Methods from Products.CMFPlone.utils (continued) -

getEmptyTitle(context, translated=True)

Returns the string to be used for objects with no title or id.

typesToList(context)

Returns all the types except the ones that are in metaTypesNotToList.

normalizeString(text, context=None, encoding=None)

Takes an input unicode text and returns a normalized version of the text.

_createObjectByType(type_name, container, id, *args, **kw)

Creates an object without performing security checks.

safeToInt(value)

Converts value to integer or just return 0 if we can't.

versionTupleFromString(v_str)

Returns version tuple from passed in version string.

getFSVersionTuple()

Reads version.txt and returns a version tuple.

transaction_note(note)

Write human readable note for the current transaction.

base hasattr(obj, name)

Same as safe_hasattr, but also disables acquisition.

safe_hasattr(obj, name, _marker=object())

Implementation of ${\tt hasattr}$ that uses 3 arguments so an ${\tt AttributeError}$ is not masked.

safe callable(obi)

Makes sure that the callable checks are ConflictError safe.

safe unicode(value, encoding='utf-8')

Converts a value to unicode, even if it is already a unicode string.

tuplize(value)

Changes the value to a tuple.

_unrestricted_rename(container, id, new_id)

Renames a particular sub-object. This is copied from ATCT migrations to avoid a dependency. This is a less strict version of manage_renameObject (no write lock check and no verify object check from PortalFolder so it's allowed to rename even disallowed portal types inside a folder).

© creative commons

This brochure is licensed by Six Feet Up, Inc. under the Creative Commons Attribution-ShareAlike 3.0 License. Find out more at www.sixfeetup.com/quickref. The

Plone name and the Plone logo are registered trademarks of the Plone Foundation. All other trademarks and brand names used herein are acknowledged as the property of their respective owners.

plone.app.layout.globals

The plone.app.layout.globals module has browser views that contain utility methods. Each of the headers below represent the view class name.

Example in python:

```
from zope.component import getMultiAdapter
context_state = getMultiAdapter((context, request),
   name=u'plone_context_state')
page url = context state.current page url()
```

Example in a page template:

```
<div tal:condition="context/@@plone_portal_state/anonymous">
    We do something specific to the anonymous user.
</div>
```

plone interface info

provides

Given an interface dotted name, determine if the context provides this interface.

class provides

Given an interface dotted name, determine if the context's class provides this interface.

plone_tools

actions

portal_actions tool

catalog

portal_catalog tool

membership

portal membership tool

properties

portal properties tool

syndication

portal syndication tool

types

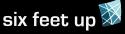
portal_types tool

url

portal url tool

workflow

portal workflow tool





plone_portal_state

portal

The Plone site object.

portal title

Title of the Plone site.

portal url

URL of the Plone site.

navigation root path

Path of the navigation root.

navigation root url

URL of the navigation root.

default_language

Default language for the Plone site.

language

Current language.

locale

Current locale.

is rtl

Whether or not the site is being viewed in a right to left language.

member

Currently authenticated member.

anonymous

Whether or not the current member is anonymous.

plone_context_state

current_page_url

URL to the current page, including template and query string.

current_base_url

Current "actual" URL from the request, excluding the query string.

canonical object

Current object unless this object is the default page in its folder, in which case the folder is returned.

canonical object url

URL of the current object unless this object is the default page in its folder, in which case the folder is returned.

view url

URL to use for viewing. Files and Images get downloaded when they are directly called, instead of with /view appended.

view template id

The id of the view template for the current context.

is_view_template

Returns True if the current URL (in the request) refers to the standard "view" of the context (i.e. the "view" tab).

obiect url

URL of the current object.

object_title

Prettified title of the current object.

workflow state

Workflow state of the current object.

parent

Direct parent of the current object.

folder

Current canonical folder.

is folderish

True if this is a folderish object, structural or not.

is_structural_folder

True if this is a structural folder.

is default page

True if this is the default page.

is_portal_root

True if this is the Plone site or the default page in the site.

is editable

Whether or not the current object is editable.

is locked

Whether or not the current object is locked.

actions

Filtered actions for the current context.

portlet_assignable

Whether or not the context is capable of having locally assigned portlets.

Methods from Products.CMFPlone.utils

The CMFPlone product has a utils module with some helpful utility code as well.

Here is an example of using the _createObjectByType method:

```
from Products.CMFPlone import utils
utils._createObjectByType('Document', portal, 'new-doc',
   title="New Doc", subject=['foo', 'bar'])
```

Methods

isIDAutoGenerated(context, id)

Returns True if the id has been auto generated using the portal type name and the date of creation.

pretty_title_or_id(context, obj, empty_value)

Returns the best possible title or id of an item (catalog brain or object). Returns an empty title marker if the id is not set.

getSiteEncoding(context)

Returns the site encoding if the site_properties exist, $\mathtt{utf-8}$ otherwise.



- Generic Setup
- Buildout
- Plone Theming