

## Desafio Parte 3: Shared Database

No último exercício do desafio, vamos **gravar as negociações no banco de dados**. Nós usaremos o MySQL que você precisa já ter instalado, caso esteja com dúvidas, temos treinamentos que focam exclusivamente no MySQL na Alura. Nosso objetivo é executar um comando SQL `insert` para cada negociação na rota.

1) Conecte-se com o MySQL pelo terminal (ou outro cliente) para criar o banco e a tabela das negociações. Um vez conectado execute os seguintes comandos:

```
create database camel;
create table negociacao (id BIGINT NOT NULL AUTO_INCREMENT, preco DECIMAL(5,2), quantidade MEDIAN)
```

2) No Eclipse abra o `pom.xml` e adicione as seguintes dependências:

```
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-jdbc</artifactId>
  <version>${camel-version}</version>
</dependency>

<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>5.1.36</version>
</dependency>
```

3) Na classe `RotaHttpPollingNegociacoes`, adicione um método que define os dados da conexão:

```
private static MysqlConnectionPoolDataSource criaDataSource() {
    MysqlConnectionPoolDataSource mysqlDs = new MysqlConnectionPoolDataSource();
    mysqlDs.setDatabaseName("camel");
    mysqlDs.setServerName("localhost");
    mysqlDs.setPort(3306);
    mysqlDs.setUser("root");
    mysqlDs.setPassword("");
    return mysqlDs;
}
```

4) No início método `main()`, crie um registro que recebe o `DataSource` do MySQL. Esse registro deve ser passado para o construtor da classe `DefaultCamelContext`:

```
SimpleRegistry registro = new SimpleRegistry();
registro.put("mysql", criaDataSource());
CamelContext context = new DefaultCamelContext(registro); //construtor recebe registro
```

5) Já sabemos que nossa rota divide o XML em várias negociações. Logo após ter feito o `unmarshal` e o `split`, adicione um `Processor` que prepara três propriedades, `preco`, `quantidade` e a `data`:

```
...
unmarshal(new XStreamDataFormat(xStream)). //já tem
split(body()). //já tem
process(new Processor() {
    @Override
    public void process(Exchange exchange) throws Exception {
        Negociacao negociacao = exchange.getIn().getBody(Negociacao.class);
        exchange.setProperty("preco", negociacao.getPreco());
        exchange.setProperty("quantidade", negociacao.getQuantidade());
        String data = new SimpleDateFormat("YYYY-MM-dd HH:mm:ss").format(negociacao.getData());
        exchange.setProperty("data", data);
    }
}).
log("${body}").
end();
```

6) Depois do `Processor()`, coloque um novo `body` com o comando SQL. Esse comando será executado pelo componente `jdbc`:

```
setBody(simple("insert into negociacao(preco, quantidade, data) values (${property.preco}, !
log("${body}"). //logando o comando sql
delay(1000). //esperando 1s para deixar a execução mais fácil de entender
to("jdbc:mysql"); //usando o componente jdbc que envia o SQL para mysql
```

7) Verifique toda a rota e execute. Funcionou?