

WildFly



The artist formerly known as JBoss AS

Harald Pehl

[@haraldpehl](https://twitter.com/haraldpehl)

WildFly?

- JBoss AS
- JBoss Community ~ 100 Projekte
- JBoss Enterprise Application Platform



WildFly!

- Folgeversion von JBoss AS 7
- Community Version
- Open Source
- FAQ @ <http://www.wildfly.org/faq/>

Highlights

- Java EE 7
- Undertow
- Ports--
- RBAC - Role Bases Access Control
- Patching

Java EE 7

- Servlet 3.1
- Web Socket Support (JSR-356)
- JSON Processing API (JSR-353)
- JAX-RS 2.0
- Batch Processing (JSR-352)
- Concurrency (JSR-236)
- JMS 2.0
- CDI

Servlet 3.1

- Undertow: Servlet 3.1 Container
 - Async I/O Support (Non-Blocking Listeners)
 - HTTP Upgrade Support

Asnyc I/O

- Bisher
 - Ein Thread pro Verbindung
 - Blocking
 - Viele Verbindungen → viele Threads
- Non blocking I/O
 - Callback, wenn die Operation fertig ist
 - Ein Thread kann viele Verbindungen bedienen
 - Complexerer Code

Async Write

```
protected void doGet(final HttpServletRequest req,
                     final HttpServletResponse resp)
                     throws ServletException, IOException {
    final AsyncContext context = req.startAsync();
    final ServletOutputStream outputStream = resp.getOutputStream();
    final String[] messages = {"Hello ", "async ", "world"};

    outputStream.setWriteListener(new WriteListener() {
        int pos = 0;

        @Override
        public synchronized void onWritePossible() throws IOException {
            while (outputStream.isReady() && pos < messages.length()) {
                outputStream.write(messages[pos++].getBytes());
            }
            if (pos == messages.length()) context.complete();
        }
    });
}
```


HTTP Upgrade

- Multiplex verschiedener Protokolle über HTTP
- Standard definiert in HTTP/1.1 RFC
- HTTP Upgrade Header
- Einfachere Firewall Konfiguration
- Setup unterschiedlicher Ports entfällt

HTTP Upgrade / WildFly

- Nur noch zwei Ports: 8080 und 9990
- EJB und JNDI → 8080
- Management → 9990
- Ziel: „One port to rule them all“

HTTP Upgrade / EJB

Client Request

```
GET / HTTP/1.1  
Host: example.com  
Upgrade: jboss-remoting  
Connection: Upgrade  
Sec-JbossRemoting-Key: dGhlIHVhbXBsZSBub25jZQ==
```

Server Response

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

HTTP Upgrade

- Prüfen des Upgrade Headers
- Setzen des entsprechenden Response Headers
- `HttpServletRequest.upgrade(Class<T> handlerClass)`
- Wenn der Request beendet ist, übernimmt der Handler die Kontrolle

HTTP 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);
    }
}
```

HTTP Upgrade API

```
public interface HttpUpgradeHandler {  
    public void init(WebConnection wc);  
    public void destroy();  
}  
  
public interface WebConnection extends AutoCloseable {  
    public ServletInputStream getInputStream()  
        throws IOException;  
  
    public ServletOutputStream getOutputStream()  
        throws IOException;  
}
```

Web Sockets

- Kommunikation in beide Richtungen möglich
- Verbindungsaufbau per HTTP Upgrade
- Implementiert in Undertow
- Java Web Socket API (JSR-356)
- Ermöglicht Einsatz von Frameworks wie Atmosphere

Server Endpoint

```
@ServerEndpoint("/websocket/{name}")
public class HelloEndpoint {

    @OnOpen //invoked when the client first connects
    public void onOpen(final Session session) {
        session.getAsyncRemote().sendText("hi");
    }

    @OnMessage
    public String message(String message,
        @PathParam("name") String name) {
        return "Hello " + name + " you sent" + message;
    }
}
```


Client Endpoint

```
@ClientEndpoint
public class AnnotatedClientEndpoint {

    @OnOpen
    public void onOpen(final Session session) {
        session.getAsyncRemote().sendText("hi");
    }

    @OnMessage
    public void onMessage(final String message, final Session session) {
        System.out.println(message);
    }
}
```

Undertow

- Neuer Webserver in WildFly
- Servlet 3.1
- Web Socket Support (JSR-356)
- HTTP Upgrade
- None Blocking I/O
- Performance↑, Memory↓
- Integrierbar in Standalone & Testumgebungen

Embedded Undertow

- Undertow ermöglicht einfaches Einbetten
- Fluent API
- „Eat your own dogfoot“

Embedded Undertow

```
public class HelloWorldServer {  
    public static void main(final String[] args) {  
        Undertow server = Undertow.builder()  
            .addListener(8080, "localhost")  
            .setDefaultHandler(new HttpHandler() {  
                @Override  
                public void handleRequest(final HttpServerExchange exchange)  
                    throws Exception {  
                    exchange.getResponseHeaders().put(  
                        Headers.CONTENT_TYPE, "text/plain");  
                    exchange.getResponseSender().send("Hello World");  
                }  
            }).build();  
        server.start();  
    }  
}
```


RBAC

- Zugriffsschutz für Management Operationen
- „Seperation of Duties“
- Sieben vordefinierte Rollen
- Scoped Roles
 - Host
 - Servergruppen
- Benutzer und Gruppen → Rollen

RBAC Rollen

	Monitor	Operator	Maintainer	Deployer	Auditor	Admin.	SuperUser
Read Config & State	x	x	x	x	x	x	x
Read Sensitive Data					x	x	x
Modify Sensitive Data						x	x
Read / Modify Audit Log					x		x
Modify Runtime State		x	x	x		x	x
Modify Persistent Config			x	x		x	x
Read / Modify Access Control						x	x

RBAC Demo

WildFly als Platform

- Apache Camel
<https://docs.jboss.org/author/display/wfcam/Home>
- Escalante
<http://escalante.io/>
- DMR.scala / DMR.repl
<https://github.com/hal/dmr.scala>
<https://github.com/hal/dmr.repl>

Resources

- www.wildfly.org
- Downloads: <http://www.wildfly.org/download>
- Forum & Wiki: <https://community.jboss.org/en/wildfly>
- Docs: <https://docs.jboss.org/author/display/WFLY8/>
- Dev Mailing List: wildfly-dev@lists.jboss.org
- IRC on freenode: #wildfly and #wildfly-dev
- Code: <https://github.com/wildfly/wildfly>