

Building Trac Plugins

Noah Kantrowitz
PyCon 2008
Saturday, March 15

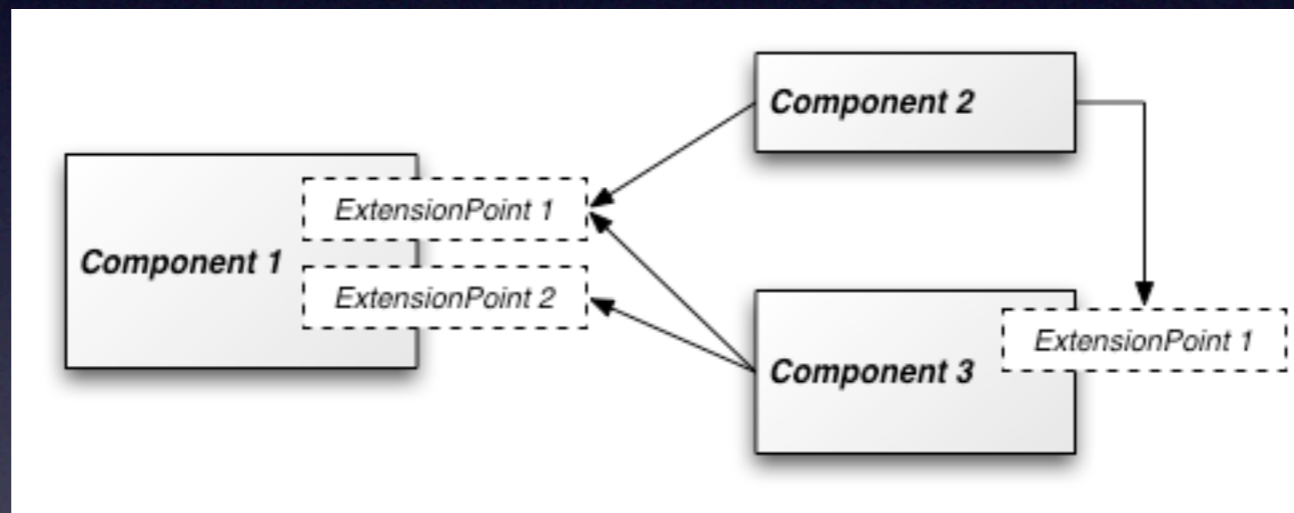
Trac

- Wiki
- Tickets
- Source Browser
- ...?

Trac Plugins

- TracBlog
- Tags
- MasterTickets
- Vote
- AccountManager
- and lots more. Check out trac-hacks.org

Basic Architecture



trac.core

- Component - Subclass this
- implements(IFoo) - I am a Foo
- ExtensionPoint(IFoo) - I want all Foo

A Trivial Example

```
class ISpam(Interface):  
    def eat(req): pass
```

```
class IEgg(Interface):  
    def scramble(): pass
```

```
class ExampleModule(Component):  
    implements(ISpam)  
    eggs = ExtensionPoint(IEgg)
```

```
def __init__(self):  
    for egg in self.eggs: print egg.scramble()
```

IRequestHandler

- `from trac.web.api import IRequestHandler`
- `def match_request(self, req):`
 `return req.path_info.startswith('/foo')`
- `def process_request(self, req):`
 `data = {}`
 `# The guts of the plugin`
 `return 'foo.html', data, mime_type`

ITemplateProvider

- `from trac.web.chrome import ITemplateProvider`
- `from pkg_resources import resource_filename`
- `def get_templates_dirs(self):`
 `yield resource_filename(__name__, 'templates')`
- `def get_htdocs_dirs(self):`
 `yield 'foo', resource_filename(__name__, 'htdocs')`

File layout

- `setup.py`
- `foo/`
 - `web_ui.py`
 - `templates/`
 - `foo.html`
 - `htdocs/`
 - `foo.css`

Magic Variables

- `self.env` - Environment object
- `self.log` - a PSL object
 - `self.log.debug('Message %s', variable)`
- `self.config` - You probably don't need this ...

