Runtime Infrastructure - Feature #5155

make it easier to deploy custom servlets into the FWD server

02/22/2021 07:29 AM - Greg Shah

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
billable:	No	**vendor_id:**	GCD

Description

History

#2 - 02/22/2021 07:45 AM - Greg Shah

Our current approach to loading web applications/servlets into the FWD server's embedded container current relies on the following process.

1. The Java servlet code for should be packaged as a .war ("web archive").
2. In the "standard project template", the .war file needs to be placed in the deploy/lib/ folder. This needs to happen before the FWD server is started and before any code is copied from there to make a deployment.
3. The servlet code must be loaded into the FWD server at runtime. The FWD server provides a Server Initialization and Termination Hook that provides this capability.
   - Implement the "loader" using these hooks, in a Java class which is then included into the application jar file.
   - The loader should look something like this: EXAMPLEEXAMPLE

```
package com.org.yourapplication.webservices;

import java.io.*;
import java.net.*;
import java.util.*;

import javax.servlet.*;

import org.eclipse.jetty.annotations.*;
import org.eclipse.jetty.plus.webapp.*;
import org.eclipse.jetty.server.session.*;
import org.eclipse.jetty.webapp.*;

import com.goldencode.p2j.main.*;
import com.goldencode.p2j.util.Utils;
import com.goldencode.p2j.web.*;

public class YourIncredibleLoader
implements InitTermListener
{
   private List<GenericWebServer> servers = new ArrayList<>();

   @Override
   public void initialize()
   {
      synchronized (servers)
      {
         servers.add(GenericWebServer.initializeWebApp("first", "one.war", "Nice API"));
         servers.add(GenericWebServer.initializeWebApp("second", "two.war", "Excellent Callbacks"));
         servers.add(GenericWebServer.initializeWebApp("third", "three.war", "Awesome Web Service"));
      }
   }

   @Override
   public void terminate(Throwable t)
   {
      synchronized (servers)
      {
         servers.forEach(server ->
```

```
         {
            try
            {
               server.shutdown();
            }
            catch (Exception e)
            {
               // TODO:
               e.printStackTrace();
            }
         });
      }
   }
}
```

- Register the hook class in the directory.xml.
- The result will be that the new websocket servlet will be loaded into the FWD server when it starts and will be available for the lifetime of the server.

4. The .war files must currently have a web.xml for configuration of the services.

I think this approach should be made much easier.

- We could implement a simple list of web services to load in the directory, eliminating the need for the init/term hooks which are an unnecessary complication. We already have to make edits to the directory for the hooks, the hooks themselves don't add any value.
- We should support annotations for configuration, eliminating the requirement for the web.xml.

The use of web.xml bothers me a bit. I think there are significant security exposures by using it. My sense is that we should be in much tighter control over the initialization in order to minimize these potential impacts.

#3 - 02/23/2021 10:07 AM - Constantin Asofiei

To allow mixed annotations and web.xml configuration for the services (servlet, web socket), the web.xml needs to have metadata-complete="false" version="3.0" set at its web-app node.

Otherwise, I agree that we should have a customWebServices section in directory.xml, where the .war and any other requirements are configured and managed, without the need of FWD server hooks. I'm thinking that the context parameters required by the WebAuthFilter may be able to be configured directly in the directory, and 'injected' via the WebAppContext to be made available as real context parameters to the servlet/filters. This would eliminate the need to have FWD-specific configuration in web.xml.