# Persisting Sessions

## Persisting Sessions

It can sometimes be useful to be able to preserve existing Sessions across restarts of Jetty. The org.mortbay.jetty.servlet.HashSessionManager now supports this feature. If persistence is enabled, the HashSessionManager will save all existing, valid Sessions to disk before shutdown completes. On restart, the saved Sessions are restored.

### Enabling Persistence

A SessionManager does just what it's name suggests - it manages the lifecycle and state of Sessions on behalf of a webapp. Each webapp must have it's own unique SessionManager instance. Enabling persistence is as simple as configuring the HashSessionManager as the SessionManager for a webapp and telling it where on disk to store the sessions:

```
<Configure
class="org.mortbay.jetty.webapp.WebAppContex
t">

 .
 .
 .
 <Set name="sessionHandler">
   <New
class="org.mortbay.jetty.servlet.SessionHand
ler">
     <Arg>
       <New
class="org.mortbay.jetty.servlet.HashSession
Manager">
         <Set
name="storeDirectory">your/chosen/directory/
goes/here</Set>
       </New>
     </Arg>
   </New>
 </Set>
 .
 .
 .
</Configure>
```

> ✅ **"Reminder"**
>
> Don't forget that if you want to persist the sessions from multiple webapps, you'll need to configure a separate HashSessionManager for each, and naturally each should have a different value for **storeDirectory**.

The above example uses a configuration file suitable for the Context Deployer - you might want to check out the documentation for it.

### Delaying Session Load

Sometimes you may need to ensure that the sessions are loaded AFTER the servlet environment has been started up (by default, sessions will be eagerly loaded as part of the container startup, but before the servlet environment has been initialized). For example, the Wicket web framework requires the servlet environment to be available when sessions are activated.

Using `SessionManager.setLazyLoad(true)`, sessions will be loaded lazily either when the first request for a session is received, or the session scavenger runs for the first time, whichever happens first. Here's how the configuration looks in xml:

```
<Set name="sessionHandler">
   <New
class="org.mortbay.jetty.servlet.SessionHand
ler">
     <Arg>
       <New
class="org.mortbay.jetty.servlet.HashSession
Manager">
         <Set name="lazyLoad">true</Set>
       </New>
     </Arg>
   </New>
 </Set>
```

## Enabling Persistence for the Maven Jetty Plugin

To enable session persistence for the maven jetty plugin, set up the HashSessionManager in the <configuration> section like so:

```
<plugin>
        <groupId>org.mortbay.jetty</groupId>

<artifactId>maven-jetty-plugin</artifactId>
        <version>6.1.6</version>
        <configuration>

<scanIntervalSeconds>1</scanIntervalSeconds>
          <webAppConfig
implementation="org.mortbay.jetty.plugin.Jet
ty6PluginWebAppContext">
            <contextPath>/foo</contextPath>
            .
            .
            .
            <sessionHandler
implementation="org.mortbay.jetty.servlet.Se
ssionHandler">
              <sessionManager
implementation="org.mortbay.jetty.servlet.Ha
shSessionManager">

<storeDirectory>${basedir}/target/your/sessi
ons/go/here</storeDirectory>
              </sessionManager>
            </sessionHandler>
            .
            .
            .
```

```
            </webAppConfig>
        </configuration>
</plugin>
```