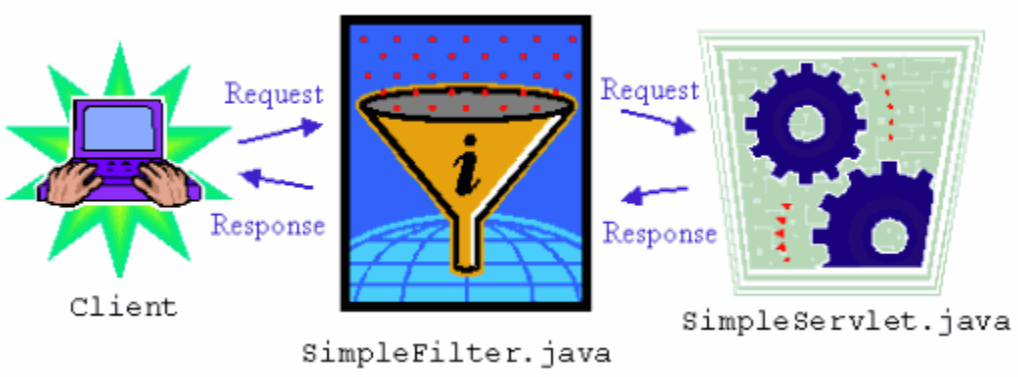
Filter Servlet

Filters

- A filter is an object that can transform a request or alter a response. Filters not create a response like servlet. It process request before it reaches to a servlet and can process response before it leaves to a servlet.

Writing a Simple Filter

- A filter is simply a Java class that implements the javax.servlet.Filter interface. It defines three methods:
  - public void doFilter (ServletRequest request, ServletResponse response, FilterChain chain)
  - public FilterConfig getFilterConfig()
  - public void setFilterConfig (FilterConfig filterConfig)


Request
Response
Request
Response
Client
SimpleFilter.java
SimpleServlet.java

- For implementing servlet we need to import *javax.servlet.Filter .* This class defines three methods
- *void init(FilterConfig config) throws ServletException*
  - Sets the filter's configuration object.
- *void destroy() :*
  - *F*or destroying the filter's configuration instance.
- *void doFilter(ServletRequest req, ServletResponse res, FilterChain chain) throws IOException, ServletException:*
  - Perform the actual filter work.

Advantages of Servlet Filter

- A filter can intercept a servlet's invocation before the servlet is called.
- Can examine a request before a servlet is called.
- Can modify the request headers and request data by providing a customized version of the request object that wraps the real request.
- Can modify the response headers and response data by providing a customized version of the response object that wraps the real response.
- Intercept a servlet's invocation after the servlet is called.

Configuring filter in web.xml

- To use this filter, you must declare it in the web.xml deployment descriptor using the <filter>tag, as shown below :

```
<filter>
        <filter-name>FilterDemo</filter-name>
        <filter-class>FilterDemo</filter-class>
</filter>
```

