

Internet em Sistemas de Informação

Profª Elyda Laiza
Aula 02 – Spring Boot - Prática

Preparando o Projeto

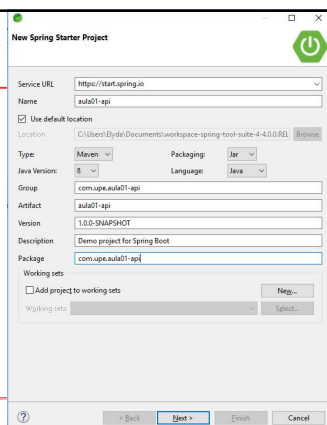
- Abra o STS



2

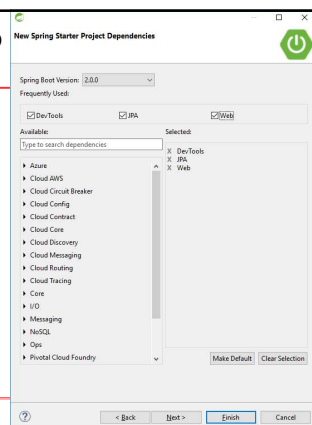
Preparando o Projeto

- Escolha a opção New -> Spring Starter Project



Preparando o Projeto

- Agora, selecione as dependências do projeto
 - Pesquise por WEB, JPA e DevTools
 - Confirme as escolhas
 - Se necessário, podem ser adicionadas outras dependências
- Clique em Finish e o projeto será criado



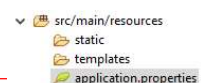
Preparando o Projeto

- O pom.xml gerencia as dependências Maven do projeto
 - Por exemplo, o spring-boot-starter-data-jpa tem uma série de dependências pro JPA funcionar
 - Você não precisa baixar nada (observe a Maven Dependencies no seu projeto)
- O mvnw
 - Maven embarcado na sua aplicação
 - Não precisa ter ele instalado na sua máquina

5

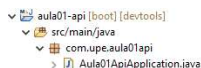
Preparando o Projeto

- Os recursos da aplicação
 - HTML
 - Templates
- application.properties
 - Arquivo de configuração do Spring boot – veremos em breve
 - Como a conexão com Banco de dados



Preparando o Projeto

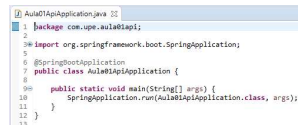
- Dentro de src/main/java ficarão nossas classes



- Agora, abra Aula01ApiApplication.java
 - É a classe que inicia nossa aplicação
 - Sem configurar servidor
 - Sem as burocracias já conhecidas

Preparando o Projeto

- @SpringBootApplication
 - Dentro dessa anotação há uma série de anotações, onde o Spring vai monitorar dependências e componentes do pacote API para fazer uma autoconfiguração



- Execute a aplicação: run as -> Spring Boot App

CONECTANDO AO BD

Sobre o BD

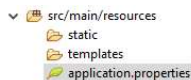
- Decidi que vamos usar o MySQL por 2 razões:
 - Uma das ferramentas que vamos usar, o Flyway, não permite o uso do XE
 - Poderíamos então baixar a versão full do Oracle, mas isso poderia pesar muito nos computadores
- Então vamos de MySQL, mas vou manter as configurações do Oracle, caso desejem usar o ambiente Oracle + Spring
 - Para o MySQL, baixe a MySQL Community Server. Para o Oracle, pelo menos, a versão 12c

Preparando o Projeto - MySQL

- Para o **MySQL**
- Acesse o application.properties e escreva

O BD será criado se não existir

```
spring.jpa.database=MYSQL
spring.datasource.url=jdbc:mysql://localhost/teste?createDatabaseIfNotExist=true
spring.datasource.username=root
spring.datasource.password=123456
server.port = 8081
```



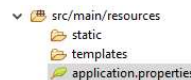
```
spring.jpa.show-sql=true
```

Preparando o Projeto - Oracle


- Para **Oracle**
- Acesse o application.properties e escreva:

```
spring.datasource.url=jdbc:oracle:thin:@localhost:1521:xe
spring.datasource.username=hr
spring.datasource.password=123456
spring.datasource.driver-class-name=oracle.jdbc.driver.OracleDriver

server.port = 8081
```



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




Preparando o Projeto

- Para **MySQL**: E no pom.xml, adicione adequadamente essa nova dependência:

```
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
</dependency>
```

13




Preparando o Projeto

- Para **Oracle**: E no pom.xml, adicione adequadamente essa nova dependência:

```
<dependency>
  <groupId>com.oracle</groupId>
  <artifactId>ojdbc7</artifactId>
  <version>12.1.0</version>
</dependency>
```

14




Preparando o Projeto

- Execute mais uma vez a aplicação run as -> Spring Boot App
 - Verifique se houve erros
 - Se não... Sigamos!

15

FLYWAY


16



Flyway

- No nosso projeto usaremos Hibernate
 - Mas como somos bons em BD, criaremos nossas tabelas
 - E usaremos scripts de migração incremental
 - Pra isso usaremos o flyway (<http://flywaydb.org/>)
 - Na medida em que o BD evolui, fica mais fácil gerenciá-lo
- O flyway procura arquivos de migração
 - Com código SQL

17



Flyway

- Vamos criar nossa primeira tabela
- Primeiro adicione a dependência no pom.xml

```
<dependency>
  <groupId>org.flywaydb</groupId>
  <artifactId>flyway-core</artifactId>
</dependency>
```

18

UPE
UNIVERSIDADE
DE PERNAMBUCO

Flyway

- Agora, crie duas pastas dentro de src/main/resources
 - db e depois migration dentro dela
 - O flyway procura dentro dela pelos scripts
 - Dentro de migration, crie um arquivo chamado VXX__criar_e_registrar_categorias.sql (XX é a versão do seu arquivo)
- Siga o padrão!
 - src/main/resources
 - db
 - migration
 - V01__criar_e_registrar_categorias.sql

19

UPE
UNIVERSIDADE
DE PERNAMBUCO

Flyway

- Agora crie sua tabela e insira 5 linhas de dados

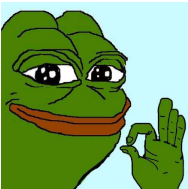

```
1 CREATE TABLE CATEGORIA (
2   CODIGO BIGINT(20) PRIMARY KEY AUTO_INCREMENT,
3   NOME VARCHAR(50) NOT NULL
4 ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
5
6 INSERT INTO CATEGORIA (NOME) VALUES ('Gastos Fixos');
7 INSERT INTO CATEGORIA (NOME) VALUES ('Alimentacao');
8 INSERT INTO CATEGORIA (NOME) VALUES ('Carro');
9 INSERT INTO CATEGORIA (NOME) VALUES ('Remedios');
10 INSERT INTO CATEGORIA (NOME) VALUES ('Outros');
```
- Uma vez criada, você não deve mais fazer mudanças nesse código (dessa migração)
 - Por exemplo, se você quiser remover uma coluna, deve usar um novo migration com o código para dropar a coluna
 - Todas as mudanças são incrementais

20

UPE
UNIVERSIDADE
DE PERNAMBUCO

Flyway

- Execute a aplicação
 - Serão criados o BD, as tabelas e os dados inseridos!
- Além disso, uma nova tabela de versões foi criada



21