



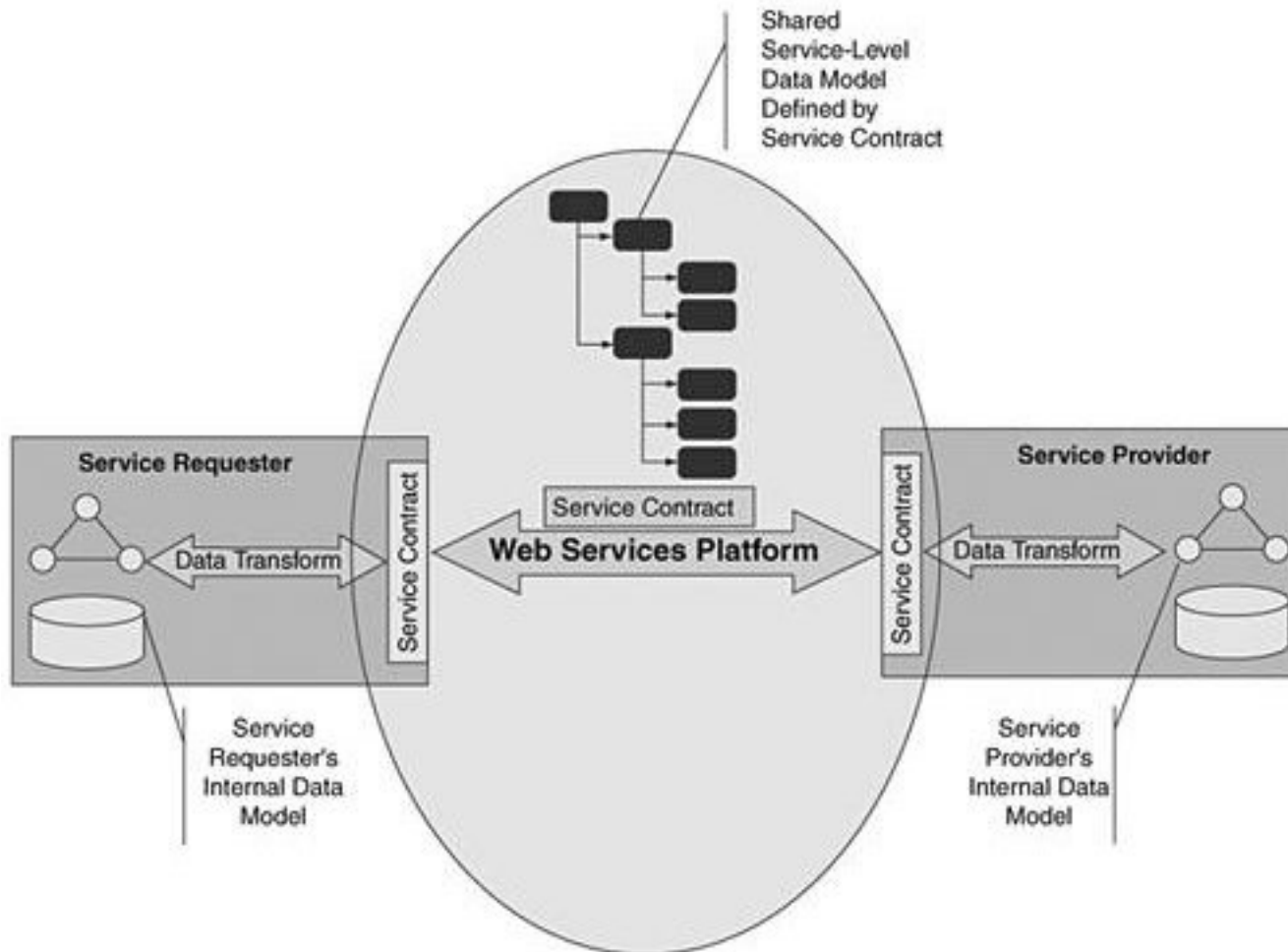
RESTful Web Services

Apresentada como tese de doutorado Roy Fielding em 2000

Conjunto de regras aplicadas à arquitetura de sistemas distribuídos

API Java JSR-311 JAX-RS

Arquitetura Distribuída



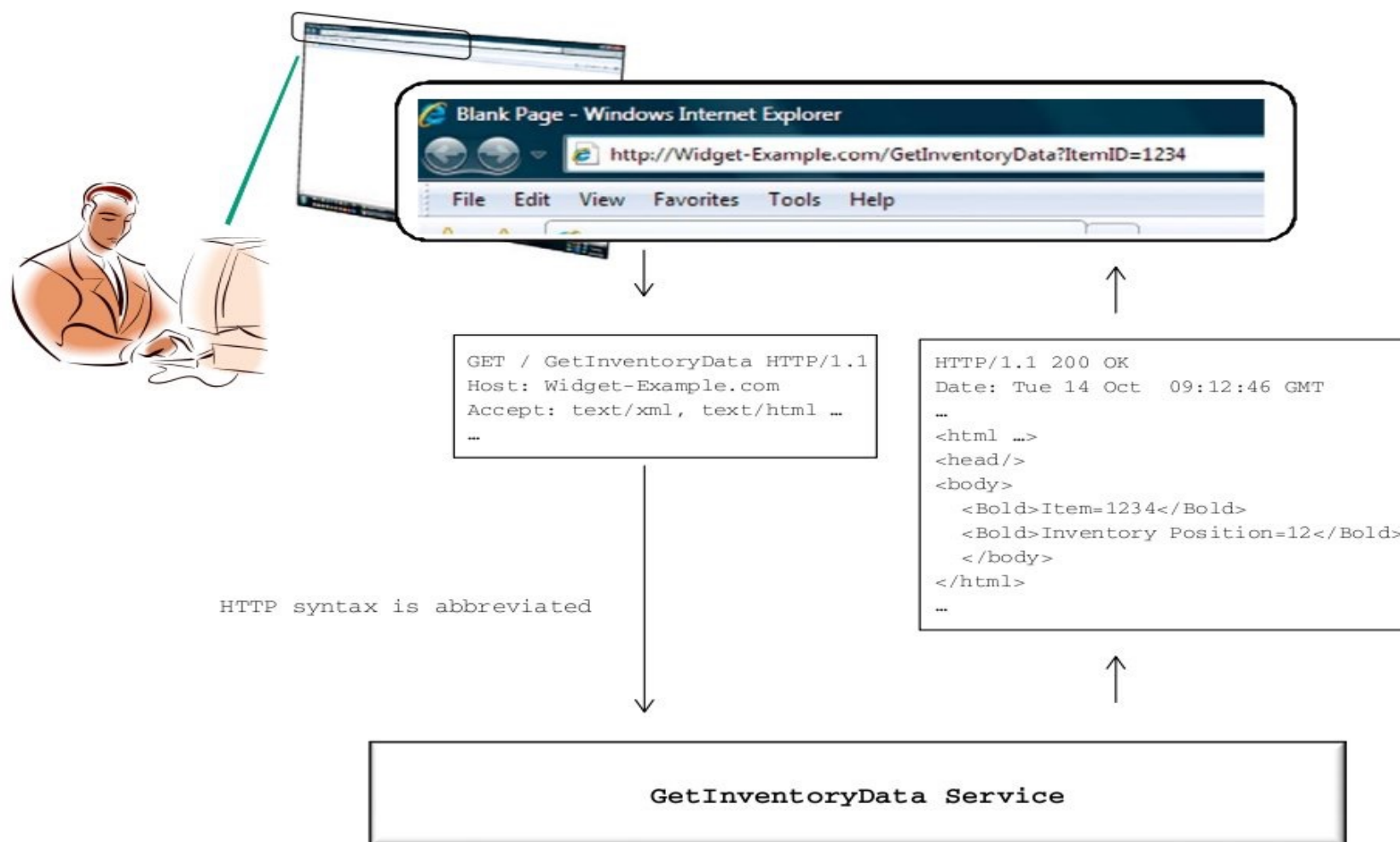
Localizável

- Cada objeto ou recurso tem um endereço único

Interface restrita e uniforme

- GET: leitura
- PUT: atualização
- DELETE: remoção
- POST: inclusão

Recurso Localizável



Declara a URI associada
a classe java

```
@Path("/bean")  
public class NomeBean {  
    private NomeDao dao;
```

Declara um método que será usado para a leitura, além de um argumento que está associado a nome, ainda os tipos das Informações que serão consumidas e produzidas

```
@GET  
@Path("/{nome}")  
@Consumes(MediaType.TEXT_PLAIN)  
@Produces(MediaType.APPLICATION_XML)  
public List<Nome> getByName(@PathParam("nome") String nome) {
```

Declara um método para
Inserção de dados

```
@POST
@Consumes(MediaType.APPLICATION_XML)
@Produces(MediaType.APPLICATION_XML)
public Nome addNome(JAXBElement<Nome> obj) {
```

Declara um método para
Exclusão de dados

```
@DELETE
@Path("/{id}")
@Consumes(MediaType.TEXT_PLAIN)
public Response delNome(@PathParam("id") int num) {
```


