07. Deploy no Jetty

Date	@24/11/2022
Categoria	Java para Web
Curso	Java Servlet: autenticação autorização e o padrão MVC

Tópicos

- Projeto da aula anterior
- Preparando o ambiente
- Servlet Container Jetty
- Observando as diferenças
- Especificação das Servlets
- Funcionamento de um Servlet Container
- Para saber mais: Servlet x HttpServlet
- Faça como eu fiz na aula
- Projeto completo
- O que aprendemos?
- Conclusão

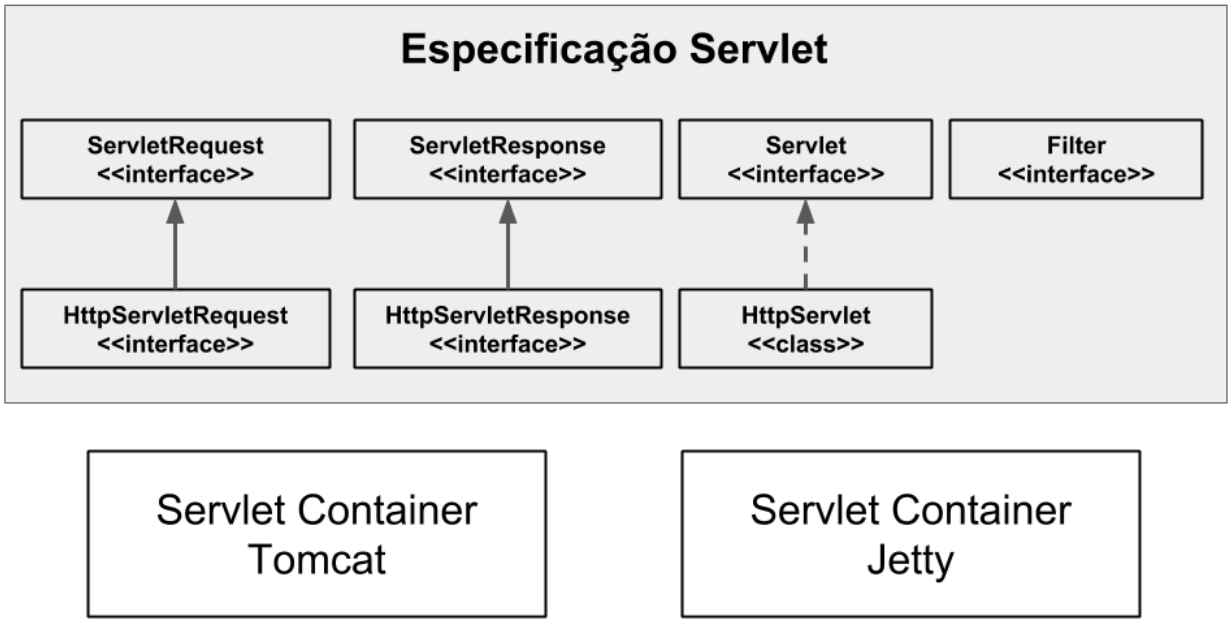
Observando as diferenças

Vimos na aula que existem dois tipos de servidores, os ***Servlet Container*** (*Servlet engine*) e os ***Application Servers***. Podemos realizar o *deploy* da nossa aplicação em servidores desses dois tipos, mas existe uma diferença entre eles. Marque abaixo a alternativa que informa corretamente a diferença entre essas duas opções para *deploy*:

- Os *Application Servers*, além de mapear requisições, são muito mais robustos e oferecem um número maior de recursos, diferentemente dos *Servlet Containers*, que apenas mapeiam as requisições do cliente.

Para saber mais: Servlet x HttpServlet

Falamos sobre a especificação Servlet, e assim a garantia que podemos rodar o nosso projeto em diferentes *Servlet Containers*. Apresentei a imagem abaixo, que mostra alguns tipos importantes da especificação, como `javax.servlet.Servlet`, `javax.servlet.http.HttpServlet`, `javax.servlet.ServletRequest`, `javax.servlet.http.HttpServletRequest`, entre outros:



Repare que existem tipos mais genéricos e tipos mais específicos, focados no protocolo HTTP, por exemplo:

- `javax.servlet.Servlet` --> `javax.servlet.http.HttpServlet`
- `javax.servlet.ServletRequest` --> `javax.servlet.http.HttpServletRequest`
- `javax.servlet.ServletResponse` --> `javax.servlet.http.HttpServletResponse`

Tirando a interface `Filter`, sempre existe um tipo mais especifico do mundo HTTP. Por quê?

A ideia inicial era que o mundo Servlets suportasse outros protocolos como FTP ou SMTP. Ou seja, as servlets e os servlet containers poderiam trabalhar com outros protocolos além do HTTP. Por isso existem essas interfaces genéricas (sem `Http` no nome), para estender e atender novos protocolos.

Por exemplo, poderia existir um *FTP Servlet Container*, que atenderia o protocolo FTP e assim estender os tipos genéricos, para criar um `FtpServlet`, ou `FtpServletRequest`. No final, não existem essas implementações, e o protocolo HTTP é o único que as servlets atendem. Isso também se dá por causa da onipresença do protocolo HTTP no dia a dia, e da baixa relevância dos outros protocolos.

O que aprendemos?

Nesta aula, aprendemos:

- A disponibilizar a nossa aplicação no *servlet container* **Jetty**
- Que Servlet é uma especificação
- Que a especificação Servlet faz parte do Java EE/Jakarta EE
- Que, ao usar Servlet, programamos independentemente do servidor/container
- A diferença entre *servlet container* e *application server*
