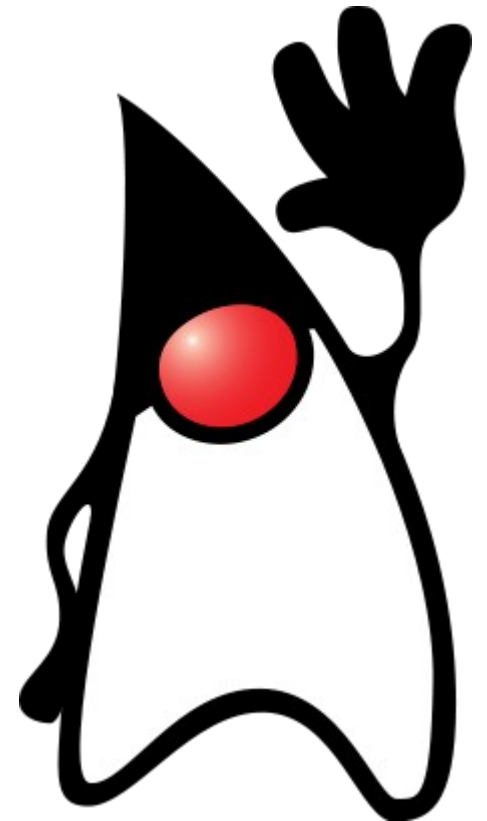


WildFly and Java EE 7

- What are they?
- New Features
- Some Examples
- Questions


Java EE 7

- Enterprise Java computing platform
- Extends Java SE (Standard Edition)
- Version History
 - J2EE 1.2 (December 12, 1999)
 - J2EE 1.3 (September 24, 2001)
 - J2EE 1.4 (November 11, 2003)
 - Java EE 5 (May 11, 2006)
 - Java EE 6 (Dec 10, 2009)
 - Java EE 7 (May 28, 2013)




Java EE 7


Web Container	WebSocket	Java SE	
	Concurrency Utilities		
	Batch		
	JSON-P		
	Bean Validation		
	EJB Lite		
	EL		
	Servlet		JavaMail
	JavaServer Faces		JSP
	Connectors		Java Persistence
	JMS		Management
	WS Metadata		Web Services
	JACC		JASPIC
	JAX-RS		JAX-WS
	JSTL		JTA
	CDI		Dependency Injection

 New in Java EE 7

EJB Container	Concurrency Utilities	Java SE	
	Batch		
	JSON-P		
	CDI		
	Dependency Injection		
	JavaMail		
	Java Persistence		
	JTA		
	Connectors		
	EJB		JMS
	Management		WS Metadata
	Web Services		JACC
	JASPIC		JAX-RS
	Bean Validation		JAX-WS

 New in Java EE 7

Application Client Container	Java Persistence Management	Java SE	
	WS Metadata		
	Web Services		
	Application Client		JSON-P
	JMS		
	JAX-WS		
	Bean Validation		
	JavaMail		
	CDI		
	Dependency Injection		

 New in Java EE 7

Java API for WebSocket

- Traditional Web is stateless and half duplex
 - Client requests a resource and server responds
 - Always initiated by the client and server can't send data without the client requesting it first
 - Limiting in today's complex web
- WebSockets allow full duplex communication over the TCP protocol

Java API for WebSocket

- Consists of two parts: handshake and data transfer

```
GET /path/to/websocket/endpoint HTTP/1.1
Host: localhost
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: xqBt3ImNzJbYqRINxEFlkg==
Origin: http://localhost
Sec-WebSocket-Version: 13
```

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: K7DJLdLooIwIG/M0pvWFB3y3FE8=
```

- Supports text (UTF-8) and binary messages
- `ws://host:port/path?query`
- `wss://host:port/path?query` (encrypted)

Java API for JSON Processing

- JSON = JavaScript Object Notation

```
{ "menu": {  
  "id": "file",  
  "value": "File",  
  "popup": {  
    "menuitem": [  
      { "value": "New", "onclick": "CreateNewDoc()" },  
      { "value": "Open", "onclick": "OpenDoc()" },  
      { "value": "Close", "onclick": "CloseDoc()" }  
    ]  
  }  
}
```

```
<menu id="file" value="File">  
  <popup>  
    <menuitem value="New" onclick="CreateNewDoc()" />  
    <menuitem value="Open" onclick="OpenDoc()" />  
    <menuitem value="Close" onclick="CloseDoc()" />  
  </popup>  
</menu>
```

- JSON allows lightweight data interchange
- First JSON processing APIs included in Java EE 7
- Object model and Stream model support

Batch Applications

- Jobs that can be executed without user interaction
- Java EE 7 provides a batch framework for creating and running batch jobs
- Consists of a:
 - runtime,
 - job specification language (XML),
 - API to interact with runtime,
 - API to implement artifacts

Batch Applications

- Job definition example

```
<job id="loganalysis" xmlns="http://xmlns.jcp.org/xml/ns/javaee" version="1.0">

  <properties>
    <property name="input_file" value="input1.txt"/>
    <property name="output_file" value="output2.txt"/>
  </properties>

  <step id="logprocessor" next="cleanup">
    <chunk checkpoint-policy="item" item-count="10">
      <reader ref="com.xyz.pkg.LogItemReader"></reader>
      <processor ref="com.xyz.pkg.LogItemProcessor"></processor>
      <writer ref="com.xyz.pkg.LogItemWriter"></writer>
    </chunk>
  </step>