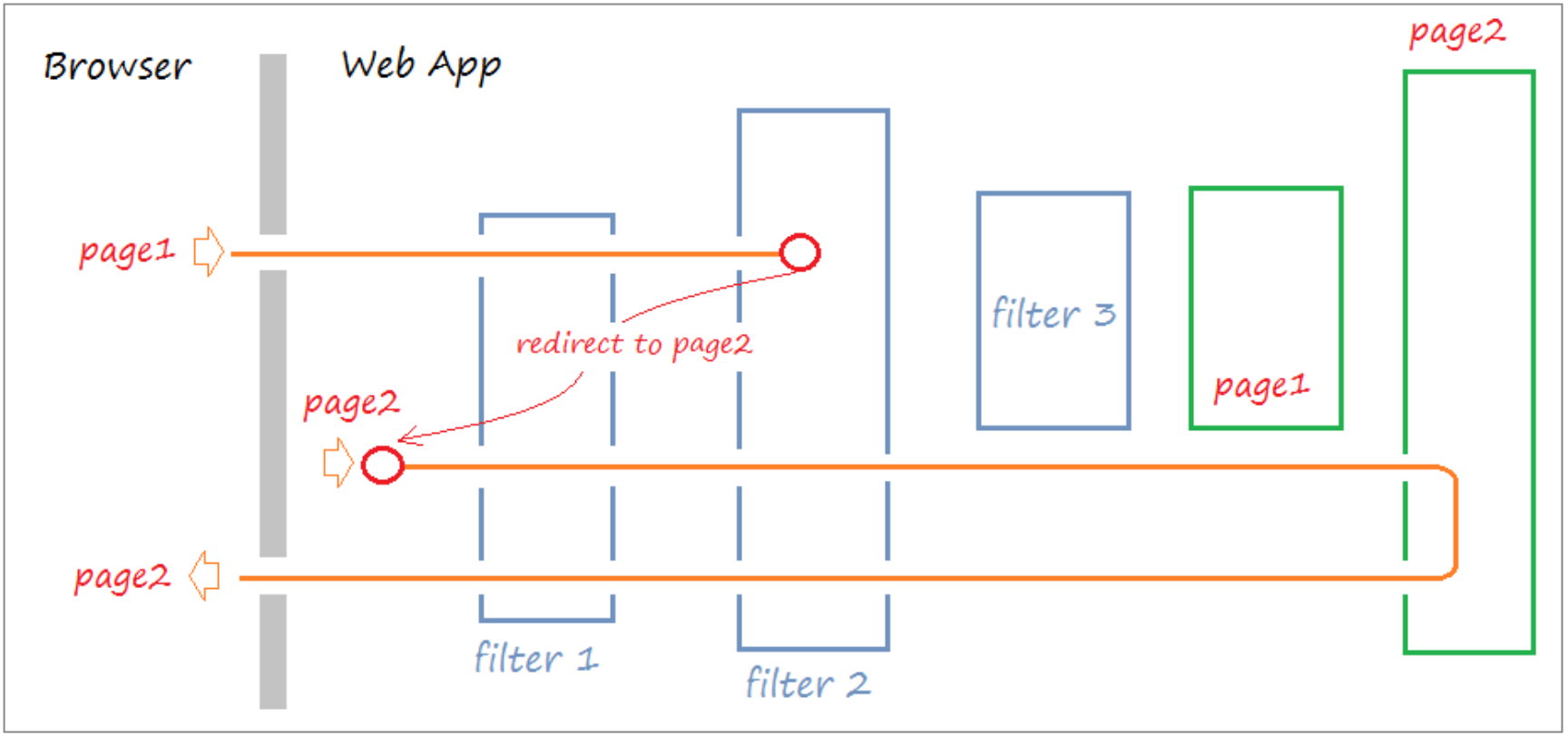
- You can apply a filter to certain URL patterns or servlet names using the <filter-mapping> tag :

```
<filter-mapping>
        <filter-name>FilterDemo</filter-name>
        <url-pattern>/*</url-pattern>
</filter-mapping>
```

Why need Server-Filter?

- Normally, when user requests a web page, a request is sent to the server, it will have to pass through the filter before before reaching the page required.
- However, there are situations where the user's request do not pass all Filters
- In case that the user requests a page (page 1), this request must pass Filters, in a certain filter, the request are redirected to a different page (page2).

Why need Server-Filter?



Why need Server-Filter?

- **Example situation:** Users submit a request for personal info.
- Request will be sent to the Server.
- It goes through Filter that records log information.
- It goes to Filter to check user login or not, this filter found that the user is not login, it will redirect the request of the user to the login page.

What can Servlet-filter do?

