**Redirecionamentos**

**Forward**

É executado internamente pelo servlet (controller);

O browser não sabe o que está ocorrendo durante o processamento no servidor, ou seja, não sabe por quais servlets ou páginas a requisição está passando;

No final do processamento da requisição a url da barra de endereços do browser não muda;

O reload da página resultante irá executar a requisição original;

**Redirect**

É um processo de dois passos, ao receber uma requisição a aplicação web “pede” ao browser para acessar uma segunda url, por isso a url muda;

O reload de página não repetirá a requisição original, mas sim a nova url (2ª requisição);

É um processo muito mais lento que um forward, pois são necessárias duas requisições, e não uma;

Objetos colocados no escopo do request original são perdidos durante o segundo request;

**EX**:

**// REDIRECT**  
protected void doPost(HttpServletRequest request, HttpServletResponse response)throws ServletException, IOException {  
   
 String email = req.getParameter("email");  
   
 Cliente cli = new Cliente();  
 cli.setEmail(email);  
   
 lista.add(cli);  
   
 resq.sendRedirect("cliente");  
   
}

**//FORWARD**  
 protected void doPost(HttpServletRequest request, HttpServletResponse response)  
 throws ServletException, IOException {  
   
 String email = req.getParameter("email");  
   
 Cliente cli = new Cliente();  
 cli.setEmail(email);  
   
 lista.add(cli);  
   
 RequestDispatcher dispatcher = req.getRequestDispatcher("cliente.jsp");  
 req.setAtribute("lista", lista);  
 dispatcher.forward(req,resp);  
   
 }

**Escopos de Servlets**

Eles duram apenas o tempo de uma requisição; ao término da mesma todos os dados setados no objeto que representa o escopo terão sido apagados. No escopo de sessão, quando um usuário acessar o sistema web, ele estabelece com o servidor uma sessão. Os dados setados no objeto que representa este escopo existem desde o instante inicial, quando o usuário acessa a aplicação, até que essa expire por inatividade, seja voluntariamente ou finalizada pela aplicação. Já no escopo de aplicação, os objetos vivem desde a inicialização do servidor de aplicação até que ele seja finalizado.

**EX:**

public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{

//setando atributos

request.setAttribute(“nome”,”valor”);

request.getSession().setAttribute(“nome”, “valor”);

getServletContext().setAttribute(“nome”,”valor”);

Object valorRequest = request.getAttribute(“nome”);

Object valorSession = request.getSession().setAttribute(“nome”);

Object valorApplication = getServletContext(“nome”).setAttribute(“nome”);

}