

# Tópicos Abordados



- RESTFul WebServices
- Operações do HTTP
- Status do HTTP
- MIME Types
- WebServices com a API JAX-RS
- @Produces e@Consumes
- Path Parameters
- Query Parameters
- Cliente para RESTFul WebServices

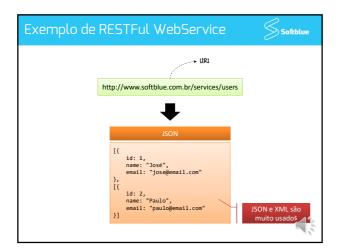


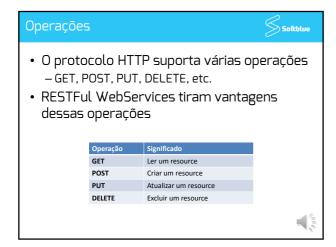
## RESTful WebServices

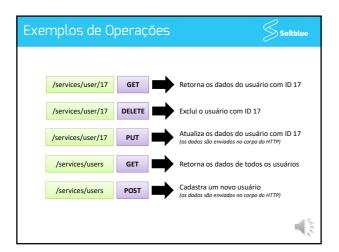


- REST é um estilo arquitetural
  - <u>Re</u>presentational <u>S</u>tate <u>T</u>ransfer
- Um WebService é visto como um recurso (*resource*)
- É identificado através de uma URI
  - <u>U</u>niform <u>R</u>esource <u>I</u>dentifier
- WebServices que usam o estilo REST são conhecidos como *RESTFul WebServices*

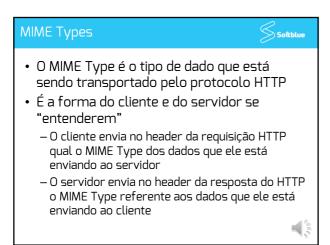


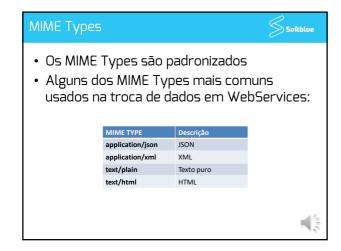






# • Toda requisição HTTP a um servidor resulta em uma resposta (*status*) Status Descrição 200 OK 201 Created Bad Request 400 403 Forbidden 404 Not Found Internal Server Error 500





# A API JAX-RS



- O JAX-RS é uma API que pertence ao Java EE
- Permite criar RESTful WebServices de forma fácil, rápida e intuitiva



# Criando Web Services com JAX-RS



• É preciso definir o caminho raiz onde os WebServices da aplicação vão responder

Define a URI base |services|\*

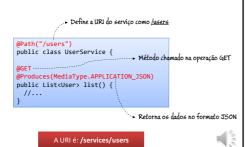
@ApplicationPath("/services")
public class JAXRSConfig extends Application {



## Criando Web Services com JAX-RS



• Uma classe Java é usada para definir o WebService









# Cliente para RESTFul Web Services • O JAX-RS permite fazer requisições a WebServices existentes a partir da interface Client User user = ClientBuilder.newClient() .target(URI) .path("15") .request(MediaType.APPLICATION\_JSON) .get(User.class);



