

DESCRIZIONE DEI SERVER

CLIENT - Chiamante.

GATEWAY - Spring Boot che ulizza la libreria ZUUL.

Il Gateway ha l'incarico di interrogare l'authorization-service per verificare che il token inviato come parametro sia valido, inoltre ha l'incarico di ricevere la chiamata, attraverso il suo endpoint a partire dal contesto (il metodo è stato pensato con questo criterio: contesto = servizio necessario) verifica interrogando il REGISTRY-SERVICE l'indirizzo del servizio richiesto. Attraverso la libreria ZUUL tutte le chiamate verso l'URL di LOGIN vengono tutte dirottate all'AUTHORIZATION-SERVICE.

Questo boot dopo aver ottenuto l'indirizzo del servizio lo interroga, prende in carico la risposta e la invia al client.

AUTHORIZATION - Spring Boot che utilizza Spring Security.

Questo boot ha 3 endpoint: - /login -> Necessario a eseguire il login.

- /tokenauthorization -> Endpoint necessario per la validazione del token. - /logout -> Endpoint utilizzato per eseguire il logout o annullare una sessione.

<u>Il boot inoltre ha un suo databse associato</u> che ha in se il registro delle sessioni. N.B.: Gli user indicati in questo database NON sono gli autori dei post ma semplicemente gli USER legati alla sessione.

Lo UserLog può essere diverso dallo user (autore del post).

REGISTRY-SERVICE - E' l'unico servizio che SA L'URL DEL REGISTRY, infatti questo boot è l'unico accesso al REGISTRY che hanno a disposizione gli altri boot. Ha un'unico endpoint che ricevere come parametro il **NOME DEL SERVIZIO** e grazie a questo, attraverso la classe **DiscoveryService nativa del framework Spring l'EUREKA** in cui si è registrato (il registry), ottiene l'indirizzo del servizio e lo restituisce al GATEWAY.

REGISTRY - E' il registro di tutti i microservizi che sono UP, ha l'unico compito di registare i servizi e comunicarli dopo ogni richiesta al **REGISTRY-SERVICE**.

POST-SERVICE - E' uno spring boot che utilizza la libreria FEIGNCLIENT.

Il boot ha tre funzioni: - Ricevere la chiamta per l'inserimento di un nuovo post.

- Inserire un post, atraverso un endpoint specifico. - Trovare tutti i post realativi a un solo autore dato un ID USER.

Al momento della recezione della chiamata per l'inserimento di un nuovo post il servizi interroga USER-SEVICE (via FEIGNCLIENT) inviandogli tutta la chiamata, quando USER-SERICE verifica la presenza nel DATABASE (suo diverso da quello di sicurezza) dello user ricveve la chiamata per l'inserimento del post andando avanti con l'eseguzione riceve inoltre un'altra chiamata da USER-SERVICE in cui gli viene richiesta la lista completa dei post relativa a un determinato USER.

Alla fine del processo POST-SERVICE ha inoltre l'incarico di ottenere la risposta da USER-SERVICE costituita da un oggetto user con all'interno tutta la sua lista di post inseriti nel database, una volta che a risposta è pronta la inoltra al GATEWAY.

USER-SERVICE - Ha la funzione di ricevere la chiamata proveniente da POST-SERVICE e verificare se lo user passato in testata della chiamata sia presente nel databse. Se lo USER è presente viene inoltrata una chiamata a POST-SERVICE per l'inserimento del post, inoltre riceve tutta la lista dei post inseriti e li mette all'interno di un oggetto USER e fa ritornare l'oggetto valorizzato a POST-SERVICE che si occupa di gestire la risposta alla richiesta. <u>IL DATABASE UTILIZZATO IN QUESTO BOOT è condiviso con POST-SERVICE</u> <u>NON E'</u> lo stesso di AUTHORIZATION-SERVICE.

SEQUENZA CHIAMATE

1 - Chiamata da parte del client.

2 - La chiamata arriva al GAETEWAY. Il GATEWAY chiede l'autorizzazione all' AUTHORIZATION.

3 KO - La chiamata non è autorizzata.

4 KO - Viene inviato al CLIENT la risposta Unauthorize.

3 OK - La chimata viene è autorizzata

4 - II GATEWAY interroga il REGISTRY-SERVICE, questo boot è l'unico che sa l'indirizzo del REGISTRY.

5 - Il REGISTRY-SERVICE chiede al registry l'indirizzo e la porta del servizio richiesto (nome ottenuto come parametro nella chiamata).

6 - Il REGISTRY risponde con l'indirizzo del servizio registrato.

7 - Il REGISTRY-SERVICE risponde con l'indirizzo appena ottenuto.

8 - II GATEWAY-SERVICE una volta ottenuto l'indirizzo contatta POST-SERVICE.

9 - II POST-SERVICE invia tutta la chiamata allo USER-SERVICE utilizzando FEIGNCLIENT.

10 - USER-SERVICE controllo che lo USER autore nel post presente come "userid" nella testata della chiamata sia presente nel DATABASE. Se è presente chiede al POST-SERVICE, sempre attraverso FEIGNCLIENT, l'inserimento del nuovo POST

11 - POST-SERVICE inserisce il POST appena acquisito dal BODY della chiamata e ritorna a USER-SERVICE tutta la lista

12 - USER-SERVICE riceve i post arrivati da POST-SERVICE e li associa all'utente autore e ritorna a POST-SERVICE l'oggetto **USER** con all'interno tutti sui post.

13 - POST-SERVICE riceve il model USER e lo invia al GATEWAY.

14 - IL GATEWAY risponde al CLIENT con lo user autore del post e la lista dei post inseriti dall'utente.

Informazioni necessarie all'eseguzionione dei test

STRUMENTI: Postman App

Eseguire il login all'indirizzo:

http://localhost:8080/secure-service/login?username=username&password=password

Come si può vedere dalla chiamata viene inviata al gateway username e password di default che sono username=username e password=password. Il servizio risponderà con un token UUID necessario all'eseguzione della seconda chiamata.

2 - Eseguire la chiamata POST per l'nseriento di un nuovo post, la chiamata deve essere strutturata cosi:

- Come parametro dell'header deve essere inserito l'id dello user con la chiave "userid" e con valore 1 (è l'unico user inserito nel DB per fare test). - Come body deve essere inseritio il post nella forma json come nell'esempio :

"title": "Titolo inserito!", "body": "Body inserito!"

- Inserire come paramtro della chiamata il token ottenuto dalla chiamata precedente.

http://localhost:8080/api/post-service/postuser/?token=47506b59-c7a6-41d9-86b9-f030fce84b41

Dopo aver scaricato dal repository i boot, importarli in un IDE come Spring Tools o Eclipse e avvire i progetti con in questa sequenza per evitare errori di mancata registrazione dei servizi:

Se la chimata è andata a buon fine il risultato dovrebbe essere un json con all'interno l'utente e tutti i post a lui associati.

1- Registry 2- Gateway

3- Registry service 4- Post-service

5- User-service

6- Authorization