# Jython webapp for DSpace

## Basic Jython webapp

1. Create the webapp directory (you may use any name you want):
   `mkdir [dspace]/webapps/jython/`
   Tip: The location `[dspace]/webapps/jython/` is used just to illustrate that the jython webapp is just another webapp like other DSpace webapps. It's possible to choose a different location - in fact, it's preferable because the `[dspace]/webapps/` directory is replaced every time you run "`ant update`" (the old webapps directory will not be deleted, it will be renamed to "webapps-[timestamp]").
2. Download the latest Jython installer jar (e.g. `jython-installer-2.7.0.jar`) from http://www.jython.org/downloads.html
3. Get `jython.jar` and the `Lib` directory.
   a. either unzip the installer jar:
      `unzip -d [dspace]/lib/ jython-installer-2.7.0.jar jython.jar 'Lib/*'`
      `unzip -d [dspace]/webapps/jython/WEB-INF/lib/ jython-installer-2.7.0.jar jython.jar 'Lib/*'`
   b. or use it to install Jython:
      `java -jar jython-installer-2.7.0.jar --console`
      Note: Installation location doesn't matter, this is not necessary for DSpace. You can safely delete it after you retrieve `jython.jar` and `Lib`
4. Associate .py files with Jython's PyServlet

**[dspace]/webapps/jython/WEB-INF/web.xml**

```
<web-app>
    <servlet>
        <servlet-name>PyServlet</servlet-name>
        <servlet-class>org.python.util.PyServlet</servlet-class>
        <load-on-startup>1</load-on-startup>
    </servlet>
    <servlet-mapping>
        <servlet-name>PyServlet</servlet-name>
        <url-pattern>*.py</url-pattern>
    </servlet-mapping>
</web-app>
```

5. Create a Hello World servlet:

**[dspace]/webapps/jython/hello.py**

```
# -*- coding: utf-8 -*-
from javax.servlet.http import HttpServlet


class hello(HttpServlet):
    def doGet(self, request, response):
        self.doPost(request, response)

    def doPost(self, request, response):
        toClient = response.getWriter()
        toClient.println('Hello World!')
```

## Access to DSpace classes from Jython

6. Copy DSpace jars to the jython webapp's lib directory:
   `cp -r [dspace]/lib/* [dspace]/webapps/jython/WEB-INF/lib/`
7. Start up DSpace kernel on webapp startup and point it to your DSpace configuration:

**[dspace]/webapps/jython/WEB-INF/web.xml**

```
<web-app>
    ...
    <!-- DSpace Configuration Information -->
    <context-param>
        <param-name>dspace-config</param-name>
        <param-value>/dspace/config/dspace.cfg</param-value>
    </context-param>
    <!-- new ConfigurationService initialization for dspace.dir -->
    <context-param>
        <description>The location of the main DSpace configuration file</description>
        <param-name>dspace.dir</param-name>
        <param-value>/dspace</param-value>
    </context-param>
    <listener>
        <listener-class>org.dspace.app.util.DSpaceContextListener</listener-class>
    </listener>
    <listener>
        <listener-class>org.dspace.servicemanager.servlet.DSpaceKernelServletContextListener</listener-
class>
    </listener>
</web-app>
```

8. Define the context in Tomcat's configuration. There are several ways how you can do that, so just use the same way you use for configuring DSpace contexts. The recommended one is to use a context fragment:

**sudo vim /etc/tomcat7/Catalina/localhost/jython.xml**

```
<Context docBase="/dspace/webapps/jython" reloadable="true" cachingAllowed="false" />
```

A few seconds after you save the file, Tomcat will notice it and load the "jython" context.

## Adding Java libraries

1. Copy the .jar to `/dspace/webapps/jython/WEB-INF/lib/`
2. Reload the context
   `sudo touch /etc/tomcat7/Catalina/localhost/jython.xml`

Tip: If you forgot which libraries you added (because it's hard to spot them among dozens of libraries which belong to DSpace), here's how you can filter out the DSpace libraries, which should leave you only with a list of libraries you added:

```
ls -1 /dspace/webapps/jython/WEB-INF/lib > /tmp/jython.txt
ls -1 /dspace/lib > /tmp/dspace.txt
diff -u /tmp/jython.txt /tmp/dspace.txt | view -
```

## Adding Python libraries

Python libraries can either be added to `/dspace/webapps/jython/WEB-INF/lib/Lib/` or to context root (`/dspace/webapps/jython/`).

## Creating nice action URLs

You may find a snipped like the one below to set up URL mapping for a Jython servlet. Unfortunately, it's not implemented in PyServlet, as it is more a demo than something to use in production (see this thread). For production deployment, Modjy is recommended.

**web.xml**

```
<web-app>
    ...
    <servlet-mapping>
        <servlet-name>SherpaRomeo</servlet-name>
        <url-pattern>/SherpaRomeo</url-pattern>
    </servlet-mapping>
</web-app>
```

## See also

- Curation tasks in Jython