
Java™

What is new in Servlet 3.1, JSR 340?

Shing Wai Chan (陳成威)
Servlet 3.1 Specification Lead
java.net/blog/swchan2

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract.
It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Servlet 3.1 Overview

- **FINAL**: Part of Java EE 7
- Upgrade from Servlet 3.0
- Scale
  - Expose Non-blocking IO API
- Support newer technologies that leverage HTTP protocol for the initial handshake
  - Support general upgrade mechanism for protocols like WebSocket
- Security enhancements

Non-blocking IO

Traditional IO Example

```
public class TestServlet extends HttpServlet
    protected void doPost(HttpServletRequest request,
 HttpServletResponse response) throws IOException,
 ServletException {
        ServletInputStream input =
 request.getInputStream();
        byte[] b = new byte[1024];
        int len = -1;
        while ((len = input.read(b)) != -1) {
            …
        }
    }
}
```

Non Blocking IO

Overview

- Add two new interfaces: **`ReadListener`**, **`WriteListener`**
- Add APIs to **`ServletInputStream`**, **`ServletOutputStream`**
- For asynchronous and upgrade only

Non-blocking IO

javax.servlet.ReadListener

```
public interface ReadListener extends EventListener {

    public void onDataAvailable() throws IOException;

    public void onAllDataRead() throws IOException;

    public void onError(Throwable t);
}
```

Non-blocking IO

javax.servlet.WriteListener

```
public interface WriteListener extends EventListener {

    public void onWritePossible() throws IOException;

    public void onError(Throwable t);
}
```

Non-blocking IO

ServletInputStream, ServletOutputStream

- **javax.servlet.ServletInputStream**
  - **public abstract boolean isFinished()**
  - **public abstract boolean isReady()**
  - **public abstract void setReadListener(ReadListener listener)**
- **javax.servlet.ServletOutputStream**
  - **public abstract boolean isReady()**
  - **public abstract setWriteListener(WriteListener listener)**

Non-blocking IO

Example

```
public class TestServlet extends HttpServlet {
    protected void doPost(HttpServletRequest req, HttpServletResponse
res) throws IOException, ServletException {
        AsyncContext ac = req.startAsync();
        …
        ServletInputStream input = req.getInputStream();
        ReadListener readListener = new ReadListenerImpl(input, output,
ac);
        input.setReadListener(readListener);
    }
}
```

Non-blocking IO

Example (cont'd)

```
public class ReadListenerImpl implements ReadListener {
    …
    public void onDataAvailable() throws IOException {
        …
        int len = -1;
        byte b[] = new byte[1024];
        while (input.isReady() && (len = input.read(b)) != -1) {
            …
        }
    }

    public void onAllDataRead() throws IOException {
        ac.complete();
    }

    public void onError(final Throwable t) {
        …
    }
}
```

Non-blocking IO

Example 2

```
public class TestServlet2 extends HttpServlet {
    protected void doPost(HttpServletRequest req, HttpServletResponse
res) throws IOException, ServletException {
        AsyncContext ac = req.startAsync();
        …
        ServletOutputStream output = req.getOutputStream();
        WriteListener writeListener = new WriteListenerImpl(output,
ac);
        output.setWriteListener(writeListener);
    }
}
```

Non-blocking IO

Example 2 (cont'd)

```
public class WriteListenerImpl implements WriteListener {
    …
    public void onWritePossible() throws IOException {
        …
        int len = -1;
        byte b[] = new byte[1024];
        while (output.isReady()) {
          …
        }
       …
    }

    public void onError(final Throwable t) {
        …
    }
}
```

Protocol Upgrade

HTTP Upgrade

- HTTP 1.1 (RFC 2616)
- Connection
- Transition to some other, incompatible protocol
  - For examples, IRC/6.9, Web Socket

Protocol Upgrade

Overview

- Allow a portable way to upgrade HTTP request
- Add API to **`HttpServletRequest`**
- Add two new interfaces
  - **`javax.servlet.http.HttpUpgradeHandler`**
  - **`javax.servlet.http.WebConnection`**
- Can use non-blocking IO API in upgrade

