

Effective Exceptions

Rethinking the SlingException

Status: IMPLEMENTED
Created: 23. December 2007
Author: fmeschbe

Inspired by Barry Ruzek's article [Effective Java Exceptions](#), I went out to revisit the exceptions we have defined in the Sling API. This is what I came out with:

- We have 4 Exceptions, all of which are checked exceptions
- The `SlingException` is a base exception and is declared almost everywhere
- The `HttpStatusCodeException` is an `IOException` not a `SlingException` and is not declared to be thrown anywhere
- We have two documented possibilities of throwing a runtime exception: The `AccessControlException` possibly thrown when accessing a resource through the `ResourceResolver`.

Thinking about these (checked) Exceptions, I propose to change the Sling API as follows:

1. The `SlingException` is a `RuntimeException` and is used as the base exception for all exceptions defined by the Sling API.
2. The `HttpStatusCodeException` is removed. Status codes are better reported back to the client using `HttpServletResponse.sendError()`.
3. Add `ResourceNotFoundException` which may be used by scripts and servlets to report a missing resource.
4. Add `QuerySyntaxException` thrown from the `ResourceResolver.findResources` and `ResourceResolver.queryResources` methods.
5. Add `ScriptEvaluationException` thrown by `SlingScript.eval` wrapping and further failure cause.
6. Drop `ServiceNotAvailableException` and the respective `ServiceLocator.getRequiredService`: The method and thus the exception are probably not very usefull. Rather the `ServiceLocator.getService` method should be used and the result checked for null.
7. Add `SlingIOException` and `SlingServletException` both extending `SlingException`. These exceptions are used to wrap `IOException` and `ServletException` instances to be able to forward them as runtime exceptions to the appropriate error handling.

This change also has an influence on the implementation of the Sling API:

1. The main point is the handling of the `SlingException` in the Sling main servlet.
2. Other locations catching `Exception` should be revisited to make sure no `SlingException` is swallowed and not treated correctly.

Another point with the Sling main servlet is, that it is important to not call error handling servlets from included requests but only from the topmost request.

The proposed API changes can be evaluated in the Sling whiteboard at http://svn.apache.org/repos/asf/incubator/sling/whiteboard/fmeschbe/effective_exceptions