

## Gerando Json com Gson

Até agora vimos como gerar código XML, mas e se eu quisesse trabalhar com JSON? Para não deixar um buraco aberto aqui, ao invés de gerar `APPLICATION_XML`, geraremos `APPLICATION_JSON` através do uso de outra biblioteca:

```
@Path("/{id}")
@GET
@Produces(MediaType.APPLICATION_JSON)
public String busca(@PathParam("id") long id) {
    Carrinho carrinho = new CarrinhoDAO().busca(id);
    return carrinho.toXML();
}
```

Ao invés de chamar o método `toXML` invocarei o método `toJson`, alteramos a chamada:

```
return carrinho.toJson();
```

E criamos o método:

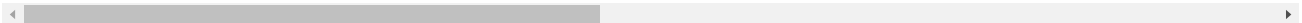
```
public String toJson() {
    return null;
}
```

Dentro dele, assim como fizemos `new XStream`, vou usar a biblioteca `gson` da google:

```
public String toJson() {
    return new Gson().toJson(this);
}
```

Vamos testar agora e ver o resultado? Levantamos o servidor rodando o `Servidor` e acessamos a URI `http://localhost:8080/carrinhos/1`. Legal, ele trouxe para mim um `Json` ao invés do código XML:

```
{"produtos":[{"preco":4000.0,"id":6237,"nome":"Videogame 4","quantidade":1}, {"preco":60.0,"id"::
```



Repare como é muito simples dizer se estou devolvendo um formato ou outro. Mais para a frente veremos como suportar dois formatos ao mesmo tempo, mas como nesse instante utilizaremos somente XML, voltamos atrás em nossa mudança, mesmo que não removendo o método `toJson`:

```
@Path("/{id}")
@GET
@Produces(MediaType.APPLICATION_XML)
public String busca(@PathParam("id") long id) {
    Carrinho carrinho = new CarrinhoDAO().busca(id);
}
```

```
    return carrinho.toXML();  
}
```

Na prática entenda se você deseja trabalhar com um media type e formato ou outro. O importante neste capítulo é entender que temos o poder de usar a biblioteca de serialização que bem entendermos, seja XStream, JAXB, Gson etc. Vamos aos exercícios?