

Federico Calò

Comunicazione tra robot medici attraverso REST API

Esame di Sistemi ad Agenti

Overview

01 Introduzione

02 Modellazione DB

03 Creazione schema di
comunicazione

04 Creazione dei servizi

05 Creazione del BFF

06 Sviluppi futuri

01 Introduzione

L'intenzione del progetto è quella di creare un sistema di comunicazione attraverso i servizi REST API tra diversi robot medici. L'ambiente in cui i robot si troveranno a dover agire potrebbe essere un ospedale o una RSA, mentre gli attori coinvolti sono i medici, gli infermieri, i pazienti e un servizio di help-desk e di enciclopedia



Tecnologie utilizzate

Il sistema è stato realizzato in Java, sfruttando il framework SpringBoot per la realizzazione delle chiamate REST API. Una piccola interfaccia grafica è stata resa possibile grazie alla dipendenza Swagger.

Il database è stato modellato attraverso il linguaggio SQL, in quanto si è adottato un database relazionale. Ciò non toglie che si potrebbe convertire in un database non relazionale, modificando la parte relativa al recupero delle informazioni dal database.

02 Modellazione DB

Inizialmente si è partiti con un'analisi del database, creando lo schema delle varie tabelle e caricando alcuni dati di esempio in esse. Durante la fase di sviluppo il database è stato adattato alle esigenze.

03 Creazione schema di comunicazione

Il primo passo dello sviluppo del sistema ha riguardato la necessità di creare uno schema di comunicazione tra i vari robot. Infatti si è ritenuto opportuno introdurre una classe di configurazione per indicare l'URL dove inoltrare le richieste ai vari robot, e fare in modo che tutte le richieste che provenissero da quell'URL fossero associate a un determinato robot.

```

Federico Calò *
@Configuration
@RequiredArgsConstructor
public class ExternalApiClientConfig {

    /**
     * The Rest template.
     */
    private final RestTemplate restTemplate;

    Federico Calò
    @Primary
    @Bean
    public com.bff.msmedico.invoker.ApiClient msMedicoApiClient() {
        com.bff.msmedico.invoker.ApiClient apiClient = new com.bff.msmedico.invoker.ApiClient(restTemplate);
        apiClient.setBasePath("http://localhost:8081");
        return apiClient;
    }

    Federico Calò
    @Primary
    @Bean
    public com.bff.msinfermiere.invoker.ApiClient msInfermieriApiClient() {
        com.bff.msinfermiere.invoker.ApiClient apiClient = new com.bff.msinfermiere.invoker.ApiClient(restTemplate);
        apiClient.setBasePath("http://localhost:8082");
        return apiClient;
    }
}

```


Un secondo passo è stato quello di definire uno schema fisso di risposta delle varie richieste REST. In questo caso si è creato un oggetto con una proprietà “payload” variabile in base a quello che la chiamata REST doveva restituire, e una proprietà “esito” fissa che descriveva se l’operazione fosse andata a buon fine e quali messaggi avesse riportato.

```
{
  "esito": {
    "codRet": "OK",
    "messaggi": [
      {
        "severita": "INFO",
        "codMsg": "string",
        "parametri": [
          {
            "nome": "string",
            "valore": "string"
          }
        ]
      }
    ],
    "operationId": "string"
  },
  "payload": {
    "idSintomo": 0,
    "nome": "string",
    "descrizione": "string"
  }
}
```

04 Creazione dei servizi

Ms Enciclopedia

Il microservizio di enciclopedia fornisce informazioni su malattie e sintomi attraverso query di ricerca.

Ms Paziente

Attraverso questo microservizio ogni paziente potrà visualizzare le informazioni della propria cartella clinica e aggiungerne altre.

Ms Infermiere

Ogni infermiere può aggiungere le informazioni relative ai pazienti, contattare i medici e i pazienti.

Ms Medico

Attraverso questo microservizio i medici possono contattare infermieri o i pazienti. Possono prescrivere medicinali e sottoministrare visite

Ms Info

Questo microservizio è pensato per l'help desk. Vengono fornite informazioni relative ai medici, ai pazienti e agli infermieri

05 Creazione del BFF

Il BFF (Backend For Frontend) è quel microservizio creato per far interfacciare tutti i microservizi con un eventuale applicativo Frontend. Impedendo al frontend di chiamare un microservizio che si interfaccia direttamente con il database, si riduce il rischio di sql injection.

06 Sviluppi futuri

Possibili futuri sviluppi possono consistere nel creare un microservizio per la gestione degli specialisti e degli specializzandi, migliorare il microservizio di enciclopedia con maggiori informazioni su malattie, medicinali e sintomi. Un'importante sviluppo consiste nell'implementare un frontend che si interfaccia con il sistema ospedaliero o della struttura RSA.