  <step id="cleanup">
    <batchlet ref="com.xyz.pkg.CleanUp"></batchlet>
    <end on="COMPLETED"/>
  </step>

</job>
```


JavaServer Faces 2.2

- Stateless views
 - No more timeouts on JSF login pages
- More HTML5 friendly
 - pass-through attributes and elements
- Faces Flow
- File Upload component



Other Java EE 7 enhancements

- Enterprise JavaBeans (EJB)
 - Asynchronous local session beans in EJB Lite
 - Non-persistent timers in EJB Lite
 - <http://docs.oracle.com/javaee/7/tutorial/doc/overview008.htm#BNACL>
- Java Message Service
 - New simplified API
 - <http://docs.oracle.com/javaee/7/tutorial/doc/overview008.htm#BNACQ>
- Concurrency Utilities
 - Provides asynchronous capabilities to application components
 - <http://docs.oracle.com/javaee/7/tutorial/doc/overview008.htm#CJAFGFCJ>
- Servlet 3.1
 - Non-blocking IO
 - HTTP protocol upgrade mechanism
 - <http://docs.oracle.com/javaee/7/tutorial/doc/overview008.htm#BNACM>

WildFly

(Formerly JBoss AS or JBoss)

- Application Server
- Now developed by Red Hat
- Written in Java
- Full Java EE 7 Certification



Undertow

- Default web server in Wildfly
- Full support of WebSockets
- Servlet 3.1 support
- Supports both blocking and non-blocking IO
- Extremely lightweight
 - < 4MB heap space



Performance

Peak plaintext responses per second, m1.large EC2 instances (14 tests)								
Framework	Peak performance (higher is better)	Cls	Lng	Plt	FE	Aos	IA	Errors
undertow	240,499 100.0%	Plt	Jav	Utw	Non	Lin	Rea	1,061
netty	146,091 60.7%	Plt	Jav	Nty	Non	Lin	Rea	0
spray	89,876 37.4%	Mcr	Sca	Spr	Non	Lin	Rea	0
gemini	42,604 17.7%	Ful	Jav	Svt	Res	Lin	Rea	0
revel	26,301 10.9%	Ful	Go	Go	Non	Lin	Rea	0
servlet	21,333 8.9%	Plt	Jav	Svt	Res	Lin	Rea	187
compojure	17,758 7.4%	Mcr	Clj	Svt	Res	Lin	Rea	0
nodejs	9,658 4.0%	Plt	JS	njs	Non	Lin	Rea	0
go	9,182 3.8%	Plt	Go	Go	Non	Lin	Rea	0
rack-jruby	5,062 2.1%	Plt	Rby	JRb	Tor	Lin	Rea	0
dart	2,964 1.2%	Plt	Dar	Dar	ngx	Lin	Rea	5,754
aspnet-mysql-raw	2,931 1.2%	Ful	C#	Net	IIS	Win	Rea	53
cpoll_cppsp	1,015 0.4%	Mcr	C++	Cpl	Non	Lin	Rea	8,203
php	809 0.3%	Plt	PHP	FPM	ngx	Lin	Rea	83,686

In this test, the framework responds with the simplest of responses: a "Hello, World" message rendered as plain text. The size of the response is kept small so that gigabit Ethernet is not the limiting factor for all implementations. HTTP pipelining is enabled and higher client-side concurrency levels are used for this test.

<http://www.techempower.com/benchmarks/#section=data-r6&hw=ec2&test=plaintext>

Port Reduction

- Undertow supports HTTP upgrade...
- WildFly can now multiplex all of its protocols on two HTTP ports
 - Management (default is 9990)
 - Application (default is 8080)

Upgraded Components

Arquillian 1.1.2.Final-wildfly-1	Byteman 2.1.4	EJB Client 2.0.0.Final
Eclipse JDT Core Compiler 4.3.1	Groovy 2.2.1	Hal 2.1.1.Final
Hibernate 4.3.1.Final	Hibernate Search 4.5.0.Final	Hibernate Validator 5.0.3.Final
Hornetq 2.4.1.Final	Invocation 1.2.1.Final	Ironjacamar 1.1.3.Final
JBeret 1.0.0.Final	JBoss Logging 3.1.4.GA	JBoss Logmanager 1.5.2.Final
JBoss Marshalling 1.4.3.Final	JGroups 3.4.2.Final	Jackson 1.9.13
Jastow 1.0.0.Final	Jipijapa 1.0.1.Final	Log4j JBoss LogManager 1.1.0.Final
Metadata 8.0.0.Final	Mod_cluster 1.3.0.Final	Mojarras 2.2.5-jbossorg-3
Msc 1.2.0.Final	Narayana 5.0.0.Final	Netty 4.0.15.Final
Netty-xnio-transport 0.1.1.Final	PicketBox 4.0.20.Final	PicketLink 2.5.2.Final
Remote naming 2.0.0.Final	Remoting 4.0.0.Final	Remoting JMX 2.0.0.Final
SASL 1.0.4.Final	Santuario 1.5.6	Undertow 1.0.0.Final
Weld 2.1.2.Final	Wildfly Security Manager 1.0.0.Final	XNIO 3.2.0.Final

Demos and Walkthrus

Questions