

Runtime Infrastructure - Feature #4405

provide the capability to deliver static HTML and resources via the FWD server's Jetty

11/12/2019 10:35 AM - Greg Shah

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:	Sergey Ivanovskiy	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		vendor_id:	GCD
billable:	No		
Description			
Related issues:			
Related to Runtime Infrastructure - Feature #5262: add access control for res...			New

History

#1 - 11/12/2019 10:39 AM - Greg Shah

Customers using FWD for a web based application often have static HTML (and other non-HTML) resources which need to be delivered securely to the browser. Examples of non-HTML resources would include javascript files, images.

We need the ability for customers to configure the Jetty environment to deliver these resources in a flexible and secure manner. We probably will need to include some integration with session level web authentication in case some resources only should be delivered to a subset of clients. This would also require a SecurityManager plugin for managing the security decisions.

#2 - 03/05/2021 02:10 PM - Greg Shah

- Assignee set to Hynek Cihlar

We need to handle this sooner rather than later.

Hynek: What would you estimate is the effort for this?

#4 - 04/12/2021 04:42 PM - Greg Shah

I think the initial approach would just allow access to FWD server resources without the security limits. It would allow 1 or more mappings to be configured in the directory. I think that each mapping would need to define:

- The URL <path> that is being exposed (https://host:port/<path>/<resource>).
- The location of the resources on the server. This could be a directory in the file system OR a location in the applicaton jar.

I expect that most such resources would be built into the application jar to ease deployment. We can implement the security features as a 2nd phase.

#5 - 04/13/2021 02:56 AM - Sergey Ivanovskiy

Is this understanding correct that these static resources are managed by the existing FWD web server that manages web clients? Should it be an independent dedicated web server with its own web certificates?

#6 - 04/13/2021 03:46 AM - Sergey Ivanovskiy

Planning to add these settings to the directory under the default server node

```
<node class="container" name="web-static-content">
  <node class="container" name="web1">
    <node class="string" name="context">
      <node-attribute name="value" value="/app/help"/>
    </node>
    <node class="string" name="resource">
      <node-attribute name="value" value="/opt/client/help/HTML"/>
    </node>
  </node>
</node>
<node class="container" name="system-help">
  <node class="string" name="context">
    <node-attribute name="value" value="/app/help"/>
  </node>
</node>
```

The nodes under web-static-content hold the mapping of static application contexts to their storage locations within the application jar or the file system on the server. The node application-help-context holds the application help context. Are these generic settings enough for this task?

#7 - 04/13/2021 05:33 AM - Sergey Ivanovskiy

I forgot to add welcome files. Thus the directory settings can be looked like

```
<node class="container" name="web-static-content">
  <node class="container" name="web1">
    <node class="string" name="context">
      <node-attribute name="value" value="/app/help"/>
    </node>
    <node class="string" name="resource">
      <node-attribute name="value" value="/opt/client/help/HTML"/>
    </node>
    <node class="string" name="file">
      <node-attribute name="value" value="index.html"/>
    </node>
  </node>
</node>
```

#8 - 04/13/2021 07:58 AM - Greg Shah

Is this understanding correct that these static resources are managed by the existing FWD web server

Use the existing Jetty in the FWD Server.

that manages web clients?

The FWD Server Jetty is used for login to the web clients, but the actual web client is in the FWD Client JVM, which is different and has its own Jetty.

Should it be an independent dedicated web server with its own web certificates?

It already exists and has its certificates, you don't have to do anything. Also, the FWD Client Jetty instances are setup with the same certificate as the server, but I don't think that matters.

For #4204 you are loading a new page from these static resources and that page will have its own origin which will be different from the web client's origin. But that only matters if there is some communication between the web client and the help page. If that is needed we can discuss the plan in #4204. It won't change how the static resources work.

#9 - 04/13/2021 08:15 AM - Greg Shah

The nodes under web-static-content hold the mapping of static application contexts to their storage locations within the application jar or the file system on the server.

This is fine.

The node application-help-context holds the application help context.

I don't understand why this is needed. Can't we create a new context (if needed) for each element in the web-static-content?

Are these generic settings enough for this task?

Yes.

#10 - 04/13/2021 08:16 AM - Greg Shah

Thus the directory settings can be looked like

Yes, this is good. How are you going to handle the jar resource case?

#11 - 04/13/2021 08:20 AM - Sergey Ivanovskiy

I am going to use the existing functionality of FileSystemOps

```
FileSystemOps.JarResource jarEntry = FileSystemOps.searchResourceJars(resource, null);
URL urlJarResource = null;
if (jarEntry != null)
{
    urlJarResource = jarEntry.getLoader().getResource(jarEntry.getName());
}

Resource targetResource = null;

if (urlJarResource != null)
{
    targetResource = Resource.newResource(urlJarResource);
}
else
{
    targetResource = Resource.newResource(resource);
}
```

#12 - 04/13/2021 08:48 AM - Greg Shah

OK, but make sure it is separated from FileSystemOps. That code calls down to the FWD client and uses a legacy 4GL security context. For this usage, we should NOT be calling anything that requires such an environment.