Protocol Upgrade

HttpUpgradeHandler, WebConnection

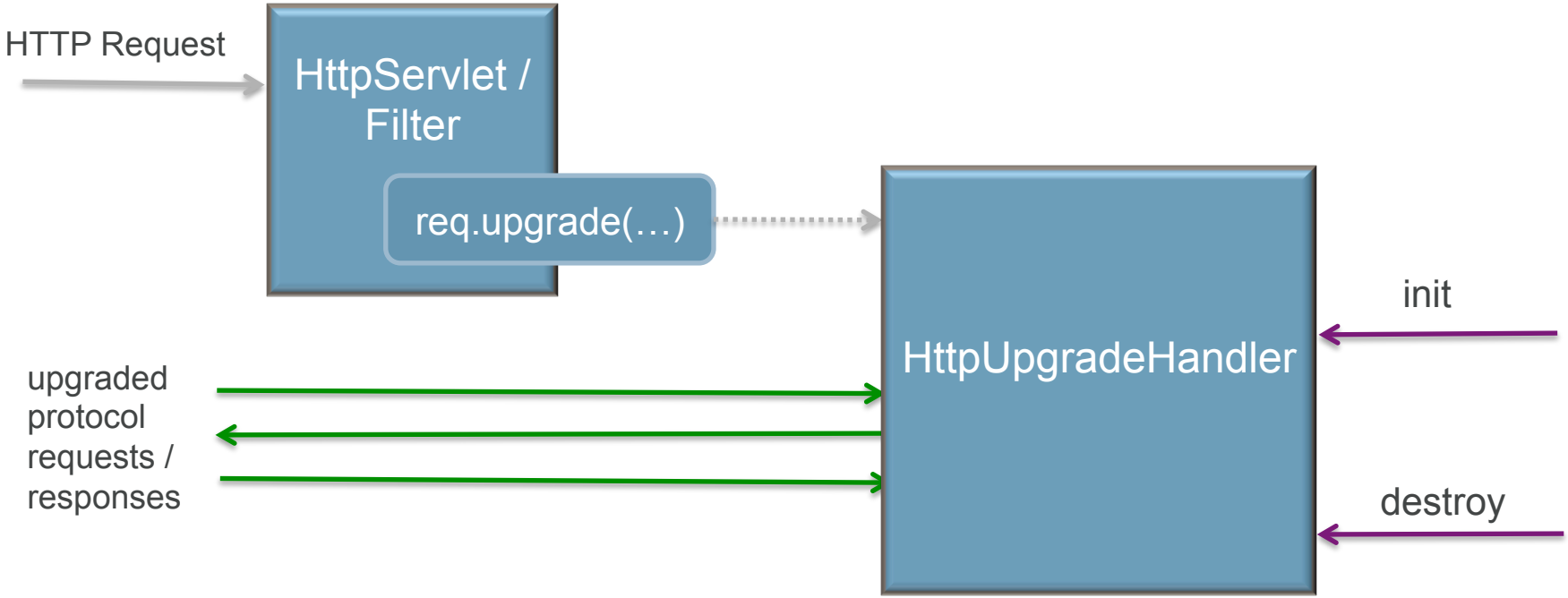
- New interface **`javax.servlet.http.HttpUpgradeHandler`**
  - **`void init(WebConnection wc)`**
  - **`void destroy()`**
- New interface **`javax.servlet.http.WebConnection`** extends **`AutoClosable`**
  - **`ServletInputStream getInputStream() throws IOException`**
  - **`ServletOutputStream getOutputStream() throws IOException`**

Protocol Upgrade

HttpServletRequest

- Add a method to **`HttpServletRequest`**
  - **`<T extends HttpUpgradeHandler>`**
    `T upgrade(Class<T> handlerClass)`
    `throws IOException, ServletException`

Protocol Upgrade



Protocol Upgrade

Example

```
public class UpgradeServlet extends HttpServlet
    protected void doGet(HttpServletRequest request,
 HttpServletResponse response) throws IOException,
 ServletException {
        …
        if (decideToUpgrade) {
            EchoHttpUpgradeHandler handler =
 request.upgrade(EchoHttpUpgradeHandler.class);
        …
    }
}
```

