Development-Custom Resource Manager

Petite Abeille

Here is a little example on how to write your own WOResourceManager:

The main methods to implement are urlForResourceNamed() and bytesForResourceNamed().

- urlForResourceNamed simply build an URL for a resource from wherever you would like (eg a jar file). Unfortunately, urlForResourceNamed uses pathForResourceNamed so you will need to rewrite that also:

```
 private URL urlForResourceNamed(String aResourceName)
 {
   return this.getClass().getResource( aResourceName );
 }

 public String pathForResourceNamed(String aResourceName, String
aFrameworkName, NSArray someLanguages)
 {
   URL  anURL = this.urlForResourceNamed( aResourceName );

   if ( anURL != null )
   {
     return anURL.toExternalForm();
   }

   return null;
 }

 public String urlForResourceNamed(String aResourceName, String
aFrameworkName, NSArray someLanguages, WORequest aRequest)
 {
   String  anURL = super.urlForResourceNamed( aResourceName,
aFrameworkName, someLanguages, aRequest );

   this.bytesForResourceNamed( aResourceName, aFrameworkName, someLanguages
);

   return anURL;
 }
```

- bytesForResourceNamed simply retrieve a resource from wherever you would like (eg a jar file):

```
public InputStream inputStreamForResourceNamed(String aResourceName, String
aFrameworkName, NSArray someLanguages)
 {
   return this.getClass().getResourceAsStream( aResourceName );
 }
```

```
 public byte[] bytesForResourceNamed(String aResourceName, String
aFrameworkName, NSArray someLanguages)
 {
   if ( aResourceName != null )
   {
     URL  anURL = this.urlForResourceNamed( aResourceName );

     if ( anURL != null )
     {
       String      aKey = anURL.toString();
       WOURLValuedElementData  anElement = (WOURLValuedElementData)
_cache.get( aKey );

       if ( anElement == null )
       {
         synchronized( this )
         {
           InputStream  anInputStream = this.inputStreamForResourceNamed(
aResourceName, null, null );

           if ( anInputStream != null )
           {
             try
             {
               InputStream  aBufferStream = new BufferedInputStream(
anInputStream );
               byte[]    someBytes = new byte[ aBufferStream.available() ];

               aBufferStream.read( someBytes );
               aBufferStream.close();
               anInputStream.close();

               {
                 Data  someData = new Data( someBytes );
                 String  aType = this.contentTypeForResourceNamed( aKey );

                 anElement = new WOURLValuedElementData( someData, aType,
aKey );

                 _cache.put( aKey, anElement );
               }
             }
             catch(Exception anException)
             {
               SZLog.warning( anException );
             }
           }
         }
       }

       return ( (Data) anElement.data() ).bytesNoCopy();
     }
```

```
      SZLog.debug( "Null url for resource named '" + aResourceName + "'." );

      return null;
    }

    throw new IllegalArgumentException (
"ResourceManager.bytesForResourceNamed: null resource name." );
```

```
  }
```

- Last but not least, you need to take care of those funky WOURLValuedElementData so dataForResourceNamed will work:

```
public NSData dataForResourceNamed(String aResourceName)
 {
   this.bytesForResourceNamed( aResourceName, null, null );

   String      aKey = this.urlForResourceNamed( aResourceName ).toString();
   WOURLValuedElementData  anElement = (WOURLValuedElementData) _cache.get(
aKey );

   return anElement.data();
 }
```

- Finally, you need to register your resource manager with WOApplication:

```
  anApplication.setResourceManager( new ResourceManager() );
```

Handling of languages and frameworks are left as an exercise to the reader.