Web Services

Prof. Hugo de Paula

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Curso de Engenharia de Software

Sumário

Serviços web – *web services*

Serviço web (*web services*)

é uma tecnologia de chamada remota de objetos que utiliza protocolos da web (p. ex. HTTP) como meio de transporte e comunicação.

Web services RESTful (*Representational State Transfer*)

utiliza URIs e métodos HTTP para disponibilizar acesso aos recursos da aplicação.

Acessando web services REST

Um serviço web REST disponibiliza dados ou recursos a partir de uma URI.

Resultado do acesso a `www.thomas-bayer.com/sqlrest/`

```
<resource xmlns:xlink="http://www.w3.org/1999/xlink">
 <CUSTOMERList xlink:href="http://(...)/sqlrest/CUSTOMER/">CUSTOMER</CUSTOMERList>
 <INVOICEList xlink:href="http://(...)/sqlrest/INVOICE/">INVOICE</INVOICEList>
 <ITEMList xlink:href="http://(...)/sqlrest/ITEM/">ITEM</ITEMList>
 <PRODUCTList xlink:href="http://(...)/sqlrest/PRODUCT/">PRODUCT</PRODUCTList>
</resource>
```

Acessando web services REST

- O exemplo indica que existem 4 web services disponíveis:
  1. **CUSTOMER**: dados de cliente.
  2. **INVOICE**: dados do pedido.
  3. **ITEM**: dados do item do pedido.
  4. **PRODUCT**: dados do produto.

Acessando web services REST

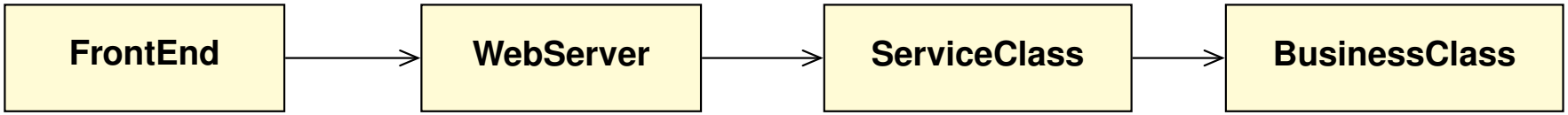
Para acessar um serviço web de clientes:
`http://www.thomas-bayer.com/sqlrest/CUSTOMER/`.

- Aparecerá a lista de clientes.

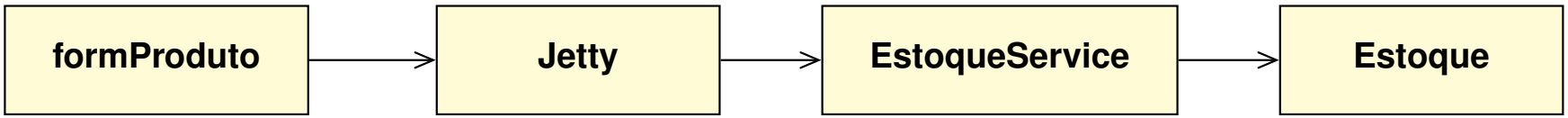
Acesse o cliente 10:
`http://www.thomas-bayer.com/sqlrest/CUSTOMER/10/`

```
<CUSTOMER xmlns:xlink="http://www.w3.org/1999/xlink">
   <ID>10</ID>
   <FIRSTNAME>Sue</FIRSTNAME>
   <LASTNAME>Fuller</LASTNAME>
   <STREET>135 Upland Pl.</STREET>
   <CITY>Dallas</CITY>
</CUSTOMER>
```

