Installazione

Il presente capitolo fornisce un'analisi approfondita sull'importanza dell'installazione del software e illustra il processo dettagliato per installare correttamente e eseguire un'applicazione web. Saranno esplorati i passaggi necessari per assicurarsi che l'installazione venga effettuata con successo e che l'applicazione web possa essere eseguita senza problemi. Saranno fornite istruzioni chiare e dettagliate per guidare gli utenti attraverso il processo di installazione, compresi i requisiti di sistema, la configurazione delle dipendenze e le procedure di avvio dell'applicazione web. L'obiettivo è fornire una guida completa per garantire una corretta installazione e un'esperienza di esecuzione senza intoppi dell'applicazione web.

8.1 Installazione software in locale

Per l'installazione in locale del software, l'applicazione viene fornita come file JAR chiamato "manvsclass.jar". È importante assicurarsi che il file di configurazione "application.properties" sia presente nella stessa cartella del file JAR per garantire un corretto funzionamento. Attraverso il file di configurazione, è possibile configurare diversi parametri, inclusi quelli relativi al database e alla connessione ad esso.

Per garantire la compatibilità, il sistema su cui viene installato il software deve avere installata una versione di Java 17 o successiva e un database documentale MongoDB. Assicurarsi che queste dipendenze siano soddisfatte prima di procedere con l'installazione.

È importante notare che eventuali modifiche apportate al progetto originale richiederanno la rigenerazione del file JAR, quindi sarà necessario generare un nuovo file JAR con le modifiche applicate prima di eseguire l'installazione o l'esecuzione dell'applicazione.

8.1.1 Generazione del file jar

Utilizzando l’IDE “Eclipse” è stato utilizzata la seguente procedura per la generazione del file Jar:

1. Importare il progetto all’interno dell’IDE. A tale scopo: File->Open Projects from File System-> Selezionare come Import source la cartella del progetto-> Spuntare la casella “Search for nested projects” -> Finish.
2. Una volta che il progetto viene caricato e buildato e le dipendenze risolte bisogna cliccare su File->Export->Java->Jar File-> Selezionare l’intera cartella “manvsclass” da esportare->Finish.
   NB: Se sono presenti dei jar file precedenti vanno prima eliminati altrimenti si genererà un errore nella generazione del file jar.

8.1.2 Esecuzione con Docker Desktop

Docker Desktop è un'applicazione che consente di creare e eseguire oggetti in un ambiente virtuale chiamati "Docker". In questo contesto, gli oggetti Docker si riferiscono all'applicazione web e al database MongoDB. Per crearli, è necessario definire i parametri di configurazione in un file apposito chiamato "docker-compose.yml". Questi parametri definiscono le dipendenze necessarie per l'esecuzione dell'applicazione.

Inoltre, è richiesto un altro file chiamato "Dockerfile" per specificare i parametri del container che si desidera generare. Questi parametri includono le porte che devono essere esposte all'esterno, il file eseguibile, l'ambiente in cui deve essere eseguito il file "manvsclass.jar" e l'entry point, che specifica quale file deve essere eseguito come file principale all'interno del container.

Il file "docker-compose.yml" contiene due sezioni: la prima riguarda la configurazione dell'applicazione principale e la seconda riguarda la configurazione del database. Per installare l'applicazione utilizzando Docker, è necessario creare i file "docker-compose.yml" e "Dockerfile" con le corrette configurazioni.

```yaml
# Versione del Compose
version: '3.12.12'

# Servizi
services:
  # Servizio controller
  controller:
    build: .
    restart: always
    ports:
      - 8080:8080
    depends_on:
      - mongo_db
    volumes:
       - ./FilesUpload:/Files-Upload
  # Servizio MongoDB
  mongo_db:
    image: "mongo:6.0.6"
    restart: always
    ports:
      - 27017:27017
    volumes:
          - ./mongo-init.js:/docker-entrypoint-initdb.d/mongo-init.js:ro
```

Docker-Compose

```
FROM maven AS build
WORKDIR /app
COPY pom.xml /app
RUN mvn dependency:resolve
COPY . /app
RUN mvn clean
RUN mvn package -DskipTests


FROM eclipse-temurin:20-jdk-alpine
COPY --from=build /app/target/*.jar manvsclass.jar
ENV PORT 8080
EXPOSE 8080
CMD ["java","-jar","manvsclass.jar"]
```

DockerFile

8.1.3 Esecuzione con parametri di Default

In questo caso l'applicazione viene fornita già con il docker-compose e DockerFile pronti all'esecuzione. Per poter creare il container e i suoi Docker bisogna eseguire nella cartella principale il comando da power shell Windows:

docker-compose build
docker-compose up

Con privilegi di amministratore

8.1.4 Definizione degli indici di ricerca nel database

Ecco una descrizione dei passaggi da effettuare per l'utilizzo dei comandi nel terminale del container MongoDB:

1. Utilizzare dal terminale del container MongoDB il comando "mongosh": Questo comando consente di avviare l'interfaccia interattiva della shell di MongoDB all'interno del terminale del container. Una volta avviata la shell, sarà possibile eseguire comandi e interrogazioni sul database.
2. Utilizzare il comando "use manvsclass" per selezionare il database "manvsclass": Questo comando consente di selezionare il database "manvsclass" come database di lavoro corrente. Tutti i comandi successivi verranno eseguiti all'interno di questo database.
3. Utilizzare i comandi :

   db.createCollection("ClassUT");

   db.createCollection("interaction");

db.createCollection("Admin");

db.createCollection("Operation");

Per creare le collezioni: Questi comandi creano le collezioni nel database corrente. In particolare, vengono create le collezioni "ClassUT", "interaction", "Admin" e "Operation".

4. Utilizzare i comandi:

   db.ClassUT.createIndex({ difficulty: 1 }) db.Interaction.createIndex({ name: "text", type: 1 }) db.interaction.createIndex({ name: "text" }) db.Admin.createIndex({username: 1})

   Per creare gli indici: Questi comandi creano gli indici sulle collezioni specificate. Ad esempio, "db.ClassUT.createIndex({ difficulty: 1 })" crea un indice sulla collezione "ClassUT" con il campo "difficulty" ordinato in modo ascendente.

Una volta generati questi indici di ricerca, le pipeline funzioneranno correttamente e sarà quindi possibile utilizzare il software in modo corretto.