GUIA EXEMPLO PRÁTICO DE DESIGN INSEGURO

Para acesso à API, clone o repositório:

```
git clone https://github.com/jjgirotto/insecure-design.git
```

Requisitos para execução do projeto:

- Java JDK 21 ou superior;
- Maven;
- Postman (para requisições).

1. Abra o arquivo '*src/main/resources/application.yml*' e altere as configurações do banco de dados MySQL Workbench para utilização na sua máquina:

```
username: root (insira o usuário desejado)
password: 123456 (insira a senha desejada)
url: jdbc:mysql://localhost:3306/alunos?createDatabaseIfNotExist=true&serverTimezone=UTC
(altere a porta se necessário em 'localhost:3306')
```

2. Execute os seguintes comandos:

```
cd appalunos
mvn clean install
mvn spring-boot:run
```

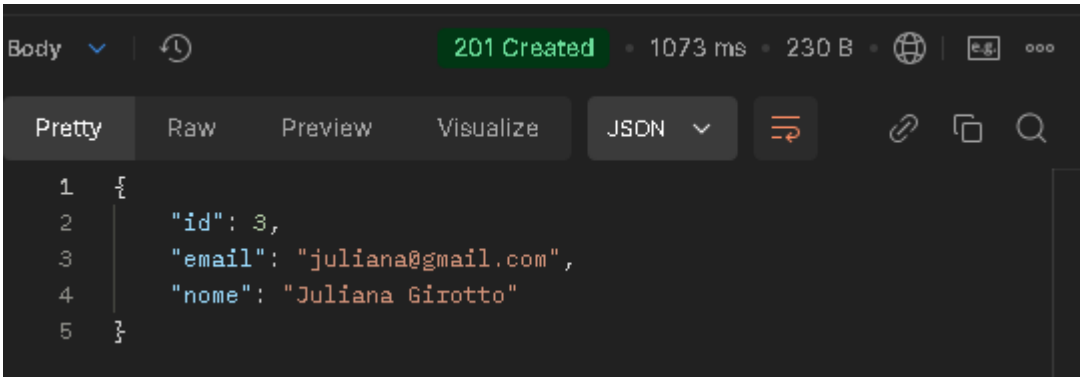
3. Consumo da API no Postman:

URL: http://localhost:8000

ENDPOINT	MÉTODO HTTP	URL	AÇÃO
Aluno	POST	/alunos	Cadastra um aluno
Nota	POST	/notas	Cadastra uma nota
Nota	GET	/alunos/notas	Busca notas
Nota	GET	/alunos/notas/{tipo}	Busca nota por tipo

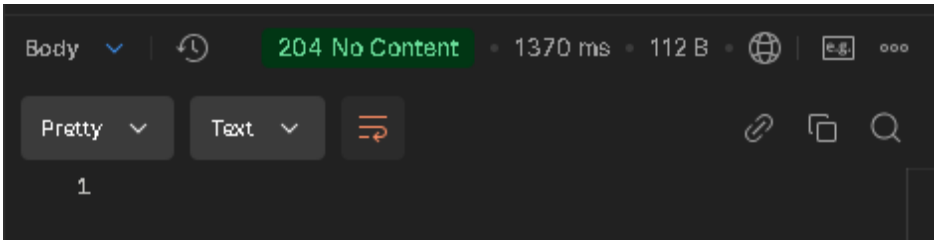
- Cadastrar um aluno:

```
curl --location 'http://localhost:8080/alunos' \
--header 'Content-Type: application/json' \
--data-raw '{
    "email": "juliana@gmail.com",
    "senha": "123456",
    "nome": "Juliana Girotto"
}'
```



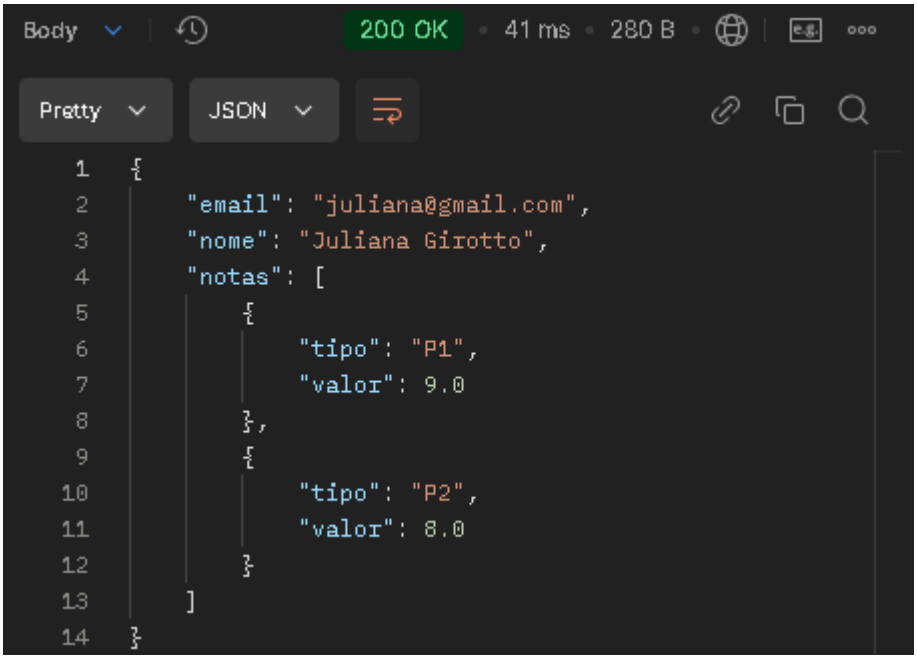
- Cadastrar uma nota:

```
curl --location 'http://localhost:8080/notas' \
--header 'Content-Type: application/json' \
--data '{
    "idAluno": 3,
    "tipo": "P1",
    "valor": 9.0
}'
```



