Zoot — design and overview document

July 1, 2010

1 About Zoot

Zoot is a minimalist cms plugin for Grails partly inspired by Comatose. The name is derived from Monty Pythons Quest for the Holy Grail. The goal of zoot is to be able to easaly create a simple CMS with a low level of irritation for developers and users.

Zoot is created as a grails plugin, and not a finished application. This means that in order to use Zoot, one must first create a Grails application and install the Zoot plugin into it.

2 Page

The Page is the central data-class in Zoot. A Page represents a page in the cms and has fields for title, slug (the last part of the URI), keywords ingress body etc.

2.1 Filters

Zoot uses the filter_type field in Page objects to determin how to edit and present the body of a Page. The currently available filter types are:

- gsp — Raw gsp. This is rendered as if it as a gsp page. Usefull if the page contains logic.
- markdown — Markdown script.
- wysiwyg html — Wysiwyg editor for html pages. This is available if the fckeditor plugin is installed in the application.

2.2 Sub-pages

Zoot pages are organized hierachically. Every Zoot-page has one parent and several children. And only one Page may have null as parent. This Page is considred the root Page.

2.3 Positioning

Zoot comes with a mechaism for sorting pages under a specific parent using the `pos` field in Page.

2.4 Layout

The layout field determines which layout soulb be used when shoing this page.

2.5 Revisions

All ZootPage history is preserved in revisions. These are automatically generated, and pages may be rolled back to any revision.

2.6 Fields

Zoot pages can have custom fields. To configure such fields one creates a list of stings which contains the names of the custom fields in Config.groovy like this: `no.zoot.fields = ["image_url", "customer"]`.

2.7 Automatic slug generation

Slugs can be automatically generated for pages by creating a closeure in Config.groovy tied to `no.zoot.slugGenerator`. A very simple UUID slug may be implemented like this:

```
no.zoot.slugGenerator = {parent ->
  return UUID.randomUUID().toString()
}
```

3 User authentication

Zoot does not have a user authentication system (it is minimalistic after all). However zoot will detect if the "authenticate"? plugin is intalled, and if so authentication will be enabled.

4 Backup of Zoot system

To back up a zoot one can simply append `.xml` to the list action in the ZootPage controller. This can be done wit a simple curl script.

Example curl script without authentication:

```
curl -L -d "http://zootApplication/zootPage/list.xml" >\\
/var/backup/zoot/list.xml
```

Example curl script with authentication:

```
curl -c /tmp/cookies -L -d \\
"success_controller=zootPage&success_action=list.xml&\\
login=zoot&password=zoot&errorController=zootPage&error_action=login"\\
"http://zootApplication/authentication/login" > /var/backup/zoot/list.xml
```