

## redirect と forward の違い

--	redirect	forward
ソース	<code>response.sendRedirect([ ページ 2 ]);</code>	<code>request.getRequestDispatcher([ ページ 2]).forward(request, response);</code>
処理の流れ	クライアント [ ページ 1 ] を要求 サーバ [ ページ 2 ] に redirect 命令を返却 クライアント [ ページ 2 ] を要求 サーバ [ ページ 2 ] を返却	クライアント [ ページ 1 ] を要求 サーバ [ ページ 2 ] を返却
パフォーマンス	悪い ( リクエストを 2 回送信する必要がある )	良い ( 通常通り 1 回のリクエストで良い )
リクエスト情報	引き継がない ( 全く別の 2 つのリクエストの為 )	引き継ぐ ( 同一の 1 つのリクエストの為 )
転送可能範囲	外部サーバも OK	同一サーバ内のみ

## Servlet

### Servlet

#### Forward

```
RequestDispatcher rDispatcher =
    request.getRequestDispatcher("/result.jsp");
rDispatcher.forward(request, response);
```

- Path が「/」で始まる場合  
コンテキストルートに対する相対パス
- Path が「/」以外で始まる場合  
現在のリクエスト URI からの相対表記であると解釈

#### API ドキュメントより

指定するのは相対的なパス名ですが、この Servlet が属しているコンテキスト外部のリソースは指定できません。パス名が"/"で始まっている場合はこのコンテキストルートに対する相対パスであると解釈されます。

#### Redirect

```
response.sendRedirect("top.html");
```

- Path が「/」で始まる場合  
Servlet コンテナのルートからの相対表記であると解釈
- Path が「/」以外で始まる場合  
現在のリクエスト URI からの相対表記であると解釈

## API ドキュメントより

指定されたリダイレクト先の URL を用いて、クライアントに一時的なリダイレクトレスポンスを送信します。URL としては相対 URL を指定することができます。従って、Servlet コンテナは相対 URL を絶対 URL に変換しなくてはなりません。指定された場所が「/」から始まらない相対表記である場合、コンテナは現在のリクエスト URI からの相対表記であると解釈します。指定された場所が「/」から始まる相対表記である場合、コンテナは Servlet コンテナのルートからの相対表記であると解釈します。

## Jsp

### Forward

```
<jsp:forward page="top.html" />
```

または

```
<% pageContext.forward("top.html"); %>
```

- Path が「/」で始まる場合  
コンテキストルートに対する相対パス
- Path が「/」以外で始まる場合  
現在のリクエスト URI からの相対表記であると解釈

### Redirect

```
<% response.sendRedirect("top.html"); %>
```

- Path が「/」で始まる場合  
Servlet コンテナのルートからの相対表記であると解釈
- Path が「/」以外で始まる場合  
現在のリクエスト URI からの相対表記であると解釈

## Struts

### Servlet

#### Forward

```
action クラス内で  
return (mapping.findForward("SUCCESS"));
```

または

```
return new ActionForward("/xxx.do?param=" + param);
```

- Path が「/」で始まる場合

コンテキストルートに対する相対パス

- Path が「/」以外で始まる場合  
現在のリクエスト URI からの相対表記であると解釈

## Redirect

```
action クラス内で  
return new RedirectingActionForward("/xxx.do?param=" + param);
```

- Path が「/」で始まる場合  
コンテキストルートに対する相対パス
- Path が「/」以外で始まる場合  
現在のリクエスト URI からの相対表記であると解釈

## 注意

HttpServletResponse.sendRedirect() が内部的に呼ばれるので Path が「/」で始まる場合は、Servlet コンテナルートからの相対表記になりそうだが、内部 (RequestProcessor#processForwardConfig あたり) で sendRedirect する前に Servlet コンテナルートからコンテキストルートの相対表記に変換している。

## API ドキュメントより

ActionForward は Action クラスの処理の結果、コントローラ Servlet(ActionServlet) によって実行される、RequestDispatcher.forward() または HttpServletResponse.sendRedirect() の遷移先を表わします。

## Jsp

### Forward

```
<logic:forward name="start" />
```

または

```
<% pageContext.forward("top.html"); %>
```

基本的に Action を指定

### Redirect

```
<logic:redirect action="/next" />
```

基本的に Action を指定

## 参照

<http://amsoft.minidns.net/modules/chapox/content.php?lid=52>

[JspForwardRedirect1.jsp]

```
<HTML>
<BODY>
<jsp:forward page="top.html" />
</BODY>
</HTML>
```

JspForwardRedirect1.jsp にアクセスするとサーバ内で top.html へのリクエストし、そのレスポンスを JspForwardRedirect1.jsp へのアクセス結果として返す。  
そのためブラウザの URL は JspForwardRedirect1.jsp にアクセスしたままになる。

[JspForwardRedirect2.jsp]

```
<HTML>
<BODY>
<% response.sendRedirect("top.html"); %>
</BODY>
</HTML>
```

JspForwardRedirect2.jsp にアクセスするとブラウザに対して「top.html へリダイレクトしなさい」という HTTP ステータスコードが返され、ブラウザが再度 top.html を読みに行く。