

INTRODUCING...

WILDFLY

AND UNDERTOW

Pepa Cacek

Wild**Fly**



WHAT IS WILDFLY?

- Previously “JBoss Application Server”
- Lightweight Managed Application Runtime
- Developer Friendly
- Supports Java EE standards and beyond
- Enterprise capabilities for production
- Open Source
- Upstream for JBoss EAP
- Cloud Ready

WILDFLY 8

- Java EE 7 Support
- New High Performance Web Server (Undertow)
- Greatly Reduced Port Usage
- Management Role Based Access Control & Auditing
- Patching Infrastructure
- Quickstarts!
- **Beta Recently Released!**

JAVA EE7 HIGHLIGHTS

- Modern Web Support
 - Non-Blocking Listeners (Servlet 3.1)
 - Web Socket Support (JSR-356)
 - JSON Processing APIs (JSR-353)
 - Enhanced REST processing (JAX-RS 2.0)
- Batch Processing (JSR-352)
- Concurrency Utilities (JSF-236)
- Improved Messaging APIs (JMS 2.0)
- CDI Everywhere
- Portability Improvements

UNDERTOW

- New webserver project integrated in WildFly 8
- Servlet 3.1 container
- HTTP Upgrade
- Embeddable

<http://undertow.io/>

UNDERTOW

HTTP UPGRADE - WHAT IS IT?

- Allows for a HTTP connection to be upgraded to another protocol
- Mechanism defined in the HTTP/1.1 RFC

HTTP UPGRADE - WHY USE IT?

- It allows you to multiplex multiple protocols over the HTTP port
- Makes firewall configuration simpler, and works around firewalls that block port other than 80 and 443
- Removes need to configure multiple ports

HTTP UPGRADE - EJB EXAMPLE

CLIENT REQUEST

```
GET / HTTP/1.1
Host: example.com
Upgrade: jboss-remoting
Connection: Upgrade
Sec-JbossRemoting-Key: dGhlIHNhbXBsZSBub25jZQ==
```

SERVER RESPONSE

```
HTTP/1.1 101 Switching Protocols
Upgrade: jboss-remoting
Connection: Upgrade
Sec-JbossRemoting-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+x0o=
```

SERVLET UPGRADE API

```
public class MyProtocolUpgradeFilter implements Filter {  
  
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws IOException, ServletException {  
  
        HttpServletRequest req = (HttpServletRequest) request;  
        if("my-protocol".equals(req.getHeader("Upgrade"))) {  
            HttpServletResponse resp = (HttpServletResponse) response;  
            resp.setHeader("Upgrade", "my-protocol");  
            req.upgrade(MyProtocolHandler.class);  
            return;  
        }  
        chain.doFilter(request, response);  
    }  
    ...  
}
```

THE END

BY PEPA 'KWART' CACEK