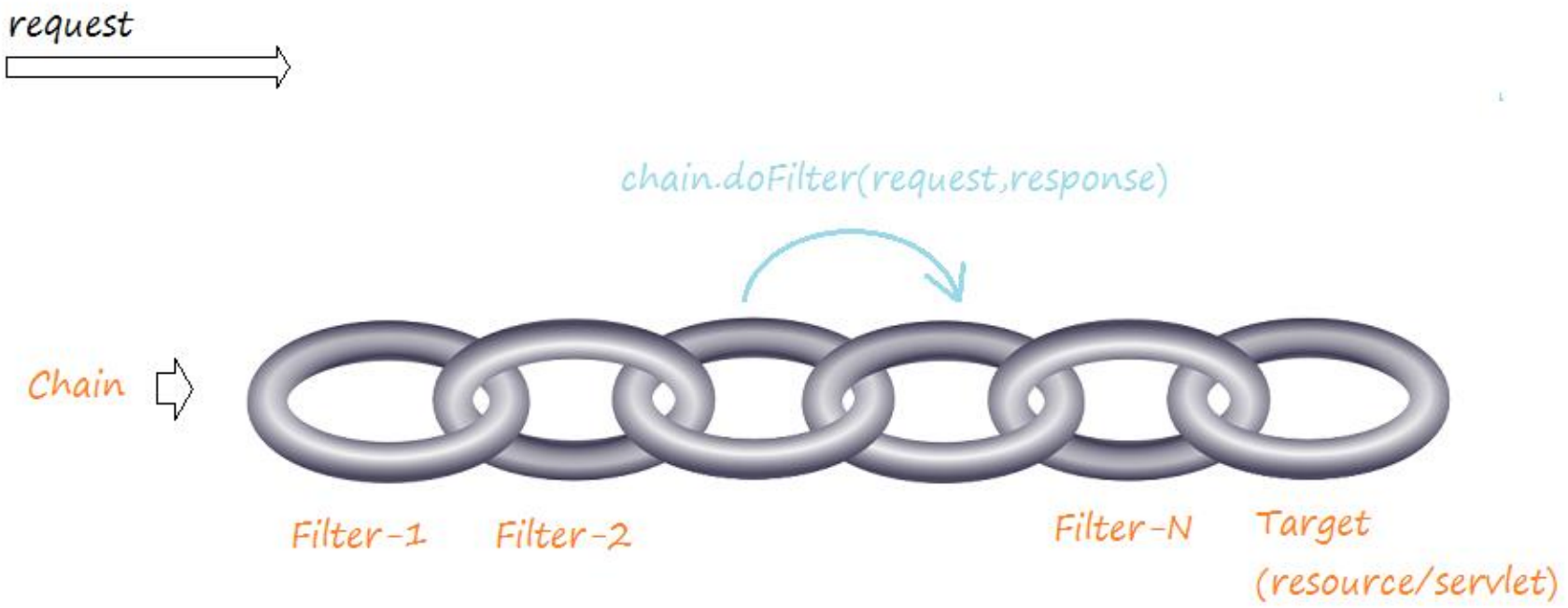
- Sometimes, you believed that Filter only used to redirect the user requests to a different page, or block access to a particular page if the user has no right. Or used to write log information.
- In fact, Filter can be used to encoding web pages. For example, set **UTF-8** encoding for the page. Opening and closing the connection to the database and preparing JDBC transaction

Filter Example

```
// Allow the request to move forward.
// It can go to the next filter or to the target.
   chain.doFilter(request, response);
```

