Servlets — Notas de aula

Flávio Velloso Laper

1 de setembro de 2020

Servlets

- Extensões de servidor escritas em Java.
- Podem ser usados para estender qualquer tipo de aplicação do modelo requisição/resposta.
- Todo servlet implementa a interface javax. servlet .Servlet; (tipicamente estende GenericServlet.)

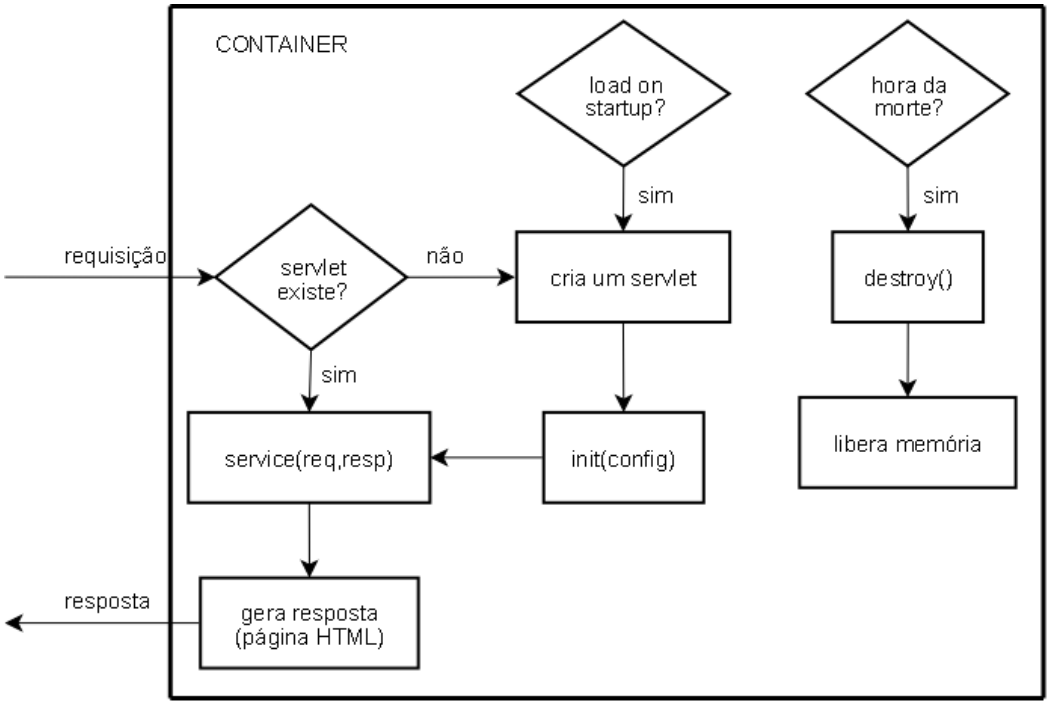
Servlets http

- Extensões para servidores web.
- Estendem javax. servlet .http.Http.Servlet.
- Lidam com características típicas de HTTP, como métodos GET, POST, sessões, etc.

Classes e interfaces importantes:

- Interfaces:
  - HttpServletRequest: estende ServletRequest e modela uma requisição.
  - HttpSevletResponse: estende ServletResponse e modela uma resposta.
  - HttpSession: guarda informações de estado de um usuário.
- Classe abstrata:
  - HttpServlet: é a principal classe estendida para criar servlets HTTP.

Ciclo de vida de um servlet



- O ciclo de vida é controlado pelo container.
- Quando o servidor recebe uma requisição, ela é repassada para o container que a delega a um servlet.
- O container:
  - Carrega a classe na memória.
  - Cria uma instância de Servlet.
  - Inicializa a instância chamando init().
- Depois que o servlet foi inicializado, cada requisição é executada em um método service() (que recebe objetos para a requisição e para a resposta).
- Quando o container decide remover o servlet da memória, ele o finaliza chamando destroy().

