

Desenvolvendo WebServices REST com Spring Boot

Guia de Referência Rápida

Curso de Sistemas da Informação - UniFacisa

Professor: Daniel Abella (daniel@daniel-abella.com / www.daniel-abella.com)

1. Verbos HTTP

RESTful		
Verbo	URI (substantivo)	Ação
POST	/bookmarks	Criar
GET	/bookmarks/1	Visualizar
GET	/bookmarks/	Visualizar
PUT	/bookmarks/1	Alterar
DELETE	/bookmarks/1	Apagar

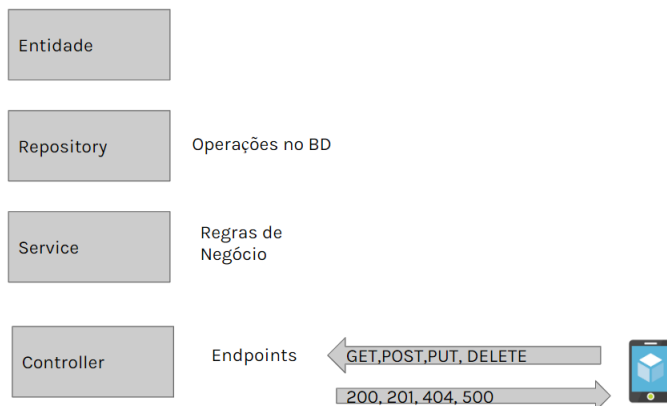
2. Retornos HTTP

100 – Continue	400 – Bad Request
200 – OK	401 – Unauthorized
201 – Created	403 – Forbidden
301 – Moved Permanently	404 – Not Found
303 – See Other	405 – Method Not Allowed
304 – Not Modified	500 – Internal Server Error

3. JSON

<pre>public class Conta { private int agencia; private int conta; private String titular; private double saldo; //getters e setters }</pre>	<pre>{ "agencia":123, "conta":12345, "titular":"Daniel", "saldo":1000.50 }</pre>	<pre>[{ "agencia":123, "conta":12345, "titular":"Daniel", "saldo":1000.50 }, { "agencia":123, "conta":12344, "titular":"Hamurabi", "saldo":100000.50 }]</pre>	<pre>[]</pre>
Classe Java	Objeto	Lista com 2 Objetos	Lista Vazia

4. Componentes



- Para Usuario: Usuario, UsuarioRepository, UsuarioService e UsuarioController
- Para Conta: Conta, ContaRepository, ContaService e ContraController

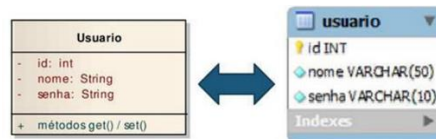
4.1 Entidade

```
import java.io.Serializable;

@Entity
@Table
public class User implements Serializable {

    @Id
    private String id;
    private String name;
    private String address;

    public User() {
    }
}
```



4.2 Repository

```
public interface UserRepository extends JpaRepository<User, String> {
}
}
```

- #1: Cria uma interface que estenda JpaRepository
- #2: Entidade na qual se relaciona o Repository (User)
- #3: Tipo do campo anotado com @Id na entidade

↓

```
@Id
private String id;
```

4.3 Service

```
@Service
@Validated
public class UserService {

    @Autowired
    private UserRepository repository;

    public User getById(String id) {
        return repository.findOne(id);
    }

    public List<User> listAllUsers() {
        return repository.findAll();
    }
}
```

4.4 Controller

```
@RestController
public class UserController {

    private final UserService participantService;

    @Autowired
    public UserController(final UserService participantService) {
        this.participantService = participantService;
    }

    @RequestMapping(value="/user", method = RequestMethod.GET)
    public ResponseEntity< List<User> > listAllParticipants() {
        return new ResponseEntity< List<User> >
            (participantService.listAllParticipants(), HttpStatus.OK);
    }
}
```

5. Query Methods

Query methods são métodos que recuperam informação do banco de dados e estão declarados na interface repository, na qual a consulta é baseada no nome do método. No exemplo abaixo, temos uma consulta por nome.

```
public interface UserRepository extends JpaRepository<User, String> {

    public User findByName(String name);
}
```

Palavra-chave	Exemplo	Código JPQL
And	findByLastnameAndFirstname	... where x.lastname = ?1 and x.firstname = ?2
Or	findByLastnameOrFirstname	... where x.lastname = ?1 or x.firstname = ?2
Is, Equals	findByFirstname, findByFirstnameIs, findByFirstnameEquals	... where x.firstname = 1?
Between	findByStartDateBetween	... where x.startDate between 1? and ?2
LessThan	findByAgeLessThan	... where x.age < ?1
LessThanEqual	findByAgeLessThanEqual	... where x.age <= ?1
GreaterThan	findByAgeGreaterThan	... where x.age > ?1
GreaterThanEqual	findByAgeGreaterThanEqual	... where x.age >= ?1
After	findByStartDateAfter	... where x.startDate > ?1
Before	findByStartDateBefore	... where x.startDate < ?1
IsNull	findByAgeIsNull	... where x.age is null
IsNotNull, NotNull	findByAge (Is) NotNull	... where x.age not null
Like	findByFirstnameLike	... where x.firstname like ?1
NotLike	findByFirstnameNotLike	... where x.firstname not like ?1
StartingWith	findByFirstnameStartingWith	... where x.firstname like ?1 (parameter bound with appended %)
EndingWith	findByFirstnameEndingWith	... where x.firstname like ?1 (parameter bound with prepended %)
Containing	findByFirstnameContaining	... where x.firstname like ?1 (parameter bound wrapped in %)
OrderBy	findByAgeOrderByLastnameDesc	... where x.age = ?1 order by x.lastname desc
Not	findByLastnameNot	... where x.lastname <> ?1
In	findByAgeIn (Collection<Age> ages)	... where x.age in ?1
NotIn	findByAgeNotIn (Collection<Age> age)	... where x.age not in ?1
True	findByActiveTrue()	... where x.active = true
False	findByActiveFalse()	... where x.active = false
IgnoreCase	findByFirstnameIgnoreCase	... where UPPER(x.firstname) = UPPER(?1)