

# Web Component Development with Servlet & JSP Technologies (EE 6)

Module-11: Asynchronous Servlets and Clients

Team Emertxe

**ORACLE®**

**Certified Expert**

Java EE 6 Web  
Component Developer



# Objectives



*Upon completion of this module, you should be able to:*

- Use the Asynchronous Servlet mechanism
- Use JavaScript to send an HTTP request from a client
- Process an HTTP response entirely in JavaScript
- Combine these techniques to create the effect of server-push

# Relevance



Discussion - The following questions are relevant to understanding what technologies are available for developing web applications and the limitations of those technologies:

- What happens if the response to an HTTP request cannot be made until a message is received from a third party? Must the servlet thread be blocked indefinitely?
- Suppose dozens of clients are cooperating, for example, in a chat room. Must a request from every client be blocked waiting until one client triggers an update?

# Asynchronous Servlets



Java EE 6 adds the option to perform request processing asynchronously. If this is done, then the servlet execution thread can be freed up to service requests from other clients, and the generation of the response may be performed in another thread, for example the thread that creates the trigger condition that allows the response to be prepared.

# Separating Request Receipt from Response Generation



- To allow for separation of request from response generation, the servlet API provides a class called `AsyncContext`. A servlet that wishes to hand off response generation to another thread can obtain an `AsyncContext` object and pass this to another thread, perhaps using a queue.
- The `AsyncContext` object provides access to the original `HttpServletRequest` and `HttpServletResponse` objects.
- `RequestDispatcher`, is achieved using one of several dispatch methods in the `AsyncContext` itself.

# Asynchronous Servlet Example



```
protected void processRequest(HttpServletRequest request,  
    HttpServletResponse response) throws ServletException, IOException {  
    AsyncContext ac = request.startAsync();  
    handler.addJob(ac);  
}
```

# Forwarding and Filtering

Asynchronous handlers are permitted to dispatch or forward their processing to other servlets or JSPs, including those that are not asynchronous (that is, they do not carry the `asyncSupported=true` element in the annotation). This allows a standard response page to be used as a view for both synchronous and asynchronous responses.

Filters can be invoked on asynchronous invocations too. If the `AsyncContext.dispatch` call is used on a matching URL, the dispatcher type will be `ASYNC`

# Asynchronous Listeners



The AsyncListener interface must be implemented by the listener that will be notified when these situations arise. The interface defines four listener methods:

- onComplete(AsyncEvent)
- onError(AsyncEvent)
- onTimeout(AsyncEvent)
- onStartAsync(AsyncEvent)



# Asynchronous JavaScript Clients(AJAX)



AJAX is a technique for creating fast and dynamic web pages.

AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.

## Combining Asynchronous Servlets With Asynchronous JavaScript



- Consider a page containing JavaScript code that makes an asynchronous request for update from the server, and a server-side implementation that uses the asynchronous servlet techniques.
- Now, if the server does not choose to respond for several minutes, the user is not inconvenienced, as the body of the page is operating normally.
- Similarly, because of the use of asynchronous servlet execution, the resource load on the server is minimized. Consequently, the response may be sent at a time convenient to the server, and the effect of server-push is achieved in an architecturally manageable way.

# Stay connected



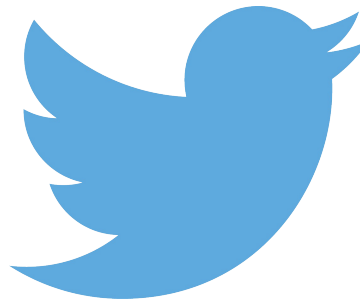
**About us:** Emertxe is India's one of the top IT finishing schools & self learning kits provider. Our primary focus is on Embedded with diversification focus on Java, Oracle and Android areas

Emertxe Information Technologies,  
No-1, 9th Cross, 5th Main,  
Jayamahal Extension,  
Bangalore, Karnataka 560046  
T: +91 80 6562 9666

E: [training@emertxe.com](mailto:training@emertxe.com)



<https://www.facebook.com/Emertxe>



<https://twitter.com/EmertxeTweet>



**slideshare**  
Present Yourself

<https://www.slideshare.net/EmertxeSlides>

Thank You