

Versioning

Taken from on e-mail from Eelco Hillenius on 2007-04-19:

Basically, it (`Component#.isVersioned()`) says whether a component supports back button or not when component replacement is used (and less importantly, when `setModel` is used).

If it doesn't support versioning, a stable URL will be generated when a change in hierarchy or state occurs, so that clicking the back button will take you back to the last bookmarkable or versioned page. The wizard component makes use of this, as you want users to use the wizard's buttons, not the browser's buttons.

If it does support versioning, an 'unstable' link will be created when a change occurs. An example of this is our implementations of pageable lists. How rolling back of changes is implemented depends on the session store that is being used. In 1.3's default, older instances of whole pages are serialized to second level store (temp dir on disk by default), and when rolling back, the appropriate version is read back in. But with the other session store implementation we provide with Wicket, `HttpSessionStore`, changes are recorded as delta objects in the page instance where it applies to. When the back button is used, a roll back is executed using the recorded delta objects. This is cheaper when (part of) the history is kept in RAM (which is the case with `HttpSessionStore`), but doesn't make sense for the second level cache store as we would still have to serialize the references to the page etc.

See also [Browser back & forward](#).