Sling Feature Flags support

See also https://issues.apache.org/jira/browse/SLING-3148 for the timeline/history of this.

For now we have a basic prototype at http://svn.apache.org/repos/asf/sling/whiteboard/bdelacretaz/feature-flags/ that allows for checking if a named feature flag is active or not. This is the easy part.

What sets a feature flag

As described by Roy Fielding on the Sling dev list (http://markmail.org/message/rueoiuacmft5fdet)

> *Likewise, you should expect to have both a global setting for each flag and a per-user mask, and the person running the flip switch control panel needs to be able to choose from*
>
> *a) off for all users*
> *b) off for unassigned users*
> *c) proportional assignment to on (X out of N users, max M)*
> *d) logical (custom code) assignment to on*
> *e) on for unassigned users*
> *f) on for all users*

Feature Flags use cases

ⓘ **Work in Progress**

This is still experimental, work in progress, not final etc...just rough ideas for now.

Show/Hide Resources

Make some Resources completely invisible in specific requests, based on feature flags.

All descendant Resources of a hidden Resource are hidden as well.

Need to be careful about caching.

Alternate between resources based on feature flags

Resource A is visible only if SomeFeature is not active.
Resource B is visible only if SomeFeature is active.
A and B are never visible simultaneously.
Having to create multiple related features for that would be a pain.

This is basic A/B testing and also applies to soft launching a new feature (B) that replaces an existing one (A).

The constraints of the show/hide resources use case above also apply here.

Alter Resource rendering

Can be achieved transparently by hiding rendering scripts and servlets.

Need to be careful about caching.

Might need to happen on the client as well as on the server, depending on the desired caching behavior.

Use variants of OSGi services

A call to sling.getService("foo") in a rendering script, for example, can return different variants of the "foo" service depending on which feature flags are active.

The usage value needs to be weighed against implementation difficulty...