Asynchronous Programming

Kevin Jones

@kevinrjones www.rocksolidknowledge.com

Objectives

Understand why asynchronous programming is important

Understand how to write asynchronous servlets

Why Asynchronous Servlets?

Slow Backend Resources

Reuse of Threads

Server Push

How do Async Servlets Work?

Get an asynchronous context

Start the context to handle the request

Use the context to return the response

(Optionally) Add a 'listener' to handle events

Mark the Servlet as Async

```
@WebServlet(urlPatterns = "/simple", asyncSupported = true)
public class FirstAsyncServlet extends HttpServlet {
}
```

Grab the Context

```
@Override
protected void doGet(
       HttpServletRequest req, HttpServletResponse resp) {

    final AsyncContext ctx = req.startAsync();
}
```

Start and Complete the Async Work

```
ctx.start(() -> {

    try {
        ctx.getResponse().getWriter().write(...);
    }
    catch (IOException e) {
        log("Problem processing task", e);
    }

    ctx.complete();
});
```

Alternatively – Dispatch the Call

```
final AsyncContext ctx = req.startAsync();
ctx.dispatch("/someUrl");
```

Async Filters

Filters can also be asynchronous

Mechanism similar to Servlet

Summary

Servlet 3 specification introduced asynchronous behaviour

- Servlets
- Filters

Be careful when using

- May end up offloading one thread to another with no benefit