- Refer Ecllipse Example

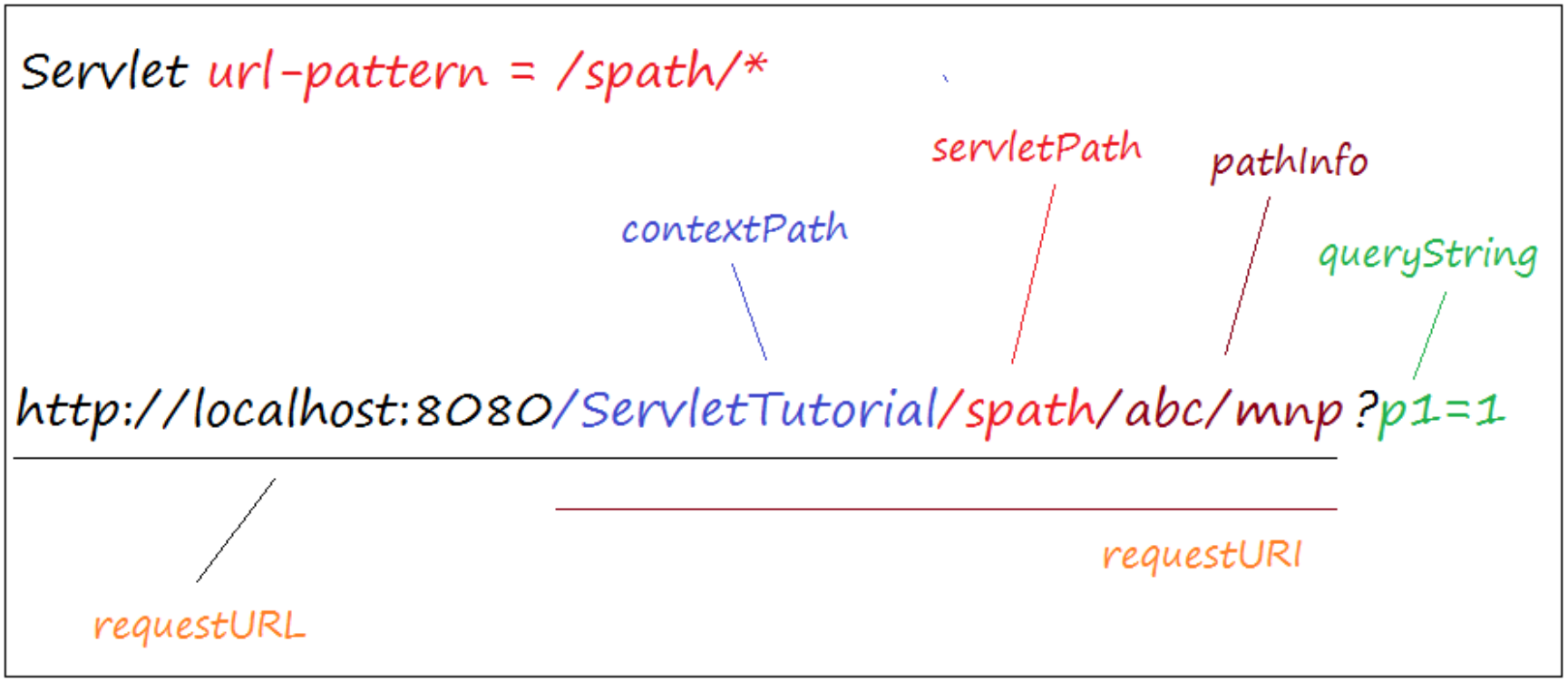
Working models of Filter



Working models of Filter

- When the user sends a request, the target may be a resource or a servlet. Request must be passed through the filter and finally the target. The filter and target is chained together like illustration previous slide.
- Use ***chain.doFilter (request, response)*** to move the request to the next stage.
- If the ***chain.doFilter (request, response)*** filter is not called, the user's request will not reach the target, it will be stopped at that filter.

