

ProposalSinglePublication

REASONS

General

1. In real world one makes one webapp for each website. In real life you manage one website with a one CMS webapp, not many sites with one CMS. I know Lenya was originally designed with large multi-publication university site-farms in mind, but that seems not to be a common case.
2. Because of 1. you probably don't need multi-publication features, do you?

Urls

3. Stripping URLs, mod_rewriting, link rewriting, etc - all methods used to cheat browser about real (real in cocoon sense of course) position is not a good idea. It forces publisher to deploy other software (ie front end httpd server). Also, when fall-back mechanism is used - it simply does not work, as fall-back url's are higher in directory structure. (Example: if you publish /default/live/ at / - and invoke unexistent document-url - default Lenya box uses /lenya/resources which are out of /default/live structure.

When not using any front-end software, you are forced to publish site with /publication-id/live/ prefix in every URL.

4. Prefix /publication-id in ULR (ie /default) is unwanted. At least because of a reason 2.
5. What about '/live' area prefix? Even if you succeed with stripping publication-id prefix - who on earth would like to publish his own site at internet address: <http://www.somecompany.com/live/> ?

This is not the flexibility Cocoon/Lenya should give you..

Directory structure

6. Directory structure is also very strange... You've got at least three sets of content / resource / xslt areas:

```
at lenya/                (for fall-back) (which doesn't work when #3)\
at pubs/publication/     ("local")\
at pubs/publication/lenya (for over-write)\
```

7. Having in mind #1 and #2, xslt designers, graphic designers have to go through deep directory structure to get into their files. 'lenya/pubs/publication-id/resources/shared..' is a bit long 😞

GOALS and IDEAS

General

1. Simplify Lenya
2. At least make one publication to be a "null, empty name", or - even make publication-id irrelevant

Urls

1. Make live accessible at /, not at /default/live/
2. Perhaps set some request uri to access "backend", for example /lenya/.
How could it work:

```
http://localhost:8080/webapp/    <---- Publication (default, one and only), Live area

http://localhost:8080/webapp/lenya/    <-- backend entry point
http://localhost:8080/webapp/lenya/admin/    <-- admin menu
http://localhost:8080/webapp/lenya/info-authoring/    <-- info-authoring menu
http://localhost:8080/webapp/lenya/authoring/    <-- authoring area
http://localhost:8080/webapp/lenya/live/    <-- live area (the same as at '/')
```

Directory structure

3. Site files shouldn't be inside Lenya "system files".. What about this way:

```
webapp/legal
webapp/WEB-INF

webapp/content/      <-- publication content directory (*)
webapp/resources/    <-- publication resources directory (*)
webapp/xslt/          <-- publication xsl stylesheets (*)

webapp/lenya/
    bin/              <-- files of current lenya/bin/
    config/           <-- files of current lenya/config/ AND "pubs/default/config/* also
    content/          <-- files of current lenya/content/
    resources/        <-- files of current lenya/resources/
    xslt/              <-- files of current lenya/xslt/
    logs/             <-- current "pubs/default/logs/" dir
    work/             <-- current "pubs/default/work/" dir (scheduler, lucene, etc..)

* with override of and fallback to ../lenya/content/
                                   ../lenya/resources
                                   ../lenya/xslt           respectively
```

Probably achieved:

- clear division between site files and Lenya CMS 'system' files
- easy override and fallback engine:
 - if "*.ext" misses - it could be found at "lenya/*.ext" match
 - if "*.ext" exists - it overrides "lenya/*.ext"
- you don't lose fallback resources when mod_proxying

WDYT?