Plone

Interfaces

Sample

```
from zope.interface import Interface, Attribute
class IExample(Interface):
    """ Interface description """
    some_attribute = Attribute("Some value")

    def some_method(some_arg='default'):
        """ Method description """
```

Implementation

```
from zope.interface import implements
class Example(object):
    """ Description """
    implements(IExample)

    some_attribute = 1

    def some_method(self, some_arg='default'):
        return some_arg
```

Checking

```
>>> IExample.implementedBy(Example)
... True
>>> obj = Example()
>>> IExample.providedBy(obj)
... True
```

Adding

```
>>> from zope.interface import alsoProvides
>>> ISomeAnotherInterface.providedBy(obj)
... False
>>> alsoProvides(obj, ISomeAnotherInterface)
>>> ISomeAnotherInterface.providedBy(obj)
... True
```

Catalog searching

```
result = context.portal_catalog(object_provides=
        IExample.__identifier__)
```

Events

Class

```
class ISomeEvent(Interface):
    """ some event """

class SomeEvent(object):
    implements(ISomeEvent)
```

Subscriber

```
@adapter(ISomeEvent)
def react(some_event_object):
    """ subscriber description """
    ...
```

Subscriber's zcml

```
<subscriber handler=".events.react"
            for=".interfaces.ISomeContentType
                 .interfaces.ISomeEvent"/>
```

Notification

```
from zope.event import notify
notify(SomeEvent())
```

Global variables

Context

current_page_url	string, current **page** url
here_url	string, current **object** url
hidecolumns	string, portlets columns visibility
isContextDefaultPage	bool, is current object default page?
is_editable	bool, has member permission 'Modify portal content' in context?
isFolderish	bool, context isPrincipiaFolderish attribute
isLocked	bool, is current page locked by another user?
isStructuralFolder	bool, does context is_folderish and also doesn't provides INonStructuralFolder?
isViewTemplate	bool, is current page "view" or not ("edit", "sharing" etc.)?
navigation_root_url	string, navigation root url
object_title	string, current object title
sl, sr	bool, true if has any left/right portlet
template_id	string, current template id
wf_state	string, workflow state

Adapters

Adapter's class

```
class SomeAdapter(object):
    implements(IProvided)
    adapts(IAdapted, IOtherAdapted)

    def __init__(self, context):
        self.context = context

    def provided_method(self):
        """ """
    ...
```

Adapter's zcml

```
<adapter factory=".adapters.SomeAdapter"
         provides=".interfaces.IProvided"
         for=".interfaces.IAdapted
              .interfaces.IOtherAdapted"
         name="some" />
```

Getting adapter

```
>>> adapted = Adapted()
>>> some = IProvided(adapted)
```

Getting multi adapter

```
from zope.component import getMultiAdapter
adapted, other = Adapted(), OtherAdapted()
getMultiAdapter((adapted, other), IProvided,
         name="some")
```

Factory

```
from zope.component import adapter
from zope.interface import implementer

class SomeAdapter(object):
    implements(IProvided)
    ...

@implementer(IProvided)
@adapter(IAdapted)
def some_adapter(context):
    return SomeAdapter(context)
```

Utilities

Global

```
class ISomeUtility(Interface):
    """ some utility """

class SomeUtility(object):
    implements(ISomeUtility)

    def __init__(self, arg):
        self.arg = arg

some1 = SomeUtility(1)
some2 = SomeUtility(2)
```

Unnamed utility zcml

```
<utility factory=".some.SomeUtility"
         provides=".interfaces.ISomeUtility"/>
```

Named utility zcml

```
<utility component=".some.some1"
         provides=".interfaces.ISomeUtility"
         name="some1"/>
```

Getting utility

```
from zope.component import getUtility
some = getUtility(ISomeUtility, name="some")

# or
from Products.CMFCore.utils import
        getUtilityByInterfaceName
getUtilityByInterfaceName('Products.CMFCore.
        interfaces.ISiteRoot')
```

Local utility profile
(componentregistry.xml)

```
<componentregistry>
 <utilities>
  <utility interface="path.to.interface"
           object="/obj_name_in_zodb"/>

  <!-- or -->

  <utility interface="path.to.interface"
           factory="path.to.utility.class"/>
 </utilities>
</componentregistry>
```