Decomposição de uma aplicação em serviços REST



Exemplo: Controle de estoque



Spark Framework

Spark framework

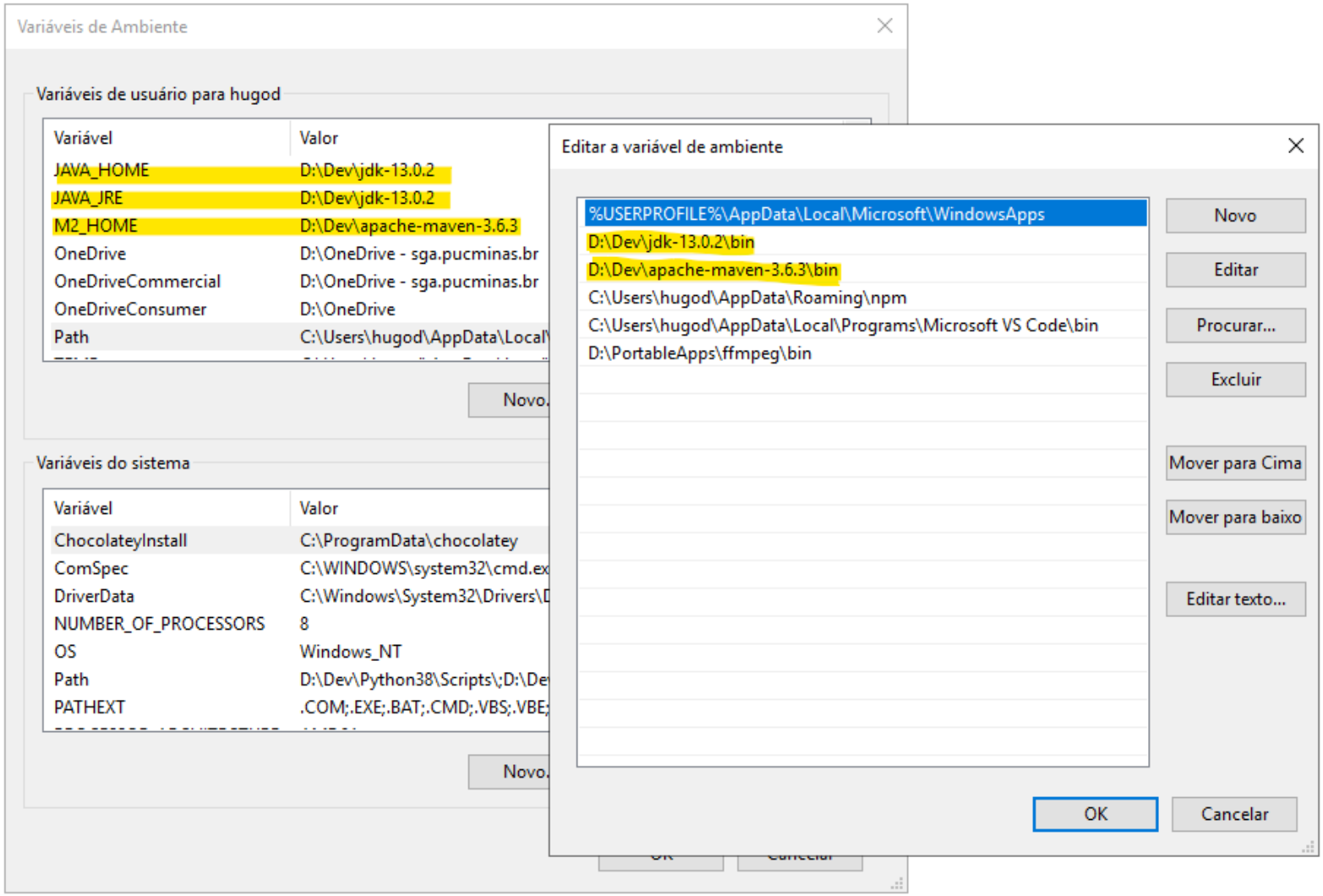
é um micro framework para a criação de aplicações web a partir do Java 8 (ou Kotlin). Disponível em: `http://sparkjava.com/`.

Para utilizar o framework em seu projeto você deve utilizar o gerenciador de dependências chamado Maven (`https://maven.apache.org/`).

