

Guia de Instalação e Execução do Projeto Spring Boot para Sistema de Revenda de Ingressos

Será disponibilizado um vídeo com o processo de execução do código.

1. Baixar o Código Fonte

Primeiramente, baixe o código fonte do projeto a partir do **arquivo disponibilizado** no e-mail. Caso tenha problemas também é possível obter o código pelo repositório do GitHub: <https://github.com/gustavo-gm-dev/TicketProject.git>

2. Configurar o Ambiente de Desenvolvimento

2.1 Instalar uma IDE (IntelliJ IDEA ou Eclipse)

Para editar e executar o código Java, você precisará de uma IDE. Recomenda-se o uso do IntelliJ IDEA ou do Eclipse.

IntelliJ IDEA Community Edition:

<https://www.jetbrains.com/idea/download/?section=windows>

(Baixar versão gratuita, basta apenas descer a tela para encontrar)

Eclipse: <https://www.eclipse.org/downloads/>

2.2 Instalar PostgreSQL

Baixe e instale o PostgreSQL 16 a partir do seguinte link:

<https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>

Para configuração funcionar deve deixar os dados padrão.

- Porta: 5432
- Username: postgres
- Senha: 123

Caso contrário será necessário modificar o **application.properties**

2.3. Instalar Postman

Baixe e instale o Postman para testar as APIs. O Postman pode ser baixado:

<https://www.postman.com/downloads/>

3. Configurar o Banco de Dados

3.1. Criar o Banco de Dados

Após instalar o PostgreSQL, crie um banco de dados chamado Tickets.

1. Abra o pgAdmin ou qualquer outro cliente de banco de dados.
2. Conecte-se ao servidor PostgreSQL.
3. Execute o seguinte comando SQL para criar o banco de dados:

```
CREATE DATABASE ticket;
```

4. Importar Coleção de APIs no Postman

1. Abra o Postman.
2. Importe a coleção de APIs fornecida pelo projeto (o arquivo .json deve estar incluído no repositório do código fonte).

5. Executar o Código Java em Spring Boot

5.1. Importar o Projeto na IDE

Abra a sua IDE preferida (IntelliJ IDEA ou Eclipse).

Importe o projeto como um projeto Maven:

- No IntelliJ IDEA: File -> Open e selecione a pasta do projeto.
- No Eclipse: File -> Import -> Existing Maven Projects e selecione a pasta do projeto.

5.2. Configurar o application.properties

Caso configure algo diferente no postgresql, basta adaptar de acordo. Edite o arquivo src/main/resources/application.properties para configurar a conexão com o banco de dados PostgreSQL. Adicione ou edite as seguintes linhas:

```
spring.application.name=ticketProject
spring.datasource.url=jdbc:postgresql://localhost:5432/ticket
spring.datasource.username=postgres
spring.datasource.password=123
spring.datasource.driver-class-name=org.postgresql.Driver
hibernate.dialect=org.hibernate.dialect.HSQLDialect

spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
spring.jpa.format-sql=true
```

(Substitua <SUA_SENHA> pela senha configurada para o usuário postgres durante a instalação do PostgreSQL.)

5.3. Executar a Aplicação

No IntelliJ IDEA:

- Abra o painel Maven (View -> Tool Windows -> Maven).
- Navegue até ticketProject -> Lifecycle -> install e clique duas vezes para construir o projeto.
- Navegue até a classe principal que contém o método main TicketProjectApplication e execute-a clicando com o botão direito e selecionando Run TicketProjectApplication

No Eclipse:

- Clique com o botão direito no projeto e selecione Run As -> Maven install.
- Navegue até a classe principal e execute-a clicando com o botão direito e selecionando Run As -> Java Application.

6. Testar as APIs com Postman

1. Abra o Postman.
2. Selecione a coleção importada.
3. Teste os endpoints fornecidos. Certifique-se de que a aplicação Spring Boot esteja em execução antes de enviar as requisições.
4. No postman importado poderá notar que possui uma sequência enumerada, execute na ordem para respeitar a regra de negocio

