Tesina Programmazione Web

Giovanni Marchetto

Informazioni base

- sorgente - per l'utilizzo non si esce dal perimetro della configurazione vista a lezione
- versione Live - username: admin@prog.com, password: 26!@W&FQLKmb

Si è cercato di utilizzare solo gli strumenti visti a lezione, o che fossero quanto più vicini ad essi, e la loro documentazione.

Back-end

Si è creata l'applicazione web a partire da un artefatto (proposto da IntelliJ), modificando opportunamente il pom e specificando l'utilizzo runtime di Java 8. Si è dunque utilizzato GAE come servlet container.

Si sono implementati dei RESTful Web Services, seguendo il paradigma HATEOAS, utilizzando la tecnologia Java JAX-RS nella declinazione di Jersey. Per la documentazione si è incluso Swagger. La scelta di implementare una REST API è stata fatta per evitare i messaggi xml che occupano molte risorse.

La sicurezza è stata realizzata in modo programmatico. Si è utilizzata la metodologia basata sui JWT, implementata utilizzando la libreria di auth0, per la sua semplicità ed eleganza nel fornire un'informazione autentica ed integra. Tuttavia, aggiunge un carico lato server.

- Autenticazione: la verifica del JWT è stata realizzata tramite filtri.
- Autorizzazione: è stata realizzata in alcuni casi nei filtri, in altri all'interno dei web services (principalmente per evitare duplicazione).

Per gestire la gerarchia dei filtri si è ricorsi alla mappatura via web.xml, siccome tramite le annotation non era possibile.

Si è utilizzato il GAE Datastore come DBMS e Objectify come ORM di collegamento (framework), poiché sono stati considerati i metodi più semplici e coerenti per realizzare la gestione dei dati.

- Le password sono state salvate con la metodologia dell'hash&salt.
- I file sono trasmessi in base64 e salvati come byte di array e per il download sono inviati come octet stream. I loghi vengono sia tramessi che salvati in base64. L'utilizzo della codifica in base64 risulta semplice ma dispendiosa in termini di dimensioni dei messaggi.

Per inviare le e-mail si è utilizzata la Mail Api di Google poiché è stata ritenuta l'implementazione più semplice e coerente con il nostro scopo.

Front-end

È stato utilizzato il framework Vue.js 2.0 con l'appoggio dell'IDE Visual Studio Code e di Vue UI.

Si è utilizzata la dipendenza di Bootstrap per Vue per gestire in modo semplice e minimale la parte di rappresentazione, anche se la dipendenza introduce dei comandi ad hoc e quindi ne limitano la portabilità.

Si è utilizzato Vue Router per la gestione della navigazione in base al ruolo dell'utente. Questa scelta ha permesso una distribuzione selettiva dei moduli e quindi un risparmio sulle richieste.

Per interfacciarsi con il back-end si è utilizzato Axios, poiché è stato reputato come il sistema più semplice per le scambio di informazioni. In particolare, si è utilizzato il localStorage del browser per il salvataggio del JWT poiché è stata considerata la miglior alternativa a Vuex.

REST client essenziale

È stato realizzato un client che permette di fare login e caricare un file eseguendo il Main.java. Settando la variabile "GUI=true" si accederà ad un'interfaccia minimale per l'inserimento dei dati, altrimenti si utilizzeranno i parametri preinseriti.

Referenze

Tutte le parti di codice che sono state ricavate dal web (e opportunamente adattate) sono state inserite nel punto di utilizzo nel progetto. Le motivazioni di questa scelta sono due: per motivi di sicurezza (password, token) e per problemi non trattati a lezione di cui non si è riusciti a trovare una soluzione.