Configuração do Maven

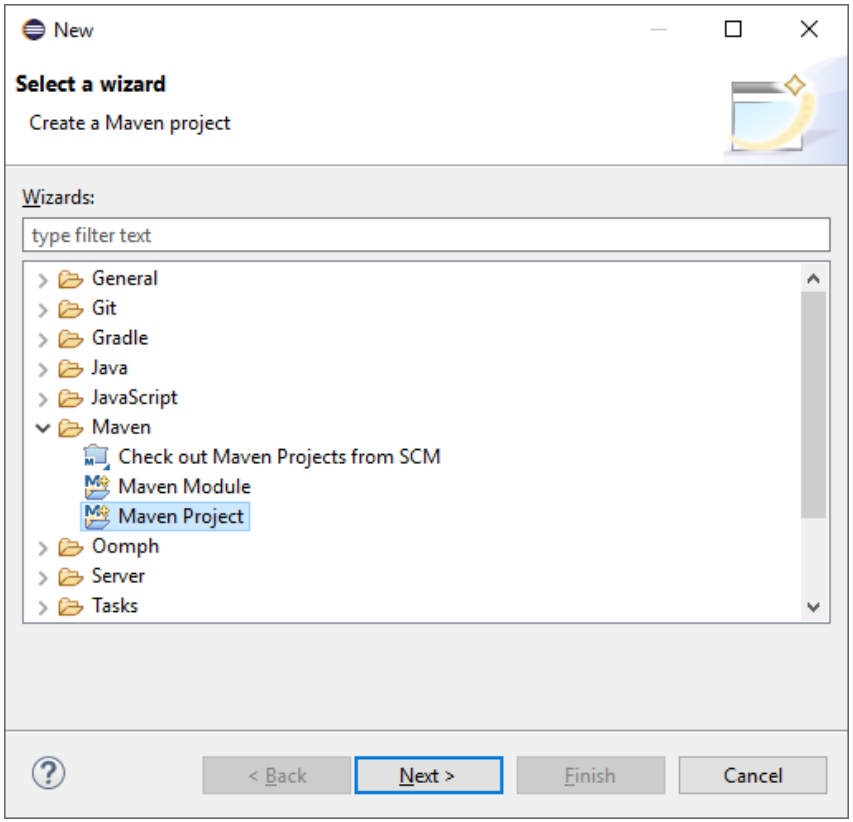
- Maven é uma ferramenta de gerenciamento de projeto.
- Controla configuração e dependências.
- Instalação:
  `https://maven.apache.org/download.cgi`
  - Baixar e descomprimir o arquivo zip.
  - Adicionar o diretório **bin** ao **PATH**.
- É necessário que a variável **JAVA_HOME** esteja configurada.

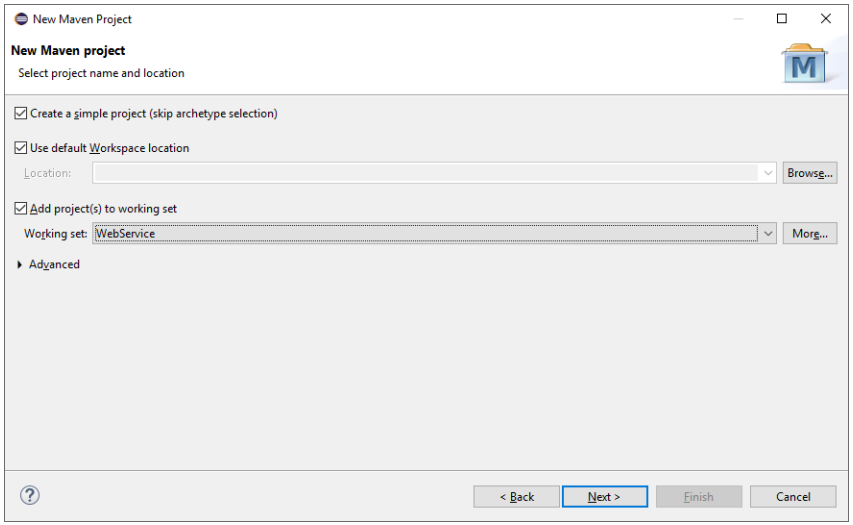
Variáveis de ambiente



Criando um projeto Maven

- No Eclipse: `New` → `Maven Project`





Configuração do projeto

- **archetypeArtifactId**:Especifica o tipo do projeto a ser criado.
- **archetypeGroupId**: Especifica o grupo em que o arquétipo está definido.
- **groupId**: Usado para identificar unicamente o projeto. Normalmente baseado no domínio. Usado como o nome do pacote.
- **artifactId**: O nome do seu projeto.

