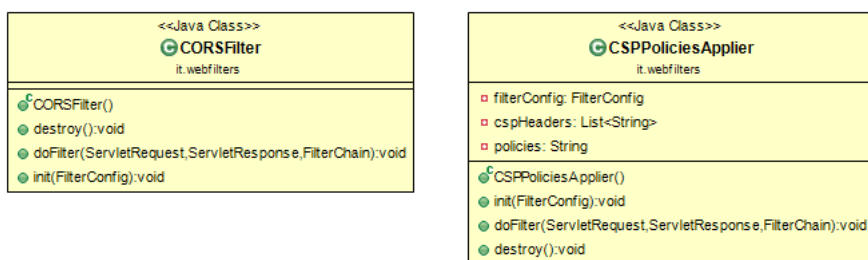


Figura 17 – Package webfilter

1.1.1.18 Package: **websocket**

Questo package contiene le classi per la gestione delle Web Socket.

Il package è visualizzato in figura 18.

1.1.1.19 Package: **xml.registration**

Il package è visualizzato in figura 19.

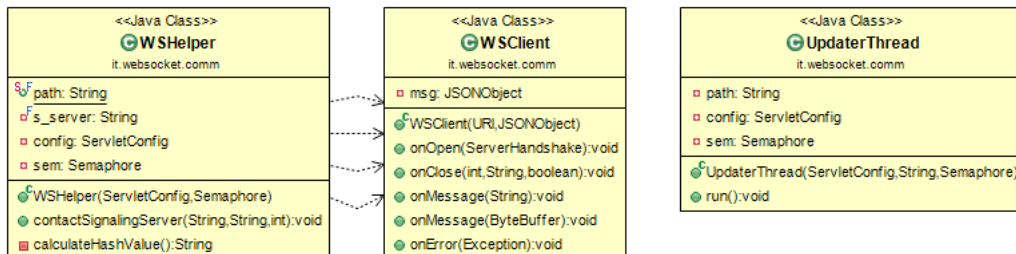


Figura 18 – Package websocket

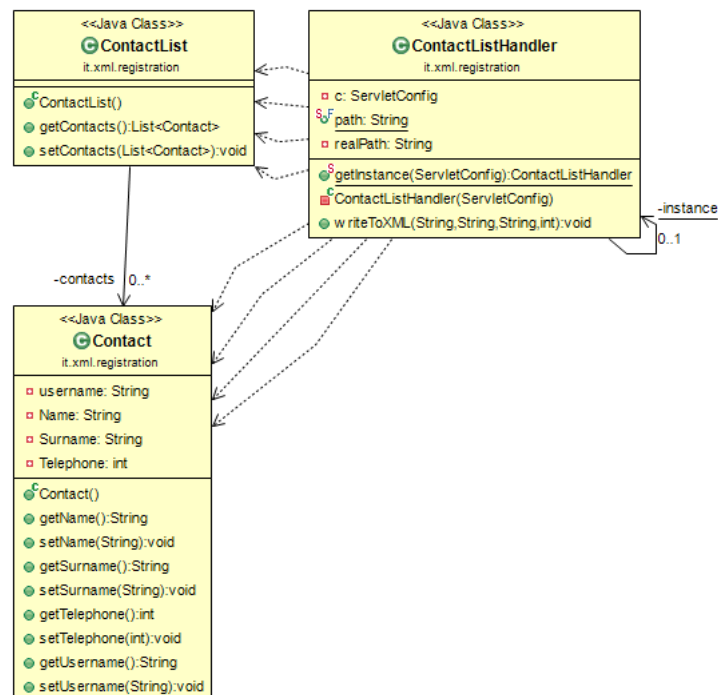


Figura 19 – Package websocket