Config Descriptors

- `trac.config`
- `Option`, `BoolOption`, `IntOption`, `ListOption`
- `class Foo(Component):`
 - `stuff = Option('section', 'key', default='a', doc='An example')`
- `if self.stuff == 'b': ...`

Genshi in 1 second

```
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:py="http://genshi.edgewall.org/"
  xmlns:xi="http://www.w3.org/2001/XInclude">
  <xi:include href="layout.html" />
  <head>
    <title>Foo Plugin</title>
  </head>
  <body>
    <h1>Foo</h1>
    Hello world.
  </body>
</html>
```

setup.py

```
entry_points = {  
    'trac.plugins': [  
        'foo.web_ui = foo.web_ui',  
    ],  
}
```

Navigation bar

- `from trac.web.chrome import INavigationContributor`
- `def get_navigation_items(self, req):`
 `yield 'mainnav', 'foo', tag.a('Foo', href=req.href.foo())`
- `def get_active_navigation_item(self, req):`
 `return 'foo'`

Href Objects

- `req.href() == '/trac'`
- `req.href.foo('bar', 'baz') == '/trac/foo/bar/baz'`
- `req.href.foo(bar='baz') == '/trac/foo?bar=baz'`
- `req.abs_href() == 'http://example.com/trac'`

ITemplateProvider (redux)

- `from trac.web.chrome import ITemplateProvider`
- `from pkg_resources import resource_filename`
- `def get_templates_dirs(self):`
 `yield resource_filename(__name__, 'templates')`
- `def get_htdocs_dirs(self):`
 `yield 'foo', resource_filename(__name__, 'htdocs')`

File layout

- `setup.py`
- `foo/`
 - `web_ui.py`
 - `templates/`
 - `foo.html`
 - `htdocs/`
 - `foo.css`

Using Static Files

- `from trac.web.chrome ...`
- `req.href.chrome('foo', 'foo.css')`
- `add_stylesheet(req, 'foo/foo.css')`
- `add_script(req, 'foo/foo.js')`

Data Models

- `trac.wiki.model.WikiPage`
- `trac.ticket.model.Ticket`
- `trac.versioncontrol.api.Repository`
- `trac.ticket.query.Query`

WikiPage

- `page = WikiPage(self.env, 'MyPageName')`
- `assert page.exists`
- `page.text += 'added text'`
- `page.save('coderanger', 'I changed it', '127.0.0.1')`

Ticket

- `tk = Ticket(self.env, 1)`
- `print tk['summary']`
- `tk['version'] = '2.0'`
- `tk.save_changes('coderanger', 'Bump version')`

Data II: The Sequel

- `db = self.env.get_db_cnx()`
- DB API 2 (PEP 249)
- `%s` for paramstyle
- All `unicode()`

Permissions

- 'WIKI_VIEW' in req.perm
- req.perm.require('WIKI_VIEW')
- 'WIKI_VIEW' in req.perm('wiki', 'PageName')

IPermissionRequestor

- `from trac.perm import IPermissionRequestor`
- `def get_permission_actions(self):`
 - `yield 'FOO_VIEW'`
 - `yield 'FOO_ADMIN', ['FOO_VIEW']`

Resource Descriptors

- realm, id, version
- parent
- `trac.resource.Resource`
- Most things will take a tuple

jQuery

- Now integrated into Trac core.
- See jquery.com

Request Filters

- `from trac.web.api import IRequestFilter`
- `def pre_process_request(self, req, handler):`
 `return handler`
- `def post_process_request(self, req, template, data, mime_type):`
 `return template, data, mime_type`

Stream Filters

- `from trac.web.api import ITemplateStreamFilter`
- `from genshi.filter.transform import Transformer`
- ```
def filter_stream(self, req, method, filename, stream, data):
 stream |= Transformer('div[id="main"]//img')\
 .attr('href', req.href.chrome('site', 'omgpony.png'))
 return stream
```

# Want more?

- #trac on freenode (I am coderanger)
- Dinner with the developers, 5PM
- BoF tonight, 7PM in LaGuardia.
- Sprint!

Questions?