

Documentação do Place Service

Jorge Fernando Ramos Bezerra

August 16, 2024

1 Tecnologias

- Spring Boot
- Spring Webflux
- Spring Data + R2DBC
- SpringDoc OpenAPI 3
- Slugify

2 Práticas Adotadas

- **SOLID:** Princípios de design de software para manter a modularidade e a facilidade de manutenção.
- **Testes Automatizados:** Implementação de testes para garantir a funcionalidade correta e a integridade do código.
- **Consultas com Filtros Dinâmicos:** Uso do Query By Example para permitir consultas flexíveis e dinâmicas.
- **API Reativa:** Implementação de uma API reativa tanto na camada web quanto na camada de banco de dados.
- **Uso de DTOs:** Utilização de Data Transfer Objects (DTOs) para expor apenas as propriedades necessárias na API.
- **Injeção de Dependências:** Utilização de injeção de dependências para promover a modularidade e o desacoplamento.
- **Geração Automática do Swagger:** Documentação da API gerada automaticamente com OpenAPI 3.
- **Geração de Slugs Automática:** Utilização da biblioteca Slugify para criar slugs baseados nos nomes.
- **Auditoria:** Implementação de auditoria para rastrear a criação e atualização das entidades.

3 Como Executar

3.1 Localmente

1. Clone o repositório Git:

```
git clone <https://github.com/jorgefernandorb/place_service>
```

2. Construa o projeto:

```
./mvnw clean package
```

3. Execute o projeto:

```
java -jar place-service/target/place-service-0.0.1-SNAPSHOT.jar
```

4. A API estará disponível em localhost:8080. O Swagger pode ser visualizado em localhost:8080/swagger-ui.html.

3.2 Usando Docker

1. Clone o repositório Git:

```
git clone <https://github.com/jorgefernandorb/place_service>
```

2. Construa o projeto:

```
./mvnw clean package
```

3. Construa a imagem Docker:

```
./mvnw spring-boot:build-image
```

4. Execute o container:

```
docker run --name place-service -p 8080:8080 -d place-service:0.0.1-SNAPSHOT
```

5. A API estará disponível em localhost:8080. O Swagger pode ser visualizado em localhost:8080/swagger-ui.html.

4 API Endpoints

Para fazer as requisições HTTP abaixo, utilize a ferramenta httpie, http POST :8080
`http://localhost:8080/places name="Place" city="City" state="State"`

```
HTTP/1.1 200 OK
Content-Length: 129
Content-Type: application/json

{
  "createdAt": "2024-08-20T19:00:07.241632",
  "name": "Place",
  "slug": "place",
  "city": "City",
  "state": "State",
  "updatedAt": "2024-08-20T19:00:07.241632"
}
```

5 Observações

- O atributo `id` é gerado automaticamente pelo banco de dados com auto incremento.
- A utilização de DTOs permite expor apenas as propriedades necessárias da entidade `Place`, evitando a exposição direta dos objetos de domínio.
- A biblioteca `slugify` é utilizada para garantir a consistência entre o nome e o slug.