



Progetto Spring Azienda

3 entity, JPA, relazioni OneToOne e OneToMany

Descrizione

- ▶ Si vuole realizzare un progetto Spring Boot per la gestione di aziende.
- ▶ Le aziende saranno identificate univocamente mediante un ID.
- ▶ Inoltre un'azienda è costituita da dipendenti, possiede una intestazione e un capitale sociale
- ▶ Il dipendente invece è definito da nome, cognome, possiede una matricola univoca, un salario, opzionalmente un posto auto.
 - ▶ Il posto auto è definito da un ID e una posizione (campo alfanumerico es. ‘A2’, ‘B8’)
- ▶ Un impiegato NON può esistere senza l'azienda di riferimento
- ▶ Un impiegato può esistere anche senza posto auto (infatti glielo si può assegnare anche dopo l'assunzione in azienda)

Funzionalità relative alle aziende

1. Inserisci nuova azienda (vuota, senza dipendenti)
2. Visualizza tutti i dati di azienda tramite l'id
3. Visualizza solo i dati base (no lista dipendenti) di un'azienda tramite l'id
4. Visualizza tutte le aziende
5. Visualizza nome azienda e numero dipendenti relativi ad un dato ID azienda
6. Visualizza i dati base dell'azienda e numero dipendenti relativi all'azienda con capitale sociale maggiore
7. Modifica il capitale sociale di un'azienda e torna dati base dell'azienda modificata
8. Modifica l'intestazione di un'azienda e torna dati base dell'azienda modificata
9. Cancella un'azienda esistente (se non possiede dipendenti)

Funzionalità relative ai dipendenti

- ▶ Fissata un'azienda → tramite l'ID
 - 10. Inserisci nuovo impiegato, senza posto auto
 - ▶ Inserisci nuovo impiegato con posto auto
 - ▶ Inserisci nuovo impiegato e assegnagli il posto auto dato dall'idAuto specificato
 - 11. Visualizza tutti gli impiegati
 - 12. Visualizza la lista dei nominativi (nome e cognome) degli impiegati
 - 13. Ricerca tutti gli impiegati con un dato salario
 - 14. Cancella un impiegato tramite la matricola e ritornalo
 - 15. Cancella un impiegato tramite la matricola e torna solo nome e cognome
 - 16. OPZIONALE: Sposta un impiegato da un'azienda in un'altra
- ▶ Fissato l'impiegato tramite matricola
 - 17. Modifica salario di un impiegato esistente e torna l'impiegato modificato
 - 18. Modifica posto auto di un impiegato esistente e torna l'impiegato modificato
 - 19. Visualizza il posto auto di un impiegato, se lo possiede

Servizi dell'azienda

Servizi dell'impiegato

Funzionalità relative ai posti auto

- ▶ Inserisci un posto auto, senza assegnarlo a nessuno
- ▶ Visualizza tutti i posti auto
- ▶ Visualizza un posto auto tramite l'id
- ▶ Cancella un posto auto, solo se non è assegnato a nessun dipendente



Servizi dell'impiegato

Architettura

- ▶ *Gestire la persistenza con MySQL server e JPA/ Hibernate*
- ▶ Utilizzare Tomcat Application Server per i servizi **REST**
- ▶ Strutturare il progetto con architettura **MVC**, prevedendo i seguenti package:
 - ▶ controller
 - ▶ service
 - ▶ dao
 - ▶ entity
 - ▶ dto

Configurazioni

- ▶ Nome pacchetto principale: **com.spring.aziende**
- ▶ Definizioni nell' application.properties:

```
server.port=8080
```

```
spring.mvc.servlet.path=/gestioneazienda
```

```
spring.datasource.url=jdbc:mysql://localhost:3306/xxxx?createDatabaseIfNotExist=true&autoReconnect=true&useSSL=false
```

```
spring.datasource.username=xxxx
```

```
spring.datasource.password=xxxx
```

Personalizzare le credenziali di MySQL

```
spring.jpa.hibernate.ddl-auto=update
```

```
spring.jpa.show-sql=true
```