Inicialização (Método init())

- Tarefa realizada uma vez quando o servlet é criado.
- Deve-se sobrepor init(config) com instruções que serão realizadas uma vez na vida do servlet:
  - Carregar parâmetros de inicialização, dados de configuração.
  - Obter recursos.
  - Uma falha na inicialização deve provocar UnavailableException.
- Exemplo:

```
1 private Servletconfig config;
2 public void init(Servletconfig config) throws ServletException {
3   this.config = config;
4   String maximo = config.getInitParameter("maximo");
5   if(maximo == null)
6     throw new UnavailableException("Configuração incorreta");
7   ————
8 }
```

Observação: o parâmetro *máximo* do trecho de código acima deve ser configurado no arquivo *web.xml* com as seguintes tags:

```
<context-param>
   <param-name>maximo</param-name>
   <param-value>5</param-value>
</context-param>
```

Finalização (Método destroy())

- Quando o container decide remover um servlet da memória, ele chama destroy().
- destroy() é usado para liberar recursos e realizar outras tarefas de "limpeza".
- Exemplo:

```
public void destroy () {
   banco.close();
   banco = null;
}
```

Métodos de serviço

- São métodos que implementam operações de tratamento da requisição enviada pelo cliente.
- Recebem dois parâmetros:
  - ServletRequest.
  - ServletResponse.
- Tarefas usuais:
  - Extrair informações de requisição.
  - Acessar recursos externos.
  - Preencher a resposta, no caso de HTTP:
    - Preencher os cabeçalhos necessários (setContentType()).
    - Obter um stream de resposta (getWriter()).
    - Escrever os dados no stream (println()).
- Método service() (abstrato):

  public void service (ServletRequest,ServletResponse);
  - Será chamado sempre que chegar uma requisição.
  - Deve ser sobreposto pelo servlet específico.
  - Para HTTP: a classe HttpServlet redireciona as requisições encaminhadas para service() para métodos que refletem os métodos HTTP:
    public void doGet (HttpServletRequest,HttpServletResponse);
    public void doPost (HttpServletRequest,HttpServletResponse);
    public void doDelete (HttpServletRequest,HttpServletResponse);
    etc.
  - Um servlet HTTP deverá estender HttpServlet e implementa pelo menos um desses métodos.
- Obtenção de dados das requisições (métodos de HttpServletRequest):
  - getParameter(param): obtém parâmetro HTTP. Exemplo:
    String s = request.getParameter("nome_parametro");

- getParameterNames(): recupera os nomes de todos os parâmetros da requisição. Exemplo:
  Enumeration pnames = request.getParameterNames();
- setAttribute(nome,obj): define atributo (objeto que pode ser agrupado à requisição e passa a acompanhá-la). Exemplo:
  request.setAttribute("quantidade",5);
- getAttribute(nome): recupera atributo a partir de seu nome. Exemplo:
  int qtd = request.getAttribute("quantidade");
- getAttributeNames(): recupera os nomes de todos os atributos da requisição. Exemplo:
  Enumeration anames = request.getAttributeNames();
- getSession(): cria ou recupera uma sessão de usuário. Exemplos:
  HttpSession session = request.getSession(); *// cria ou recupera sessão*
  HttpSession session = request.getSession(**false**); *// recupera sessão se existir*

Sessões

- Como HTTP não mantém estado, as aplicações web devem mantê-lo quando necessário.
- Sessões são representados por objetos HttpSession e são obtidas a partir do objeto de requisição:
  HttpSession session = request.getSession();
- É possível guardar objetos em uma sessão através dos métodos getAttribute() e setAttribute() da mesma forma que foi feito para as requisições.
- Gerenciamento:
  - Para destruir uma sessão: session.invalidate().
  - Sessões podem expirar; para definir um tempo de duração da sessão: session.setMaxInactiveInternal().
  - O acesso à sessão é implementado com cookies se o cliente suportar.
- Observações:
  - Sempre que uma página contiver a url da outra, utilizar o método encodeUrl().
  - Se o cliente não suportar cookies, o identificador de sessão será passado como parâmetro de requisição.
  - Exemplo: out.println("<a␣href␣=␣'"+ encodeUrl("link.html")+ "'>").