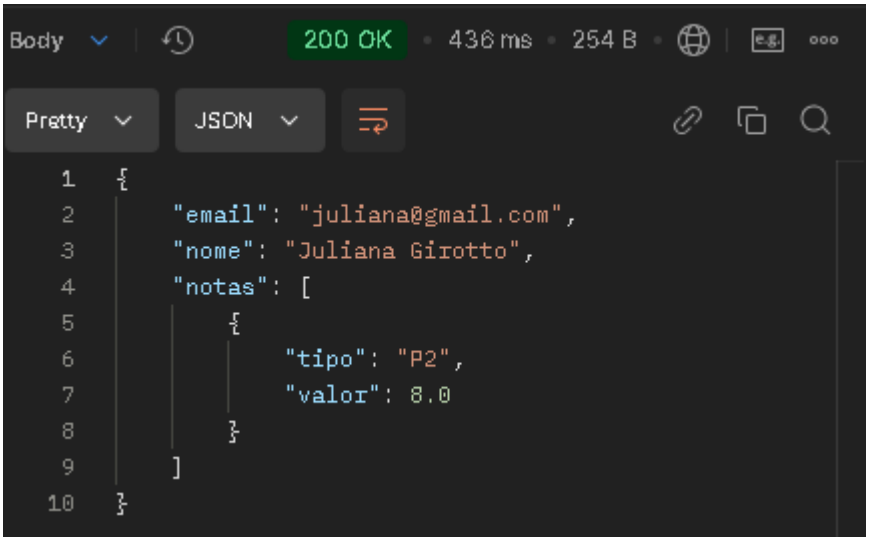
- Consultar notas:

```
curl --location --request GET 'http://localhost:8080/
    alunos/notas' \
--header 'Content-Type: application/json' \
--data-raw '{
    "email": "juliana@gmail.com",
    "senha": "123456"
}'
```



- Consultar notas por tipo:

```
curl --location --request GET 'http://localhost:8080/alunos/notas/P2' \
--header 'Content-Type: application/json' \
--data-raw '{
    "email": "juliana@gmail.com",
    "senha": "123456"
}'
```



4. Em '*application.yml*', altere o item *ddl-auto*:

```
ddl-auto: none
```

5. Agora, no MySQL Workbench, execute a seguinte query para testar se a API foi implementada pensando em um design seguro:

```
ALTER TABLE notas DROP COLUMN tipo;
```

6. Execute a aplicação novamente com as alterações realizadas. Volte ao passo 4 e realize o método *GET* de buscar notas por tipo e utilize o mesmo corpo de requisição utilizado anteriormente. A resposta será a seguinte:


```
{
    "timestamp": "2024-11-22T21:55:37.414+00:00",
    "status": 500,
    "error": "Internal Server Error",
    "trace": "org.springframework.dao.InvalidDataAccessResourceUsageException: JDBC exception executing SQL
        [select n1_0.id,n1_0.id_aluno,n1_0.tipo,n1_0.valor from notas n1_0 where n1_0.id_aluno is null]
        [Unknown column 'n1_0.tipo' in 'field list'] [n/a]; SQL [n/a]\r\n\tat org.springframework.orm.jpa.
        vendor.HibernateJpaDialect.convertHibernateAccessException(HibernateJpaDialect.java:277)\r\n\tat org.
        springframework.orm.jpa.vendor.HibernateJpaDialect.translateExceptionIfPossible(HibernateJpaDialect.
        java:241)\r\n\tat org.springframework.orm.jpa.AbstractEntityManagerFactoryBean.
        translateExceptionIfPossible(AbstractEntityManagerFactoryBean.java:550)\r\n\tat org.springframework.
        dao.support.ChainedPersistenceExceptionTranslator.translateExceptionIfPossible
        (ChainedPersistenceExceptionTranslator.java:61)\r\n\tat org.springframework.dao.support.
        DataAccessUtils.translateIfNecessary(DataAccessUtils.java:335)\r\n\tat org.springframework.dao.support.
        PersistenceExceptionTranslationInterceptor.invoke(PersistenceExceptionTranslationInterceptor.java:160)
        \r\n\tat org.springframework.aop.framework.ReflectiveMethodInvocation.proceed
        (ReflectiveMethodInvocation.java:184)\r\n\tat org.springframework.data.jpa.repository.support.
        CrudMethodMetadataPostProcessor$CrudMethodMetadataPopulatingMethodInterceptor.invoke
```


7. Para correção da exceção com dados sensíveis do banco de dados, recomenda-se a implementação de uma classe global para tratamento de exceções. Portanto, no arquivo *'src/main/java/com/juliana/appalunos/application/exception/GlobalExceptionHandler.java'* e desfaça o comentário que contém o tratamento previamente realizado da exceção em estudo: *DataAccessException*.

8. Execute a aplicação novamente com as alterações realizadas. Volte ao passo 4 e realize o método *GET* de buscar notas por tipo e utilize o mesmo corpo de requisição utilizado anteriormente. A resposta será a seguinte: