



Getting started with Winstone

Minimal servlet container



What is Winstone?

Winstone is a small servlet container, consisting of a single JAR file.

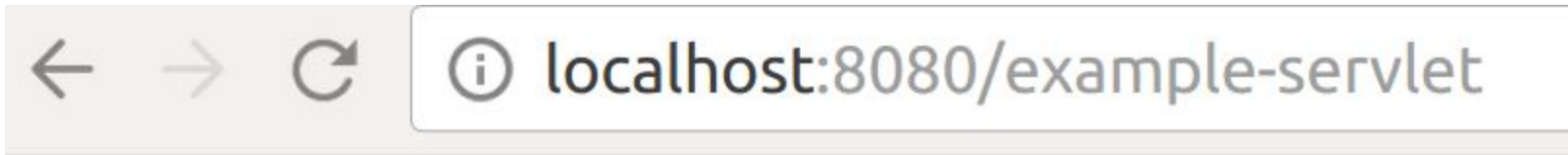
You can run Winstone on your computer using Java, and get a Servlet up and running very fast.

Running Winstone will start a server listening on port 8080 on localhost, which will forward requests to servlets, according to your settings and configuration.

<https://sourceforge.net/projects/winstone/files/latest/download?source=files>

Small example

```
$ java -jar winstone-0.9.10.jar --webroot webroot  
[Winstone 2017/01/04 17:16:37] - AJP13 Listener started: port=8009  
[Winstone 2017/01/04 17:16:37] - HTTP Listener started: port=8080  
[Winstone 2017/01/04 17:16:37] - Winstone Servlet Engine v0.9.10 running:  
controlPort=disabled
```



Hello from a servlet!

The directory layout

```
.  
|-- webroot  
|   |-- WEB-INF  
|       |-- classes  
|           |-- ExampleServlet.class  
|           |-- ExampleServlet.java  
|           |-- lib  
|           |-- web.xml  
|-- winstone-0.9.10.jar
```

```
$ java -jar winstone-0.9.10.jar --webroot webroot
```

The web.xml configuration file

```
.
|-- webroot
|   `-- WEB-INF
|       |-- classes
|       |   |-- ExampleServlet.class
|       |   `-- ExampleServlet.java
|       |-- lib
|       `-- web.xml
`-- winstone-0.9.10.jar
```

```
$ java -jar winstone-0.9.10.jar --webroot webroot
```

The web.xml configuration file - name and class

```
$ cat webroot/WEB-INF/web.xml
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
```

```
<servlet>
```

```
  <servlet-name>example</servlet-name>
```

```
  <servlet-class>ExampleServlet</servlet-class>
```

```
</servlet>
```

```
<servlet-mapping>
```

```
  <servlet-name>example</servlet-name>
```

```
  <url-pattern>/example-servlet</url-pattern>
```

```
</servlet-mapping>
```

```
</web-app>
```

The web.xml configuration file - mapping

```
$ cat webroot/WEB-INF/web.xml
<?xml version="1.0" encoding="utf-8"?>

<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">

  <servlet>
    <servlet-name>example</servlet-name>
    <servlet-class>ExampleServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>example</servlet-name>
    <url-pattern>/example-servlet</url-pattern>
  </servlet-mapping>
</web-app>
```

The web.xml configuration file - URL pattern

```
$ cat webroot/WEB-INF/web.xml
```

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
```

```
<servlet>
```

```
  <servlet-name>example</servlet-name>
```

```
  <servlet-class>ExampleServlet</servlet-class>
```

```
</servlet>
```

```
<servlet-mapping>
```

```
  <servlet-name>example</servlet-name>
```

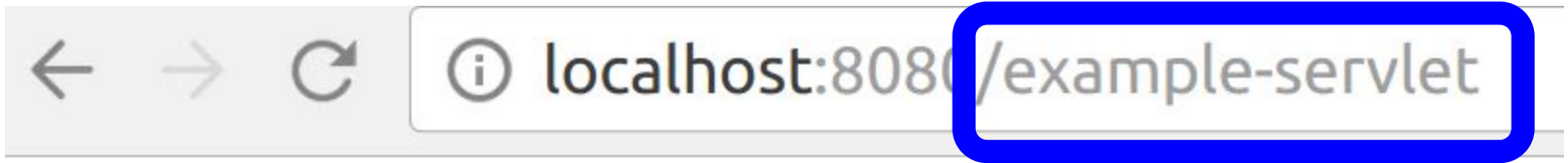
```
  <url-pattern>/example-servlet</url-pattern>
```

```
</servlet-mapping>
```

```
</web-app>
```


Mapping the request to the servlet

`<url-pattern>/example-servlet</url-pattern>`



Hello from a servlet!

The example servlet

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.PrintWriter;
import java.io.OutputStreamWriter;
import static java.nio.charset.StandardCharsets.UTF_8;
import java.io.IOException;

public class ExampleServlet extends HttpServlet{
    @Override
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException{
        PrintWriter out =
            new PrintWriter(new OutputStreamWriter(response.getOutputStream(),
                UTF_8), true);
        out.println("<html><head><title>Example servlet</title></head>");
        out.println("<body><p>Hello from a servlet!</body>");
        out.println("</html>");
        out.close();
    }
}
```

Compiling the servlet

You need to have the servlet-api jar file on your class path.

<http://central.maven.org/maven2/javax/servlet/javax.servlet-api/3.1.0/javax.servlet-api-3.1.0.jar>

```
$ wget http://central.maven.org/maven2/javax/servlet/javax.servlet-api/3.1.0/javax.servlet-api-3.1.0.jar -O servlet-api-3.1.jar
```

```
$ javac -cp .:servlet-api-3.1.jar webroot/WEB-INF/classes/ExampleServlet.java
```

Using SQLite from a servlet

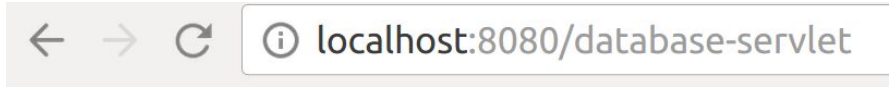
As expected, you'll need to have the JAR file from xerial with the JDBC driver for SQLite if you use any of the classes from the org.sqlite packages, when you compile.

But more specifically, you'll need to have the JAR on your class path when you run the servlet. So how does Winstone find the JAR file?

It's simple. Put it in the webroot/WEB-INF/lib/ directory.

Example with a small student database

```
.
|-- servlet-api-3.1.jar
|-- students.db
|-- webroot
|   |-- WEB-INF
|       |-- classes
|           |-- DBServlet.class
|           |-- DBServlet.java
|           |-- ExampleServlet.class
|           |-- ExampleServlet.java
|       |-- lib
|           |-- sqlite-jdbc-3.8.11.2.jar
|       |-- web.xml
|-- winstone-0.9.10.jar
```



Anna Andersson
Beata Bengtsson
Cecilia Carlsson
David Davidsson
Erik Eskilsson
Fader Fourah

The database and web.xml

```
$ echo '.schema students' | sqlite3 students.db  
CREATE TABLE students(id integer primary key not null, name text not  
null);
```

```
<servlet>  
  <servlet-name>dbexample</servlet-name>  
  <servlet-class>DBServlet</servlet-class>  
</servlet>  
<servlet-mapping>  
  <servlet-name>dbexample</servlet-name>  
  <url-pattern>/database-servlet</url-pattern>  
</servlet-mapping>
```

Servlet source code - init() method

```
public class DBServlet extends HttpServlet{

    public void init() throws ServletException { // called only once
        try{
            Class.forName("org.sqlite.JDBC");
        }catch(ClassNotFoundException cnfe){
            System.err.println("Could not load driver: "+cnfe.getMessage());
        }
    }
}
```

<https://www.tutorialspoint.com/servlets/servlets-life-cycle.htm>

<http://docs.oracle.com/javaee/5/tutorial/doc/bnafu.html>

Servlet source code - doGet() method

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException{
    PrintWriter out=new PrintWriter(new OutputStreamWriter(response.getOutputStream(),
                                                            UTF_8), true);
    out.println("<html><head><title>Example servlet</title></head><body>");
    try{
        Connection con = DriverManager.getConnection("jdbc:sqlite:students.db");
        Statement stm = con.createStatement();
        ResultSet rs = stm.executeQuery("SELECT name FROM students");
        while(rs.next()){
            out.println(rs.getString("name") + "<br />");
        }
    }catch(SQLException sqle){ out.println("Database error: " + sqle.getMessage()); }
    out.println("</body></html>");
    out.close();
}
```


Servlet import statements

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.PrintWriter;
import java.io.OutputStreamWriter;
import static java.nio.charset.StandardCharsets.UTF_8;
import java.io.IOException;
import java.sql.*;
```

Bonus - calling the Servlet from Java

```
$ java HTTPClient http://localhost:8080/database-servlet
<html><head><title>Example servlet</title></head>
<body>
Anna Andersson<br />
Beata Bengtsson<br />
Cecilia Carlsson<br />
David Davidsson<br />
Erik Eskilsson<br />
Fader Fourah<br />
</body>
</html>
```

HTTPClient.java

```
import java.io.*;
import java.net.*;
public class HTTPClient{
    public static void main(String[] args){
        HTTPClient client = new HTTPClient();
        try{
            client.getFile(args[0]);
        }catch(IOException ioe){ioe.printStackTrace();}
    }
    public void getFile(String addr) throws IOException {
        URL url = new URL(addr);
        HttpURLConnection connection = (HttpURLConnection) url.openConnection();
        connection.setRequestMethod("GET");
        BufferedReader in = new BufferedReader(new InputStreamReader(connection.getInputStream()));
        String line;
        while ((line = in.readLine()) != null) {
            System.out.println(line);
        }
        in.close();
    }
}
```

HttpClient.java - The getFile() method

```
public void getFile(String addr) throws IOException {
    URL url = new URL(addr);
    HttpURLConnection connection = (HttpURLConnection) url.openConnection();
    connection.setRequestMethod("GET");
    BufferedReader in =
        new BufferedReader(new InputStreamReader(connection.getInputStream()));
    String line;
    while ((line = in.readLine()) != null) {
        System.out.println(line);
    }
    in.close();
}
```

Show me the code!

<https://github.com/progund/java-web>