Portal

default_language	string, default language
language	string, language
isRTL	bool, is right to left?
portal	Plone Site, portal root
portal_tabs	list, top level tabs
portal_title	string, portal title
portal_url	string, portal url
site_properties	site_properties sheet of PropertiesTool
visible_ids	1 if user can edit ids, None otherwise

User

checkPermission	method of MembershipTool
isAnon	bool, is user anonymous?
member	authenticated member
membersfolder	members folder object

Tools

acl_users	PAS, acl_users
atool	ActionsTool, portal_actions
ifacetool	InterfaceTool, portal_interface
mtool	MembershipTool, portal_membership
portal_properties	PropertiesTool, portal_properties
putils	PloneTool, portal_utils
syntool	SyndicationTool, portal_syndication
utool	URLTool, portal_url
wtool	WorkflowTool, portal_workflow

Utils

normalizeString(text, relaxed=False)	method, normalizes a title to an id
toLocalizedTime(time, long_format=None)	method, convert time to localized time
uniqueItemIndex	RealIndexIterator, iterator used for tabindex attribute
ztu	module ZTUtils (Batch, TreeMaker, LazyFilter etc)
Iterator	IndexIterator, deprecated
tabindex	IndexIterator, start at 30000, deprecated

Actions

actions	dict, all actions
folder_actions	list, actions with category *folder*
global_actions	list, actions with category *global*
keyed_actions	dict of dicts with keys of categories and action ids
user_actions	list, actions with category *user*
workflow_actions	list, actions with category *workflow*

Global views

Available as eg. *portal_state = getMultiAdapter((context, request), name=u"plone_portal_state")* in code, or *tal:define="portal_state context/@@plone_portal_state"* in template.

plone

cropText(text, length, ellipsis='...') crop text on a word boundary, displayContentsTab() bool, show content tabs?, getCurrentFolder(), getCurrentFolderUrl(), getCurrentObjectUrl(), getCurrentUrl(), getIcon(item) return ContentIcon object for item, getParentObject() parent for context, getViewTemplateId(), have_portlets(manager_name, view=None), icons_visible() bool, are item's icons visible?, isDefaultPageInFolder(), isFolderOrFolderDefaultPage(), isPortalOrPortalDefaultPage(), isStructuralFolder(), navigationRootPath(), navigationRootUrl(), showEditableBorder() bool, show border?, toLocalizedTime(time, long_format=None), visibleIdsEnabled() check if user can edit object id,

plone_portal_state

anonymous(), default_language(), friendly_types() get types list except site_properties/types_not_searched, is_rtl(), language(), locale(), member(), navigation_root_path(), navigation_root_url(), portal(), portal_title(), portal_url()

plone_tools

actions(), catalog(), interface(), membership(), properties(), syndication(), types(), url(), workflow()

plone_context_state

actions() actions list except portal_workflow, folder_buttons and object_buttons, canonical_object() get 'self.is_default_page() and self.parent() or self', canonical_object_url(), current_base_url() get request/ACTUAL_URL or

Various utils*... from some utils.py files*

Products.CMFCore.utils

getToolByName(obj, name, default=_marker)	get tool object by it's name, eg. getToolByName(self, 'portal_types')
getUtilityByInterfaceName(dotted_name, default=_marker)	get tool/utility object by it's full dotted interface name
cookString(text)	make a Zope-friendly ID from *text* (remove spaces and lowercase)
_getAuthenticatedUser()	shortcut to getSecurityManager().getUser()
_checkPermission(permission, obj)	shortcut to getSecurityManager().checkPermission(permission, obj)
_ac_inherited_permissions(ob, all=0)	get all permissions not defined in ourself that are inherited
_modifyPermissionMappings(ob, map)	modifies multiple role to permission mappings
... and others	

Products.CMFDefault.utils

formatRFC822Headers(headers)	convert the key-value pairs in *headers* to valid RFC822-style headers
parseHeadersBody(body, headers=None, rc=re.compile(r'\n|\r\n'))	parse any leading 'RFC-822'-ish headers from an uploaded document, returning a dictionary containing the headers and the stripped body
seq_strip(seq, stripper=lambda x: x.strip())	strip a sequence of strings
scrubHTML(html)	strip illegal HTML tags from string text
isHTMLSafe(html)	would current HTML be permitted to be saved?
translate(message, context)	translate i18n message
getBrowserCharset(request)	get charset preferred by the browser
makeEmail(mtext, context, headers={})	make email message
checkEmailAddress(address)	check email address
... and others	