Configuração do projeto: Arquivo POM.XML

```
<properties>
    <java.version>13</java.version>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
</properties>


<dependencies>
    <dependency>
        <groupId>com.sparkjava</groupId>
        <artifactId>spark-core</artifactId>
        <version>2.8.0</version>
    </dependency>
    <dependency>
        <groupId>org.slf4j</groupId>
        <artifactId>slf4j-simple</artifactId>
        <version>1.7.21</version>
    </dependency>
</dependencies>
```

Métodos HTTP

```
public static void main(String[] args) {
    port(6789);
    get("/hello", (request, response) -> "<h1>Nosso primeiro Web Server!</h1>");

    post("/hello", (request, response) -> "Web Server recebeu: " + request.body());

    get("/private", (request, response) -> { response.status(401);
                                             return "Área privada!!!"; });

    get("/users/:name", (request, response) -> "Usuário selecionado: " +
                                               request.params(":name"));

    get("/news/:section", (request, response) -> { response.type("text/xml");
                            return "<?xml version=\"1.0\" encoding=\"UTF-8\"?><news>" +
                                   request.params("section") + "</news>";
    });

    get("/protected", (request, response) -> {
        halt(403, "I don't think so!!!");
        return null; });

    get("/redirect", (request, response) -> { response.redirect("/news/world");
                                              return null; });

    get("/", (request, response) -> "root");
}
```

Roteamento de requisições

- Uma rota é formada por três elementos:
  - verbo: get, post, put, delete, head, trace, connect, options
  - caminho: hello, /users/:name
  - *callback*: (request, response) -> { }

```
get("/", (request, response) -> {          // Exibe alguma coisa
});

post("/", (request, response) -> {         // Cria alguma coisa
});

put("/", (request, response) -> {          // Atualiza alguma coisa
});

delete("/", (request, response) -> {       // Apaga alguma coisa
});

options("/", (request, response) -> {      // Verifica alguma coisa
});
```