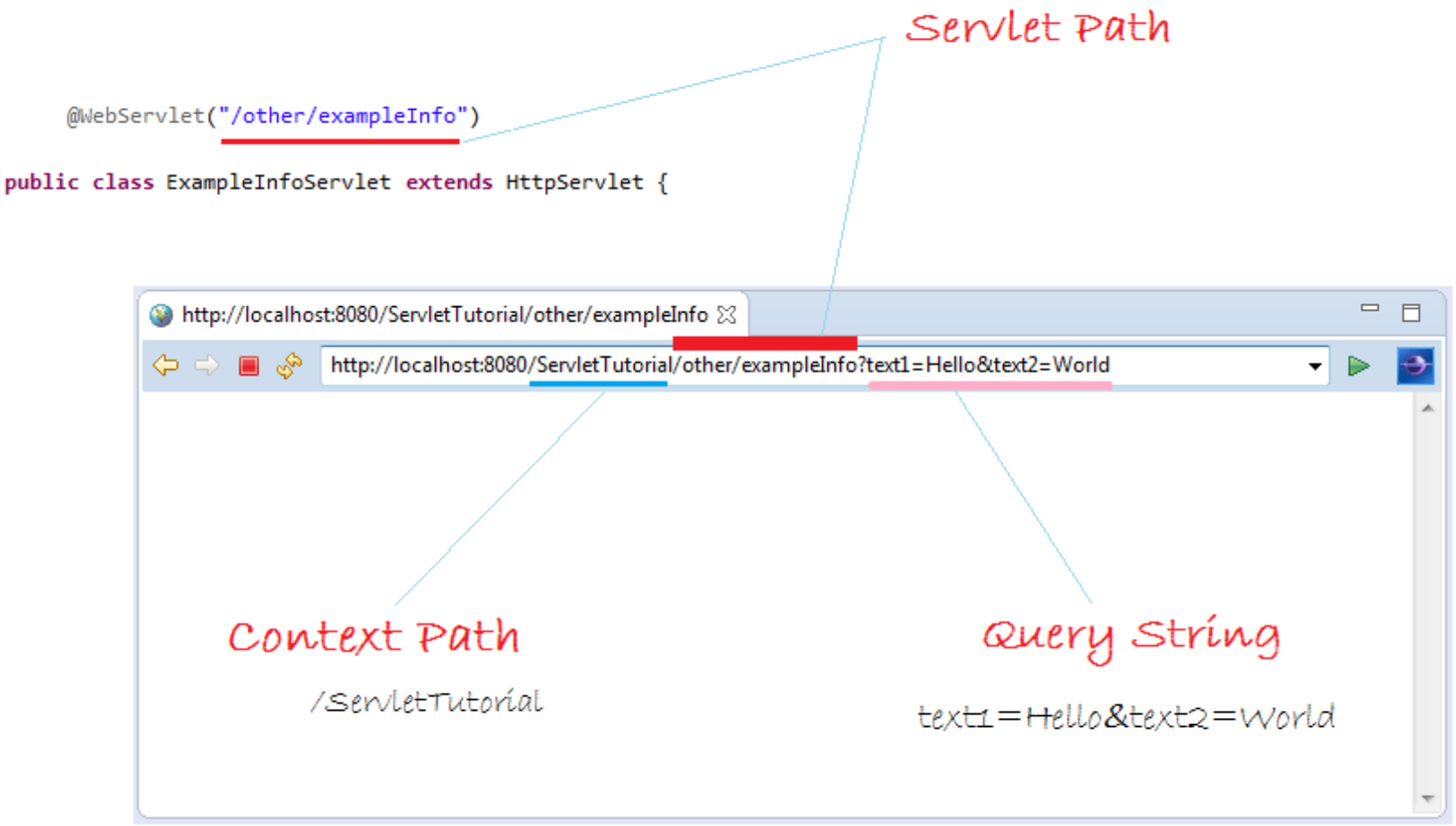
Servlet Url Pattern



4 ways to configure a URL path

URL Pattern	Examples
/*	http://example.com/contextPath
	http://example.com/contextPath/status/abc
/status/abc/*	http://example.com/contextPath/status/abc
	http://example.com/contextPath/status/abc/mnp
	http://example.com/contextPath/status/abc/mnp?date=today
	~~http://example.com/contextPath/test/abc/mnp~~
*.map	http://example.com/contextPath/status/abc.map
	http://example.com/contextPath/status.map?date=today
	~~http://example.com/contextPath/status/abc.MAP~~
/	Đây là Servlet mặc định.

URL Annotations



Example Code URL Pattern

```
// You can configure one or multiple 'URL Patterns' can access this Servlet.
@WebServlet(urlPatterns = { "/annotationExample", "/annExample" },
    initParams = {
@WebInitParam(name = "emailSupport1", value = "abc@example.com"),
@WebInitParam(name = "emailSupport2", value = "tom@example.com") })
public class AnnotationExampleServlet extends HttpServlet
```