

Deploy WAR e Servlet Contêiners

Nosso próximo passo é pegar essa nossa aplicação que a gente levanta hoje rodando o servidor Grizzly e rodar dentro de um servlet container, dentro de um container que suporta Java EE e servlet API. Isso pode ser um Tomcat, um Jetty, um JBoss ou qualquer outra implementação que suporte servlet container. Vamos lá? Então, para isso, a gente vai pegar o nosso projeto, que é o projeto que eu tenho aqui, e vamos ter que zipar um diretório em um arquivo .war, que é uma aplicação web, uma aplicação web que o Java EE utiliza, o arquivo .war.

Repara que dentro do nosso diretório `src/main`, a gente tem aqui o diretório `webapp`. Dentro do diretório `webapp` a gente tem o `index.html`, que é uma página bem simples, e a gente tem o `WEB-INF`, que tem o `web.xml`.

No `web.xml`, já está configurado o Jersey:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- This web.xml file is not required when using Servlet 3.0 container,
     see implementation details http://jersey.java.net/nonav/documentation/latest/jax-rs.html -->
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <servlet>
        <servlet-name>Jersey Web Application</servlet-name>
        <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
        <init-param>
            <param-name>jersey.config.server.provider.packages</param-name>
            <param-value>br.com.alura.loja.resource</param-value>
        </init-param>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>Jersey Web Application</servlet-name>
        <url-pattern>/webapi/*</url-pattern>
    </servlet-mapping>
</web-app>
```

E o Jersey já está configurado falando: "Olha, procura o diretório, o pacote `br.com.alura.loja.resource` dentro desse pacote, Jersey, você vai encontrar as classes do meu projeto, o jaxrs que eu implementei, os meus resources que eu implementei estão dentro desse pacote, então, por favor, procure aqui dentro".

Aqui, se você usa um outro pacote pro seu projeto, utilize outro pacote, não tem problema. Tudo bem servlet do Jersey, você lerá os recursos desse pacote. Mas qual a URI que você vai acessar? É o mapeamento da servlet, né? No mapeamento da servlet tá mapeando `/webapi/` qualquer coisa. Então pra eu acessar aquele `/carrinhos/1`, vira então `/webapi/carrinhos/1`.

Então na minha aplicação eu vou poder ter a servlet do Jersey servindo os meus serviços REST, e posso ter outros servlets servindo outras coisas, não tem problema nenhum, é compatível. Então aqui eu vou acessar `/webapi/carrinhos/1`. Legal. Agora eu quero pegar isso tudo então e zipar num arquivo war.

Vou lá no meu projeto, que é esse diretório aqui que eu tenho workspace loja. Dentro do diretório `src/main/webapp`, eu tenho o meu `index` e o `WEB-INF`. O `WEB-INF` já tá todo configurado, porque o Eclipse já cospe as minhas classes no diretório `classes`. As bibliotecas já estão no diretório `lib`, então é isso aqui que eu quero compactar: o `index` e o `WEB-INF`. Esses dois caras eu quero clicar na direita, comprimir em um arquivo `.zip`



Comprimi um arquivo .zip, vou chamar esse arquivo .zip de `loja.war`, que é o padrão webarchive. Ele é um arquivo zip, certo? Mas é um webarchive.

Legal, agora eu vou baixar o Tomcat. Vou baixar aqui o Tomcat 8. Eu clico aqui em download versão tar.gz pro meu Linux.

Baixei o meu Tomcat, agora eu vou pegar esse meu Tomcat e descompactar. Dentro do diretório do Tomcat, eu tenho o diretório bin, onde eu levanto e destruo o Tomcat, e tenho o diretório webapps, onde ficam minhas aplicações web. Bom, se minhas aplicações web ficam lá dentro, vou jogar o `loja.war` aqui dentro.

Legal, agora vou no terminal levantar o Tomcat. Vou entrar no diretório Downloads :

```
cd Downloads/apache-tomcat-8.0.0-RC5
cd bin
```

e vou executar o startup:

```
./startup.sh
```

Executei o startup, deve estar levantando, né? Vamos ver, olha, ele levanta, tanto que ele já descompactou o arquivo `loja.war`. Legal.

Onde é que fica o arquivo de log do Tomcat? No diretório logs. Aqui a gente tem o catalina.out, que mostra o log do Tomcat.

Legal. Vou aqui no navegador e vou acessar agora `localhost:8080/carrinhos`? Não! Qual que é o nome do meu contexto? O nome do meu contexto é o nome da minha aplicação web. A aplicação web se chama loja. Se lembra que ela chama `loja.war`? Então a aplicação web chama loja.

```
http://localhost:8080/loja
```

Qual que foi a configuração que eu fiz do JAX-RS? Não foi webapi? Então

```
http://localhost:8080/loja/webapi
```

Agora sim eu tô dentro do JAX-RS. Legal, tô dentro do JAX-RS, quê que eu quero acessar mesmo? `carrinhos/1`?

```
http://localhost:8080/loja/webapi/carrinhos/1
```

Acessei o meu carrinho. Então é bem simples eu deployar essa mesma aplicação num servlet container. Basta eu ter um servlet container, um Tomcat, um Jetty, seja lá quem for, e pegar a minha aplicação, isso é, a estrutura da aplicação com

WEB-INF, lib, classes e o web.xml configurado e compactar tudo isso. Isso tudo vai pra produção.

É claro, eu tenho que concatenar o path que eu colocar no meu web.xml. E você pode falar: "Ah Guilherme, eu não queria /webapi/*, eu queria tudo. Não quero ter que ficar digitando /webapi".

Então é só tirar daqui o /webapi. Salvei, vou gerar agora o meu arquivo .war denovo, então aqui eu vou chamar de loja.war, vou remover o loja.war antigo e vou esperar o Tomcat perceber que eu removí, aí ele vai remover o contexto do Tomcat.

Legal, removeu? Agora eu copio o meu loja.war de novo. Agora eu já estou sem o webapi. Assim que ele descompactar, eu vou pro navegador e testo. Vou no navegador, tiro o webapi

`http://localhost:8080/loja/carrinhos/1`

Testo só /loja/carrinhos/1. Bonito, tá funcionando. A configuração do contexto /loja, ou raiz etc, aí são configurações do próprio Tomcat. Então, como a gente viu, dentro do diretório lib, do nosso Jersey, da aplicação do Jersey, a gente já deixou as bibliotecas que suportam tanto o Jersey com o Grizzly quanto o Jersey com o Tomcat com o servlet api.

Vamos pros exercícios?