Objetivo:

- ter o servico usando POST/json

----------------------------

**\*\* COMANDOS IMPORTANTES HEROKU\*\***

----------------------------

\*heroku login (faz login no heroku)

\*heroku create (cria um vinculo heroku/git com aquela pasta)

\*heroku git:remote -a newname (renomeia o repositorio)

\*heroku open (abrir o a aplicação)

\*heroku logs --tail ou heroku logs(exibe logs)

\*heroku ps (exibe quantos containers estao sendo executados)--> dyno

\*heroku ps:scale web=1 (sobe um container para executar as aplicações)

\*git init (criar um repositorio)

\*git clone (obter um repositorio ja existente)

\*git push heroku master (envia codigo para o git)

\*git add . (adiciona todas as alteracoes)

\*git commit -m "Demo" ("envia" alteração para o repositorio)

\*git push heroku master (envia efetivamente a alteração para o repositorio master)

\*git pull heroku master (busca a alteração para o repositorio master)

\*git status(exibe o que tem de diferenca do repositorio com o local)

\*git checkout \* (descarta todas as alteracoes locais)

*dica: http://rogerdudler.github.io/git-guide/index.pt\_BR.html*

-------------------------------

**\*\*\*\* Para rodar localmente**

WINDOWS 🡪 target\bin\webapp.bat

LINUX 🡪 sh target/bin/webapp

-------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**RESTEASY USANDO POST**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. Com form

-------------------------------

<form method="POST" action="login">

Email Address: <input type="text" name="email">

<br>

Password: <input type="text" name="password">

<input type="submit">

</form>

-------------------------------

@Path("/")

public class LoginService

{

@Path("login")

@POST

public String login(@**FormParam**("email") String e, @**FormParam**("password") String p) {

return "Logged with " + e + " " + p;

}

1. Sem form (e usando json)

\*necessário usar a biblioteca **Jackson JSON processor para converter objetos json e devolver json para o cliente**

**POM.XML**

**<dependency>**

**<groupId>org.jboss.resteasy</groupId>**

**<artifactId>resteasy-jackson-provider</artifactId>**

**<version>3.0.9.Final</version>**

**</dependency>**

**Classe modelo**

**package br.edu.fcv.model;**

**public class Produto {**

**private String nome;**

**private int quantidade;**

**public String getNome() {**

**return nome;**

**}**

**public void setNome(String nome) {**

**this.nome = nome;**

**}**

**public int getQuantidade() {**

**return quantidade;**

**}**

**public void setQuantidade(int quantidade) {**

**this.quantidade = quantidade;**

**}**

**}**

**JSONService.java**

**package br.edu.fcv.rest;**

**import javax.ws.rs.Consumes;**

**import javax.ws.rs.GET;**

**import javax.ws.rs.POST;**

**import javax.ws.rs.Path;**

**import javax.ws.rs.Produces;**

**import javax.ws.rs.core.Response;**

**@Path("/json ")**

**public class JSONService {**

**@GET**

**@Path("/get")**

**@Produces("application/json")**

**public Produto getProdutoJSON() {**

**Produto produto = new Produto();**

**produto.setNome("Moto X - Android");**

**produto.setQuantidade(10);**

**return produto;**

**}**

**@POST**

**@Path("/post")**

**@Consumes("application/json")**

**public Response criarProdutoJSON(Produto produto) {**

**String result = "Product created : " + produto;**

**return Response.status(201).entity(result).build();**

**}**

**}**

**Mude o web.xml**

Get : <http://localhost:8080/rest/json/get>

Post: <http://localhost:8080/rest/json/post/>

* **Postman 🡪 post 🡪 body 🡪 raw 🡪 {"nome":"Notebook","quantidade":"10"}**

*Fonte:* [*http://www.mkyong.com/webservices/jax-rs/integrate-jackson-with-resteasy/*](http://www.mkyong.com/webservices/jax-rs/integrate-jackson-with-resteasy/)

*Criar uma raiz para o json*

*\*\* criar uma ApiResponse.java*

**package** br.edu.fcv.model;

**import** java.util.List;

**public** **class** ApiResponse {

**public** ApiResponse() {

}

**private** List<Lembrete> lembretes;

**public** List<Lembrete> getLembretes() {

**return** lembretes;

}

**public** **void** setLembretes(List<Lembrete> lembretes) {

**this**.lembretes = lembretes;

}

}

*\*\* alterar o serviço, adicionando um novo método*

@GET

@Path("/listajson")

@Produces("application/json")

**public** ApiResponse getListaLembretes(){

List<Lembrete> novaLista = **new** ArrayList<Lembrete>();

**for** (Lembrete lembrete : dao.list()){

Lembrete novoLembrete = **new** Lembrete();

novoLembrete.setId(lembrete.getId());

novoLembrete.setDescricao(lembrete.getDescricao());

novaLista.add(novoLembrete);

}

ApiResponse apiResponse = **new** ApiResponse();

apiResponse.setLembretes(novaLista);

**return** apiResponse;

}

----- android – LembreteActivity.java

\*\*adicione uma variável

private static final String JSON = "listajson/";

\*\*oncreate

Button btnJson = (Button) findViewById(R.id.btnJson);

btnJson.setOnClickListener(this);

\*\*na execução da trhead

case R.id.btnJson:

novoPath = novoPath + JSON;

break;

default:

break;

resultado = converse(HOST, 80, novoPath);

if (idBotaoSelecionado == R.id.btnJson) {

JSONObject root = new JSONObject(resultado);

}