Products.CMFPlone.utils

parent(obj)	shortcut to aq_parent(aq_inner(obj))
createBreadCrumbs(context, request)	return bredcrumbs
createNavTree(context, request, sitemap=False)	return navtree
createSiteMap(context, request, sitemap=False)	return sitemap
isDefaultPage(obj, request)	is obj default page?
getDefaultPage(obj, request)	get obj default page
pretty_title_or_id(context, obj)	return the best possible title or id of an item
getSiteEncoding(context)	return site encoding
portal_utf8(context, str, errors='strict')	encode str from portal encoding to utf-8...
utf8_portal(context, str, errors='strict')	... and reverse
typesToList(context)	list of types except navtree_properties.metaTypesNotToList
normalizeString(text, context=None, encoding=None)	normalize string
_createObjectByType(type_name, container, id, *args, **kw)	create an object without performing security checks
safeToInt(value)	convert value to integer or return 0
transaction_note(note)	write human legible note for transaction
safe_hasattr(obj, name)	like hasattr but not mask exceptions
base_hasattr(obj, name)	like safe_hasattr but disable acquisition
safe_callable(obj)	callable safe for ConflictError
safe_unicode(value, encoding='utf-8')	converts a value to unicode, even it is already a unicode
directlyProvides(obj, *interfaces)	like zope.interface.directlyProvides but safe for Zope2 interfaces
classImplements(class_, *interfaces)	like zope.interface.classImplements but safe for Zope2 interfaces
classDoesNotImplement(class_, *interfaces)	like zope.interface.classImplementsOnly but safe for Zope2 interfaces
scale_image(image_file, max_size=None, default_format=None)	scales an image down
... and others	

request/VIRTUAL_URL or request/URI or context.absolute_url, current_page_url() get page url with query string, folder(), is_default_page(), is_editable(), is_folderish(), is_locked(), is_portal_root(), is_structural_folder(), is_view_template(), keyed_actions(), object_title(), object_url(), parent(), portlet_assignable() bool, is ILocalPortletAssignable provided by context, view_template_id(), view_url() object_url with added "/view" for files and images, workflow_state()

plone_interface_info

class_provides(dotted_name) bool, is dotted_name interface provided by aq_base(context).__class__ ?, provides(dotted_name) bool, is dotted_name interface provided by aq_base(context)?

TAL and METAL

TAL (xmlns:tal="http://xml.zope.org/namespaces/tal) *... in order of execution*

<table>
<tr><td>define</td><td>define variable, eg.
<code>&lt;div tal:define="global title here/title_or_id"&gt;</code></td></tr>
<tr><td>condition</td><td>test condition, eg. <code>&lt;div tal:condition="not: here/title"&gt;</code></td></tr>
<tr><td>repeat</td><td>loop iterable, eg. <code>&lt;div tal:repeat="row search_results"&gt;</code>. Extra variables available in loop:
<table>
<tr><td>index</td><td>int, iteration number, starting from 0</td></tr>
<tr><td>number</td><td>method returning int, iteration number, starting from 1</td></tr>
<tr><td>even / odd</td><td>method returning bool, true for even / odd rows</td></tr>
<tr><td>start / end</td><td>bool, true for first / last row</td></tr>
<tr><td>length</td><td>method returning int, number of iterations</td></tr>
<tr><td>letter / Letter</td><td>method returning str, iteration number as lower / uppercase letter (a, b, c… z, aa, ab…)</td></tr>
<tr><td>roman</td><td>method returning str, iteration number as lowercase roman number (i, ii, iii, iv…)</td></tr>
<tr><td>first / last</td><td>method returning bool, true if value passed as argument occures first / last time in this iteration</td></tr>
</table></td></tr>
<tr><td>content</td><td>replace tag content, eg. <code>&lt;div tal:content="structure here/title"&gt;</code></td></tr>
<tr><td>replace</td><td>replace tag content and remove tag, eg.
<code>&lt;div tal:replace="structure here/title"&gt;</code></td></tr>
<tr><td>attributes</td><td>set tag attribute, eg.
<code>&lt;div tal:attributes="href here/absolute_url"&gt;</code></td></tr>
<tr><td>omit-tag</td><td>remove tag but keep content, eg.
<code>&lt;div tal:omit-tag="optional_condition"&gt;</code>; if optional condition evaluates to false tag won't be removed</td></tr>
<tr><td>on-error</td><td>hendle error caused by any other tag, eg.
<code>&lt;div tal:on-error="optional_expression"&gt;</code>; if optional expression is provided it will be evaluated in case of errors</td></tr>
</table>

