

PluggingIntoStruts

Plugging into Struts

HiveMind does offer a servlet filter for initializing the registry but if your servlet engine does not support filters or you want to control the registry from Struts then a plugin is the way to go. This is not an example of how to configure HiveMind but how to configure a plugin to use HiveMind. It's a quick and easy way to get HiveMind into your Struts application.

The Interface (StrutsHive.java)

```
public interface StrutsHive
{
    public static final String STRUTSHIVE_REGISTRY = "org.apache.struts.hivemind.STRUTSHIVE_REGISTRY";

    Object lookup( String name, Class type );
    Object lookup( Class type );

    List getConfiguration( String configuration );
    void cleanupThread();
}
```

The **PlugIn** (StrutsHivePlugIn.java)

```
public class StrutsHivePlugIn implements StrutsHive, PlugIn
{
    private static final Log _log = LogFactory.getLog( StrutsHivePlugIn.class );

    private Registry _registry;
    private String _errorHandler;
    private String _classResolver;

    private String _language;
    private String _country = "";
    private String _variant = "";

    public void setLanguage(String language)
    {
        this._language = language;
    }

    public void setCountry(String country)
    {
        this._country = country;
    }

    public void setVariant(String variant)
    {
        this._variant = variant;
    }

    public void setErrorHandler( String errorHandler )
    {
        _errorHandler = errorHandler;
    }

    public void setClassResolver( String classResolver )
    {
        _classResolver = classResolver;
    }

    public void init(ActionServlet actionServlet, ModuleConfig moduleConfig) throws ServletException
    {
        ClassResolver resolver;
        RegistryBuilder builder;
```

```

if( _log.isDebugEnabled() )
{
    _log.debug("Initializing StrutsHivePlugIn.");
}

if( _classResolver == null )
{
    if( _log.isDebugEnabled() )
    {
        _log.debug( "No class resolver was defined using HiveMinds default class resolver." );
    }

    resolver = new DefaultClassResolver();
}
else
{
    try
    {
        resolver = (ClassResolver) initializeClass( _classResolver );
    }
    catch( Exception e )
    {
        throw new ServletException( e.getMessage() );
    }
}

if( _errorHandler == null )
{
    builder = new RegistryBuilder();
}
else
{
    try
    {
        ErrorHandler errorHandler = (ErrorHandler) initializeClass( _errorHandler );
        builder = new RegistryBuilder( errorHandler );
    }
    catch( Exception e )
    {
        throw new ServletException( e.getMessage() );
    }
}

builder.processModules( resolver );
_registry = builder.constructRegistry( getLocale() );
actionServlet.getServletContext().setAttribute( STRUTSHIVE_REGISTRY, this );

if( _log.isDebugEnabled() )
{
    _log.debug("StrutsHivePlugIn initialized.");
}
}

public Object lookup( String name, Class type )
{
    if( _registry.containsService( name, type ) )
    {
        return _registry.getService( name, type );
    }

    throw new IllegalArgumentException( type.getName() + " service was not found in the registry");
}

public Object lookup( Class type )
{
    if( _registry.containsService( type ) )
    {
        return _registry.getService( type );
    }
}

```

```

        throw new IllegalArgumentException( type.getName() + " service was not found in the registry");
    }

    public List getConfiguration( String configuration )
    {
        if( _registry.containsConfiguration( configuration ) )
        {
            return _registry.getConfiguration( configuration );
        }

        throw new IllegalArgumentException( configuration + " configuration was not found in the registry");
    }

    public void cleanupThread()
    {
        _registry.cleanupThread();
    }

    public void destroy()
    {
        if( _log.isDebugEnabled() )
        {
            _log.debug("Shutting down StrutsHivePlugIn");
        }

        _registry.shutdown();
    }

    protected Locale getLocale()
    {
        if( _language != null )
        {
            return new Locale( _language, _country, _variant );
        }
        else
        {
            return Locale.getDefault();
        }
    }

    protected Object initializeClass( String type ) throws Exception
    {
        Thread thread = Thread.currentThread();
        ClassLoader loader = thread.getContextClassLoader();
        Class clazz = loader.loadClass( type );
        return clazz.newInstance();
    }
}

```

Configuration (struts-config.xml)

The plugin has multiple set-properties but all are optional

Property	Optional	Description
language	Y	define a locale different from the default
country	Y	used in conjunction with the language property but not mandatory
variant	Y	used in conjunction with the country property but not mandatory
errorHandler	Y	include your own HiveMind error handler
classResolver	Y	include your own HiveMind class resolver

```
<plug-in className="<your package name goes here>.StrutsHivePlugIn">
  <set-property property="language" value="en"/>
</plug-in>
```

Using it in your application

```
public MyAction extends Action
{
    ...

    public Object getService( String service, Class type )
    {
        StrutsHive hive = (StrutsHive) getServlet().getServletContext().getAttribute( StrutsHive.
STRUTSHIVE_REGISTRY );
        return hive.lookup( service, type );
    }

    ...
}
```

Conclusion

Not much to talk about here but if you want to use it, go nuts! If you have improvements let me know or add to the Wiki.

Contributed by [KurtHoehn](#)