#13 - 04/17/2021 08:43 AM - Greg Shah

- Assignee changed from Hynek Cihlar to Sergey Ivanovskiy

- % Done changed from 0 to 70

As part of the changes for #4204, Sergey implemented "public" static HTML resources. The changes can be seen in 3821c revisions 12280, 12284, 12286, 12289, 12295.

The resources can reside in the application's jar file. The ResourceHandler code will fallback to the server file system if the configured path is not found in the jar. In itself, this is correct. Please confirm the following:

- Is it possible to "escape" from the resource base directory, possibly by using relative paths that include ../?
- If the configured path is not an absolute filesystem location (e.g. starts with /), then is it correct that the ResourceHandler will look for the relative path under the server's working directory and in the classpath?

#14 - 04/17/2021 08:58 AM - Greg Shah

Please document how to configure the web-static-content section of the directory in the [Web Server](#) wiki page. You can pattern the page after [4GL Runtime Compatibility Options](#).

#15 - 04/17/2021 09:31 AM - Greg Shah

- Related to Feature #5262: add access control for resources exposed via web services/protocols using the SecurityManager added

#16 - 04/17/2021 10:07 AM - Greg Shah

I've put together a simple page to describe the feature [Static Web Content](#). It links to the directory configuration page which needs content.

Please make edits to my simple page as well, if you see that something is missing or is incorrect.

#17 - 04/19/2021 11:27 AM - Sergey Ivanovskiy

Greg Shah wrote:

As part of the changes for #4204, Sergey implemented "public" static HTML resources. The changes can be seen in 3821c revisions 12280, 12284, 12286, 12289, 12295.

The resources can reside in the application's jar file. The ResourceHandler code will fallback to the server file system if the configured path is not found in the jar. In itself, this is correct. Please confirm the following:

- Is it possible to "escape" from the resource base directory, possibly by using relative paths that include ../?

ContextHandler.getResource and ResourceHandler.getResource returns null if the canonical path of the provided input doesn't start with /. The canonical path is built by org.eclipse.jetty.util.URIUtil

```
/* ----- */
/** Convert a path to a canonical form.
 * All instances of "." and ".." are factored out. Null is returned
 * if the path tries to .. above its root.
 * @param path the path to convert
```

```

    * @return path or null.
    */
    public static String canonicalPath(String path)

```

This is the responsibility of web server implementation to cut down such requests.

- If the configured path is not an absolute filesystem location (e.g. starts with /), then is it correct that the ResourceHandler will look for the relative path under the server's working directory and in the classpath?

ResourceHandler is responsible for incoming requests processing, but the Resource (JarResource, FileResource, PathResource, URLResource) class represents the target resource delivered by the web server. The requested document by this url `https://localhost:7443/app/help/index.html` is `index.html` iff the context path is `/app/help/` but not this one `/app/help/index.html`. So the path of the target requested document is added to the base resource path of the ResourceHandler.

#18 - 04/19/2021 11:49 AM - Sergey Ivanovskiy

Please look at this part of `WebServer.getWebContentHandlers`

```

        try
        {
            Function<JarClassLoader, Function<String, String>> searchFn = jcl -> (n -> jcl.getResourceName(n,
true));
            Utils.JarResource jarEntry = Utils.searchResourceJars(resource, searchFn, null);
            URL urlJarResource = null;
            if (jarEntry != null)
            {
                urlJarResource = jarEntry.getLoader().getResource(jarEntry.getName());
            }
            Resource targetResource = null;

            if (urlJarResource != null)
            {
                targetResource = Resource.newResource(urlJarResource);
            }
            else
            {
                targetResource = Resource.newResource(resource);
            }

            if (targetResource == null || !targetResource.exists())
            {
                LOG.log(Level.WARNING,
                    "web-static-content/resource entry not found from this path [" + resource + "]);
                continue;
            }
            webHandler.setBaseResource(targetResource);
        }
        catch (IOException e)
        {
            LOG.log(Level.SEVERE, e.getMessage());
        }

```

The `Resource.newResource(resource)` and `Resource.newResource(urlJarResource)` delegate(use) to `Resource.newResource(URL url, boolean`

useCaches) that builds PathResource, JarResource, JarFileResource, or URLResource based on the input URL.

#19 - 04/22/2021 10:10 AM - Greg Shah

OK. I will close this task when the documentation for [#4405-14](#) is complete.

#20 - 05/25/2021 03:22 PM - Greg Shah

Sergey: Please edit the [Web Server](#) page to clarify these questions:

- What is web1? Does the name matter?
- Are the values case-sensitive?
- Please explain where the resources would reside on the server or in the jar. In particular, to what location are these values relative or are they absolute paths?

#21 - 05/26/2021 07:37 AM - Sergey Ivanovskiy

Greg Shah wrote:

Sergey: Please edit the [Web Server](#) page to clarify these questions:

Please review the updated wiki page.

- What is web1? Does the name matter?

This name can be any string identifier.

- Are the values case-sensitive?

Yes.

- Please explain where the resources would reside on the server or in the jar. In particular, to what location are these values relative or are they absolute paths?

Values for context and resource are absolute paths starting from the root /. The resources can be in the application jar or on the server file system.

#22 - 05/26/2021 07:48 AM - Greg Shah

- % Done changed from 70 to 100

- Status changed from New to Closed