METAL (xmlns:metal="http://xml.zope.org/namespaces/metal")

define:macro	mark part of template to be referencable later, eg. `<div metal:define-macro="box">...</div>`
use:macro	use macro defined in another template, eg. `<div metal:use-macro="/path/to/template/macros/box" />`
define:slot	part of template which should be filled later, eg. `<div metal:define-slot="main" />`; must be inside define-macro
fill:slot	fill slot defined in another template, eg. `<div metal:fill-slot="main">...</div>`; must be inside use-macro

Installation

`$ wget http://peak.telecommunity.com/dist/ez_setup.py`		get ez_setup.py
`$ python ez_setup.py`		install easy_install
`$ easy_install ZopeSkel`		install ZopeSkel
`$ paster create --list-templates`		list of installed templates
`$ paster create -t plone3_buildout test`		create *test* project
`$ cd test && python bootstrap.py`		bootstrap project

`$ ./bin/buildout`	`[-o offline]`
	`[-N non newest]`
		build project

`$ ./bin/instance start`		start instance

Roles

Plone	
Member	each logged in user
Reader	can read
Editor	can edit
Contributor	can add
Reviewer	can review
Manager	site admin
Zope	
Anonymous	not logged in
Authenticated	logged in (this role can't be granted or revoked)
Owner	owner, usually creator of object

use checkPermission to check rights, eg.

```
>>> from AccessControl import getSecurityManager
>>> getSecurityManager.checkPermission('ModifyPortalContent', context)
```

Chosen events

Contents

```
Products.Archetypes.interfaces.IEditBegunEvent
Products.Archetypes.interfaces.IEditCancelledEvent
Products.Archetypes.interfaces.IObjectInitializedEvent
Products.Archetypes.interfaces.IObjectEditedEvent
```

copying, moving and removing objects

```
zope.lifecycleevent.interfaces.IObjectCopiedEvent
zope.app.container.interfaces.IObjectMovedEvent
OFS.interfaces.IObjectClonedEvent
OFS.interfaces.IObjectWillBeRemovedEvent
zope.app.container.interfaces.IObjectRemovedEvent
zope.app.container.interfaces.IContainerModifiedEvent
```

other events triggered many times by portal_factory during object creation

```
zope.lifecycleevent.interfaces.IObjectCreatedEvent
OFS.interfaces.IObjectWillBeAddedEvent
zope.app.container.interfaces.IObjectAddedEvent
zope.app.container.interfaces.IContainerModifiedEvent
zope.lifecycleevent.interfaces.IObjectModifiedEvent
OFS.interfaces.IObjectWillBeMovedEvent
zope.app.container.interfaces.IObjectMovedEvent
```

Workflow

```
Products.CMFCore.interfaces.IActionWillBeInvokedEvent
Products.DCWorkflow.interfaces.IBeforeTransitionEvent
Products.DCWorkflow.interfaces.IAfterTransitionEvent
Products.CMFCore.interfaces.IActionRaisedExceptionEvent
Products.CMFCore.interfaces.IActionSucceededEvent
```

Login

```
Products.PlonePAS.interfaces.events.IUserInitialLoginInEvent
Products.PluggableAuthService.interfaces.events.IUserLoggedInEvent
Products.PluggableAuthService.interfaces.events.IUserLoggedOutEvent
```

Working copies (iterate)

```
plone.app.iterate.interfaces.IBeforeCheckoutEvent
plone.app.iterate.interfaces.ICheckoutEvent
plone.app.iterate.interfaces.ICancelCheckoutEvent
plone.app.iterate.interfaces.ICheckinEvent
plone.app.iterate.interfaces.IWorkingCopyDeletedEvent
plone.app.iterate.interfaces.IAfterCheckinEvent
```

Other

site configuration change

```
plone.app.controlpanel.interfaces.IConfigurationChangedEvent
```

by Piotr Furman, 2009 | Plone® and the Plone logo are registered trademarks of the Plone Foundation |