Configuração JBoss 5

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee" xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  id="WebApp_ID" version="2.5">
  <display-name>ProdutoWeb</display-name>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
  <servlet>
    <servlet-name>Jersey Web Application</servlet-name>
    <servlet-class>com.sun.jersey.spi.container.servlet.ServletContainer</servlet-class>
    <init-param>
      <param-name>com.sun.jersey.config.property.packages</param-name>
      <param-value>br.com.text.restWebServices</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>Jersey Web Application</servlet-name>
    <url-pattern>/rest/*</url-pattern>
  </servlet-mapping>
</web-app>
```

Nome do Servlet que Implementa JAX-RS

Nome do package contendo as classes anotadas

URI associada ao servlet

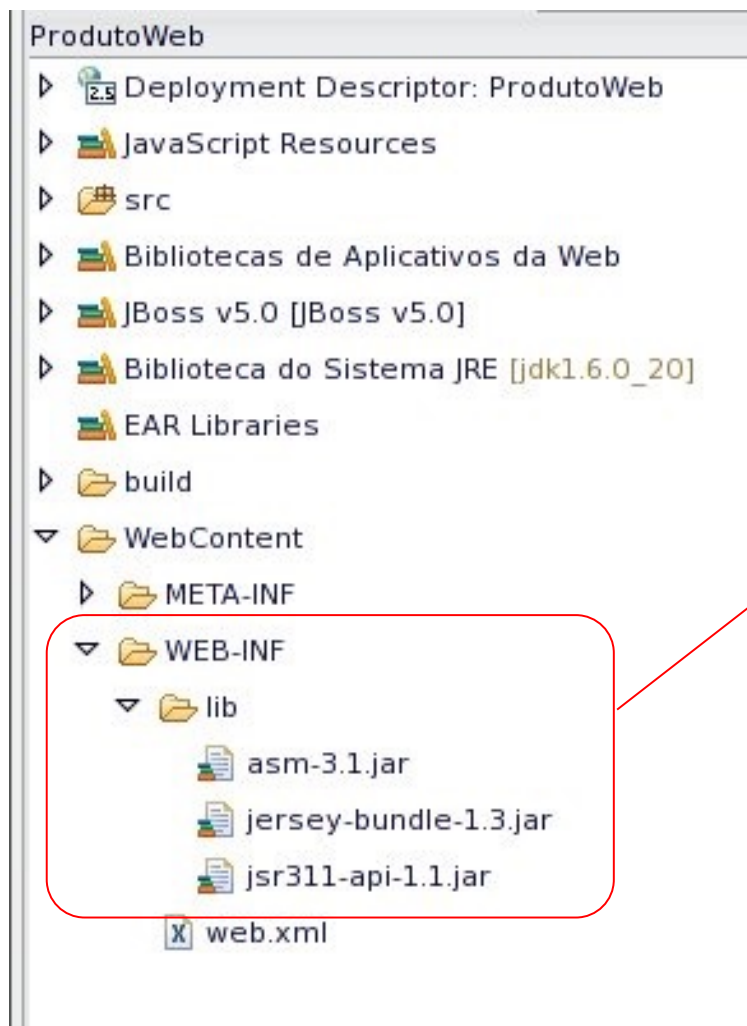
Configuração JBoss 6 e 7

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
  id="WebApp_ID" version="3.0">
  <display-name>NomeWeb</display-name>
  <servlet>
    <servlet-name>Resteasy</servlet-name>
    <servlet-class>org.jboss.resteasy.plugins.server.servlet.HttpServletDispatcher</servlet-class>
    <init-param>
      <param-name>resteasy.scan</param-name>
      <param-value>true</param-value>
    </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name>Resteasy</servlet-name>
    <url-pattern>/*</url-pattern>
  </servlet-mapping>
</web-app>
```

Nome do Servlet que Implementa JAX-RS

Habilita a pesquisa das classes anotadas

URI associada ao servlet



Local de instalar as bibliotecas da implementação do JAX-RS (utilizadas somente quando a aplicação for publicada no JBoss 5 ou Tomcat)

```
// Conecta ao RESTful Web Service
ClientConfig config = new DefaultClientConfig();
Client client = Client.create(config);
WebResource service = client.resource(
    UriBuilder.fromUri("http://127.0.0.1:8080/ProdutoWeb")
    .build());
```

URI de conexão ao
Web Service

```
// Insere um Objeto Nome via Web Service
```

```
Nome nome = new Nome();
nome.setIdade(18);
nome.setPrimeiroNome("fulano");
ClientResponse resp = service.path("rest").path("bean")
    .accept(MediaType.APPLICATION_XML).post(ClientResponse.class, nome);
```

Selecionando o tipo da resposta

Seleciona a classe

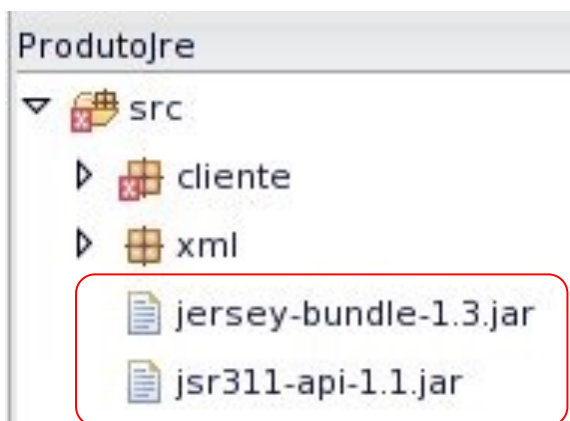
Tipo de Objeto
para a resposta

```
System.out.println(resp.getStatus());
```

Seleciona o método e a ação

Mostrando o resultado
da inserção de um objeto

Configuração no Cliente



As bibliotecas que devem ser copiadas para o projeto cliente

Devem ser colocadas no caminho de construção (CLASSPATH)

