The ResourceInit Handler

What is a ResourceInit handler ?

- It allows to read a different resource than the one shown in the URI
- "Normal" behaviour of OpenCms
  - The URI maps to a resource in the VFS
  - If the URI maps to a non existing file in the VFS, a 404-Error is thrown
- With a ResourceInit handler, we can use a "virtual" URI that does not exist in the VFS

The ResourceInit Handler

Alkacon

OpenCms DAYS 2009

Example:

www.myserver.com/opencms/opencms/index.html

is mapped to

/sites/default/index.html

With ResourceInit Handler the following is possible:

www.myserver.com/opencms/opencms/special/index.html

is mapped to

/sites/default/internal/specialfolder/index.html

But the URI keeps the same!

The ResourceInit Handler

Implementation & Configuration

- ResourceInit Handler must implement the interface *org.opencms.main.I_CmsResourceInit* and the method *initResource*

- Is configured in the opencms-system.xml:

```
<resourceinit>
...
<resourceinithandler class="com.alkacon.MyInitHandler"/>
...
</resourceinit>
```

Implementation & Configuration

- Multiple ResourceInit handlers can configured
- Handlers will executed in the order of the configuration
- ResourceInit hander analyses the URI and returns a CmsResource
- Each handler should have an exclusive trigger in the URI which signals that it should be used

- Important: Each request into OpenCms will run though all ResourceInit handler, so take care what you implement in them: PERFORMANCE!

The ResourceInit Handler

Code example:

```
public class MyInitHandler implements I_CmsResourceInit {
    /** trigger path. */
    public static final String HANDLER_PATH = "/special/";

    public CmsResource initResource(
        CmsResource resource,
        CmsObject cms,
        HttpServletRequest request,
        HttpServletResponse response) throws CmsResourceInitException {
        // only process this handler if there is a real request
        if (request != null) {
            String uri = cms.getRequestContext().getUri();
           // only process those requests that start with the trigger path
            if (uri.startsWith(HANDLER_PATH)) {
                // extract the uri of the resource we really wanted to read
                uri = uri.substring(HANDLER_PATH.length(), uri.length());
                // do your implementation here
            }
        }
        return resource;
    }
}
```