Roteamento de requisições: CRUD

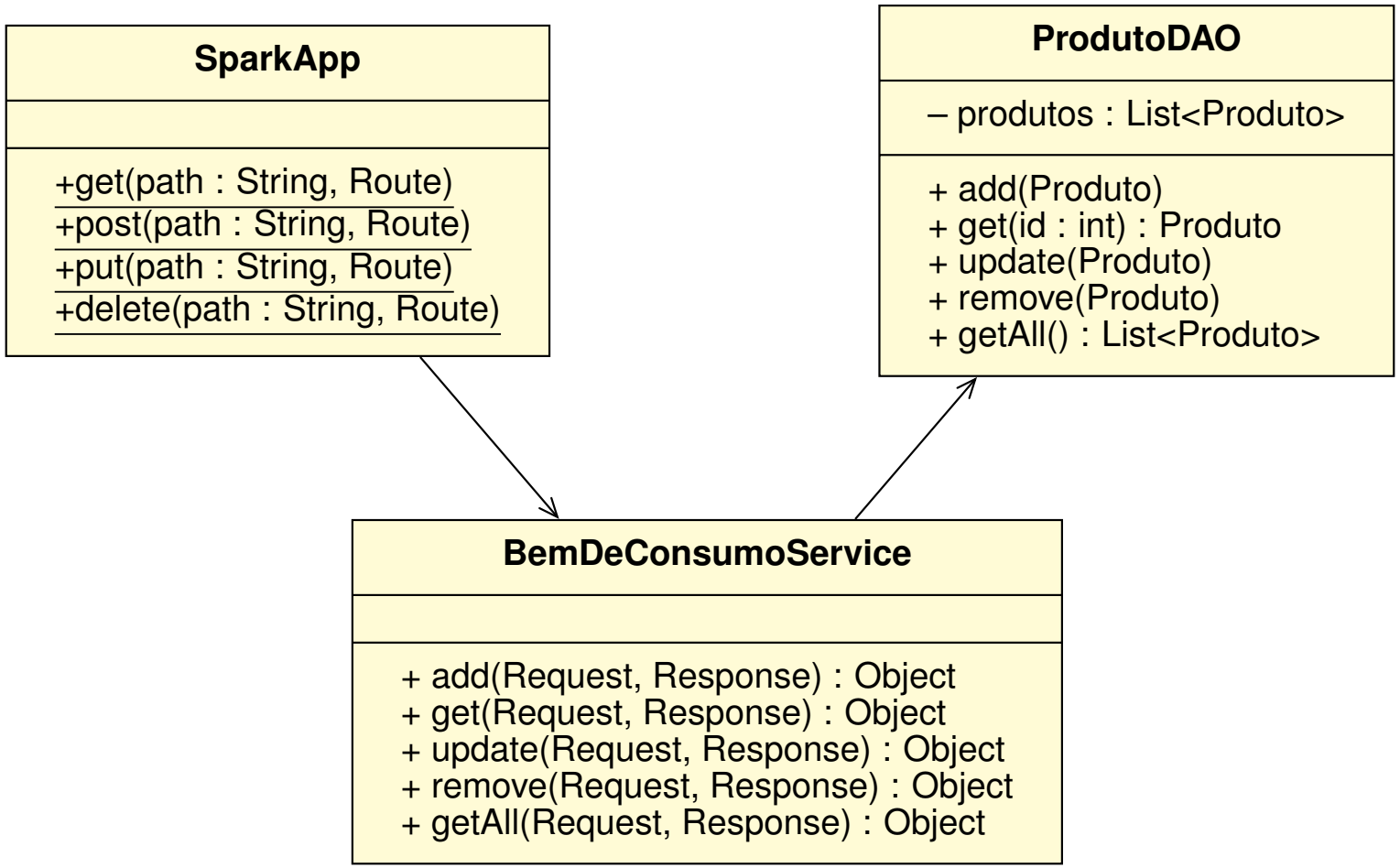
- Importante definir uma tabela de rotas antes de começar a codificação.
- Uma tabela padrão de CRUD (criar, consultar, atualizar e apagar) segue o seguinte padrão:

verbo	**path**	**descrição**
post	/produto	adiciona um novo produto no Estoque
get	/produto/:id	recupera os dados do produto :id
put	/produto/:id	atualiza os dados do produto :id
delete	/produto/:id	remove o produto :id
get	/produtos	recupera toda a lista de produtos

Disponibilização de arquivos estáticos

- Os arquivos de *front-end* (.html) são considerados arquivos estáticos.
- Eles devem ser disponibilizados para poderem ser acessados a partir do servidor.
- O comando `staticFileLocation("/public");` avisa que os arquivos estáticos estarão disponíveis em uma pasta `public`, a partir da base do projeto. NO caso do Maven, a base do projeto é o diretório `"/main/resources"`.

Disponibilização do serviço na Web



JavaScript Object Notation

JavaScript Object Notation – JSON

É um formato leve para troca de informação. Baseado na linguagem JavaScript, é fácil de ser entendido pelo ser humano e pela máquina.

Baseado em duas estruturas:

- `JSONObject`: uma coleção de `pares <nome, valor>`, tais como um dicionário ou Map, em Java.
- `JSONArray`: uma lista ordenada de valores, ou `Array`.