Protocol Upgrade

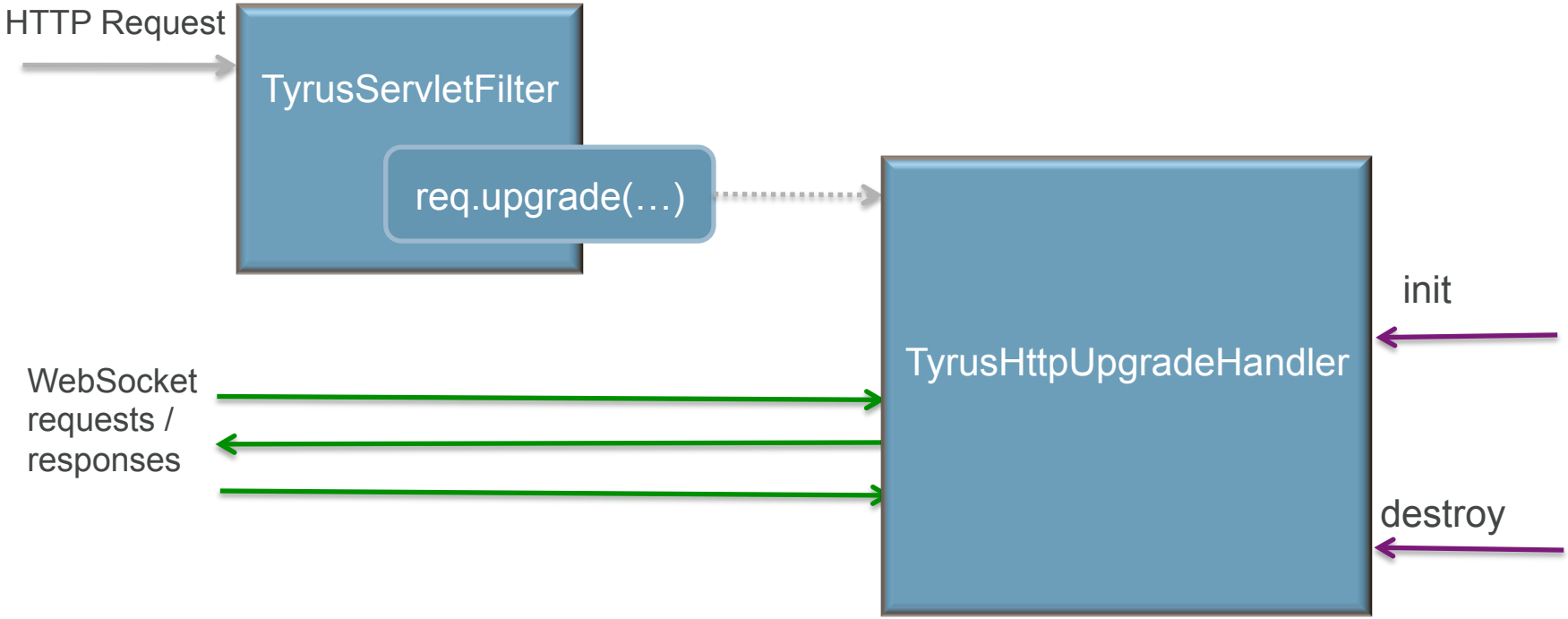
Example (cont'd)

```
public class EchoProtocolHandler implements HttpUpgradeHandler {
     public void init(WebConnection wc) {
          try {
               ServletInputStream input = wc.getInputStream();
               ServletOutputStream output = wc.getOutputStream();
               ReadListener readListener = …;
               input.setReadListener(readListener);
          …
     }

     public void destroy() {
          …
     }
}
```

Protocol Upgrade

Example 2: Reference Implementation of JSR 356, Java API for WebSocket



Security Enhancements

Session Fixation Attack

- Emails or web pages from hackers containing
  - http://abank.com?SID=ABCDEFGHIJ
- Change Session id on authentication
  - Add to interface **`HttpServletRequest`**
    - **`public String changeSessionId()`**
  - New interface **`javax.servlet.http.HttpSessionIdListener`**
    - **`void sessionIdChanged(HttpSessionEvent se, String oldSessionId)`**

Security Enhancements

Any authenticated users

- Roles “**”, any authenticated users
- For example,
  - `@WebServlet(“/foo”)`
    `@ServletSecurity(@HttpConstraint(rolesAllowed={“**”}))`

Security Enhancements

Others

- **`deny-uncovered-http-methods`** in **`web.xml`**
- Clarification on run-as
  - **`Servlet#init`**, **`Servlet#destroy`**

Miscellaneous

Overview

- **`ServletResponse#reset`** and **`#setCharacterEncoding`**
- **`HttpServletResponse#sendRedirect`**
  - //anotherhost.com/b/a.jsp (Network Path Reference)
- Add Generic
  - **`ServletRequestWrapper#isWrapperFor(Class<?> c)`**
  - **`ServletResponseWrapper#isWrapperFor(Class<?> c)`**
  - **`HandlesTypes#value`** return **`Class<?>[]`**

Miscellaneous

Overview (cont'd)

- Add method `javax.servlet.http.Part#getSubmittedFileName()`
- Add method `ServletContext#getVirtualServerName()`

- Add method `ServletRequest#getContentLengthLong()`
- Add method `ServletResponse#setContentLengthLong(long len)`

Resources

- JavaOne Shanghai 2013, CON 1387,
  What is new in JSR 340, Servlet 3.1?
  - Tue. July 23, 4:30 pm - 5:30 pm, Expo Centre – Room 430
  - Wed. July 24, 10:15 am – 11:15 am, Expo Centre – Room 422/423
- Spec and Javadoc
  - http://jcp.org/en/jsr/detail?id=340
  - http://servlet-spec.java.net
- GlassFish 4.0
  - http://glassfish.java.net
  - webtier@glassfish.java.net
- My blog
  - http://www.java.net/blog/swchan2

Java™ ORACLE®


Java™