

# Aula 3 Servlet

Disciplina: Tecnologia da Web

Professor: Flavio Barbieri Gonzaga



# Funcionamento de um Servlet

- O ciclo de vida de um servlet é determinado por três de seus métodos:
  - init
  - service
  - destroy

# Funcionamento de um Servlet

- Ao receber a primeira solicitação, o servlet é carregado para a memória.
- Antes de poder atender a qualquer requisição, o método `init` é executado.

## Funcionamento de um Servlet

- Todas as solicitações são tratadas pelo método service do servlet.
- O método service é chamado uma vez para cada requisição.

## Funcionamento de um Servlet

- O método destroy é chamado quando o contêiner de servlets decide tirá-lo de execução (por exemplo, para liberar memória).

# Métodos da Interface Servlet

- void init(ServletConfig config)
  - Através do argumento config, podemos receber configurações iniciais (definidas no arquivo web.xml) para o servlet.
- void service(ServletRequest request, ServletResponse response)
  - Contêiner servlet chama esse método para responder a uma solicitação do cliente para o servlet.

# Métodos da Interface Servlet

- `void destroy()`
  - Método chamado quando um servlet é encerrado pelo contêiner.
- `ServletConfig getServletConfig()`
  - Retorna uma referência para o objeto que implementa a interface `ServletConfig`.
- `String getServletInfo()`
  - Pode ser definido pelo programador para retornar uma string contendo informações como nome do autor e a versão do Servlet.

# Demonstração do ciclo de vida de um Servlet

```
import javax.servlet.*;
import java.io.IOException;

public class Basico implements Servlet{

    public void init(ServletConfig config) throws ServletException {
        System.out.println("metodo init executado");
    }

    public void service(ServletRequest request, ServletResponse response)
        throws ServletException, IOException {
        System.out.println("metodo service executado");
    }

    public void destroy() {
        System.out.println("metodo destroy executado");
    }

    public String getServletInfo() {
        return null;
    }

    public ServletConfig getServletConfig() {
        return null;
    }
}
```

# Entendendo o código

Importando as classes do pacote javax.servlet e a classe java.io.IOException

```
import javax.servlet.*;  
import java.io.IOException;
```

- Sempre que trabalharmos com Servlets, é interessante importarmos o pacote javax.servlet para evitarmos possíveis erros.
- A classe java.io.IOException irá tratar erros durante o processamento do fluxo (lendo ou gravando dados no cliente).

# Entendendo o código

Classe declarada com o nome Basico, implementando a classe javax.servlet.Servlet

```
public class Basico implements Servlet{
```

- Todo servlet deve implementar a classe Servlet (direta ou indiretamente).
- Não se esqueça de que o nome da classe aqui declarada como Basico precisa ser o mesmo do arquivo. No exemplo, Basico.java

# Entendendo o código

```
public class Basico implements Servlet{  
  
    public void init(ServletConfig config) throws ServletException {  
        System.out.println("metodo init executado");  
    }  
  
    public void service(ServletRequest request, ServletResponse response)  
        throws ServletException, IOException {  
        System.out.println("metodo service executado");  
    }  
  
    public void destroy() {  
        System.out.println("metodo destroy executado");  
    }  
  
    public String getServletInfo() {  
        return null;  
    }  
  
    public ServletConfig getServletConfig() {  
        return null;  
    }  
  
}
```

Implementação dos métodos descritos anteriormente. As linhas System.out.println() exibirão a mensagem no prompt de comando quando executadas.

# Configurando o arquivo web.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app
PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
  <servlet>
    <servlet-name>Basico</servlet-name>
    <servlet-class>Basico</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>Basico</servlet-name>
    <url-pattern>/Aula/Basico</url-pattern>
  </servlet-mapping>
</web-app>
```

- A configuração do arquivo web.xml deverá ficar semelhante à esta. Para lembrar a função das tags, recorra à aula 2.

# Executando o Servlet

```
22/05/2006 01:50:44 org.apache.catalina.startup.Catalina start  
INFO: Server startup in 15766 ms  
metodo init executado  
metodo service executado
```

- Ao ser executado pela primeira vez, o servlet escreverá as frases “metodo init executado” e “metodo service executado”.

## Executando o Servlet

```
22/05/2006 01:50:44 org.apache.catalina.startup.Catalina start  
INFO: Server startup in 15766 ms  
metodo init executado  
metodo service executado  
metodo service executado  
metodo service executado  
metodo service executado  
metodo service executado  
metodo service executado  
metodo service executado  
metodo service executado  
metodo service executado
```

- As demais execuções, quando solicitadas, escreverão apenas “metodo service executado”, mostrando que o método init é executado apenas ao se carregar o servlet.

# Incômodo ao implementar a classe `javax.servlet.Servlet`

- Ao se criar classes que implementam a classe `javax.servlet.Servlet` existe um incômodo:
  - Você precisa oferecer todos os métodos da interface `Servlet`, mesmo que só use o método `service`.

## Resolvendo o incômodo

- Para se resolver essa questão, os pacotes de servlets definem duas classes abstract que implementam a interface Servlet
  - Classe GenericServlet (pacote javax.servlet)
  - Classe HttpServlet (pacote javax.servlet.http)

## Próximas aulas

- Nas próximas aulas estaremos apresentando e desenvolvendo servlets utilizando a classe `HttpServlet`.
- Optamos pela utilização da classe `HttpServlet` pelo fato de a mesma ser mais utilizada do que a classe `Generic Servlet`.

# Trabalho

- Parte 1:
  - Leia e faça um resumo das páginas 15 à 34 do livro Java para Web com Servlets, JSP e EJB.
  - A ser entregue em forma de trabalho escrito (a apresentação do conteúdo escrito também será avaliada).
- Parte 2:
  - Coloque o Código Config funcionando no Tomcat.
  - Entenda o mesmo, bem como o arquivo web.xml que o acompanha.

# Trabalho

- Parte 1:
  - A ser entregue no dia 29/05/2006 (não sendo tolerado atraso).
- Parte 2:
  - Cada aluno deverá explicar o funcionamento do código, dando ênfase em:
    - void init(ServletConfig config): parâmetro config do método init
    - ServletContext contextoServlet: objeto do tipo ServletContext

# Trabalho

- Considerações finais:
  - Trabalho Individual;
  - Fontes utilizadas devem constar na Bibliografia;
  - Não serão toleradas Cópias.

## Bibliografia

- Java para a Web com Servlets, JSP e EJB – 1ª Edição. Budi Kurniawan – Editora Ciência Moderna
- Java como Programar – 6ª Edição. H. M. Deitel, P. J. Deitel – Editora Pearson Prentice Hall
- [www.argonavis.com.br](http://www.argonavis.com.br) (Helder da Rocha)