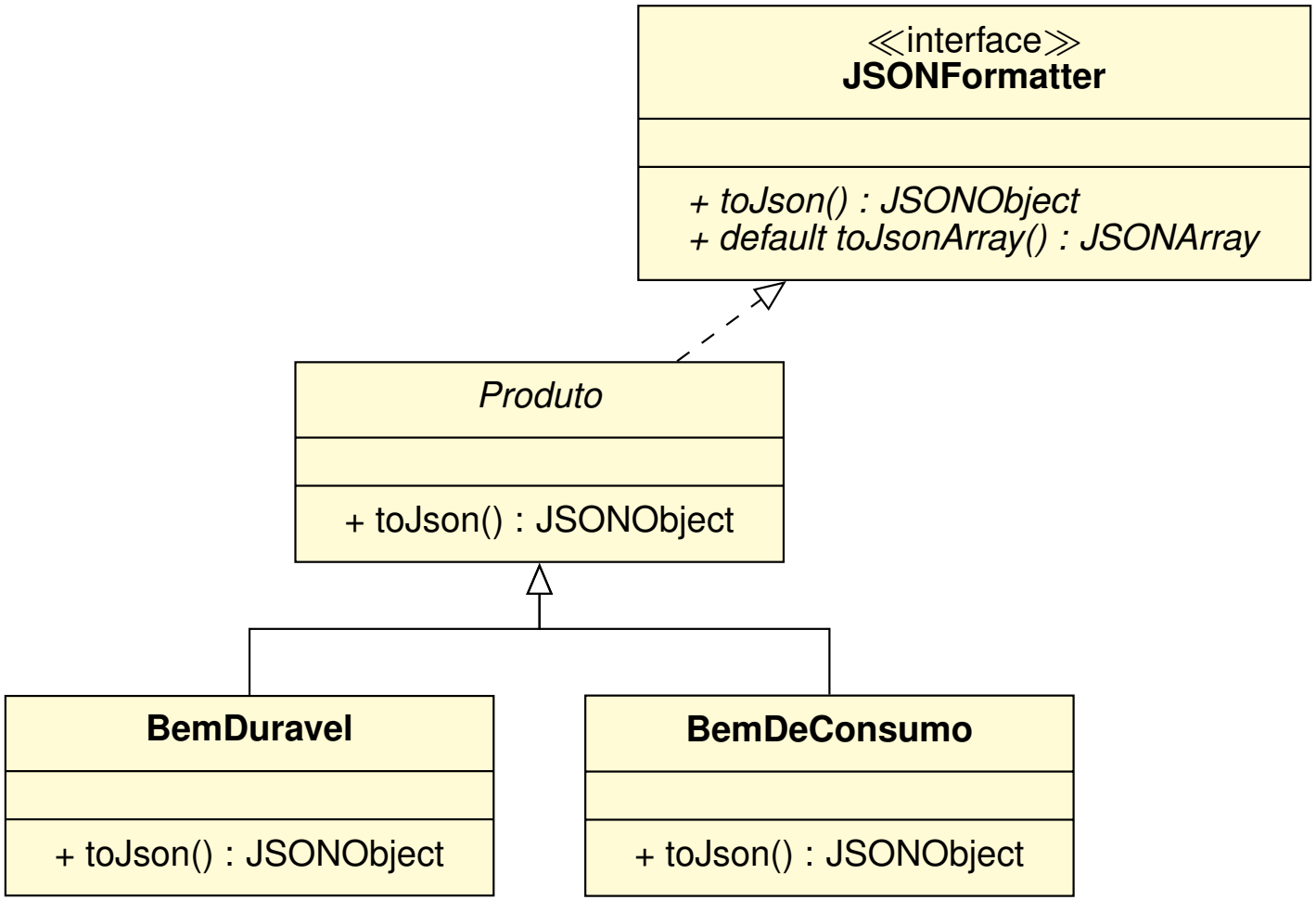
Mapeamento entre JSON e entidades Java

JSON	Java
`string`	java.lang.String
`number`	java.lang.Number
`true|false`	java.lang.Boolean
`null`	**null**
`array`	java.util.List
`object`	java.util.Map

```
JSONObject obj = new JSONObject();

obj.put("string", "Shulambs");
obj.put("inteiro", new Integer(1));
obj.put("real", new Double(1.99));
obj.put(logico", new Boolean(true));
```

Exemplo: Convertendo objetos Java para Json



Redirecionamento

- Um redirecionamento ocorre quando uma requisição é redirecionada para outro endereço.
- Isto é útil quando Se deseja criar um fluxo de páginas entre as requisições.
- Por exemplo, se um produto não for encontrado, pode-se desejar redirecionar a requisição para uma tela de erro, ou voltar para a tela de cadastro.
- No Spark Java, isso é feito pelo código:

  response.redirect("/novo");.
- Existem outras formas de redirecionamento baseado em métodos HTTP. Veja a documentação para mais detalhes.