

# Azax

## Bringing DOM to the server

Plone Conference

Vienna 2005

Godefroid Chapelle - BubbleNet

# Introduction

## Framework

- Real code

started with Tarek Ziadé in Gotenborg

- Still embryonic

→ foster interest

→ create a community

Let's dream of a big success like Five :-)

# Introduction - Ajax

- Buzzword ;-)
- New name - Old technique
  - Zope 3 dynamic tree built with Paul Everitt in december 2002
- Integration of techniques
  - HTML
  - Javascript
  - CSS
  - XMLHttpRequest

# Outline

- Why ?  
Reasons for the architecture
- What ?  
Main parts of the framework
- Demo  
CompositePack
- How ?  
Some design and some code

# Why ?

- More and more dynamic UI  
→ almost a demand
  - Gmail
  - Google maps
  - Ruby on Rails
- Hard → Expensive
  - JS is other language
  - Cross-browser
- We need a framework

# Why another framework ?

- Cost of development
  - The developer
  - The maintenance
- Savvy developers prefer to learn and reuse previously debugged software
  - Presentation in the browser
  - Security in Zope
  - Schemas in Archetypes
- Learning is ok but not too much

# Why a framework on the server ?

- Learn JS when needed

cfr CSS

- Keep the model on the server to avoid synchronization
- Integration in the current development model

# What ?

- Behavioral stylesheets
  - Ajax independent of HTML structure
  - Application accessibility
    - i18n included
- Generic client JS engine
  - Knows nothing of the model
    - As HTML or CSS
- DOM on the server
  - Use server-side techniques

# What ?

## Behavioral stylesheets

- Loaded with the page
- Setup calls
  - To the server
  - (To JS library)
- XML payload
  - could be css like syntax

# What ?

## Client JS engine

- Process commands
- Updates the DOM
- Written once
  - Debug once, really ;-) ?
  - Reuse

# What ?

## DOM on the server

- Python API
- Build updates as a set of atomic commands
- XML payload
  - With interleaved HTML
  - Human readable output

# Demo – CompositePack

- Tool to add composability on any archetypes content
- Design view
  - Drag and drop reordering
  - Kupu drawers to select content
- 1.0 RC 1

# How ?

- JS processor of stylesheets
- AjaxResponse on the server
- JS processor of commands

# How ?

## JS Stylesheet processor

- Fetch the stylesheet according to the `<link>` tag in the `<head>`
- Parsing of the stylesheet
- Setup of asynchronous calls in events on nodes

# Code : Stylesheet

```
<styles>
  <style>
    <selector>button#copyFrom</selector>
    <event>click</event>
    <action>copyFromDivContent</action>
  </style>
</styles>
```

# How ? Server API

- AzaxResponse class
  - Method calls representing client commands
  - Builds the XML payload
- Based on Ixml
- Five infrastructure

# Code : Server API

```
response = AjaxResponse
    (self.request.response)

# produce html excerpt

added_html = destination.getEditingViewlet
    (new_el)

selector = '#%s' % target_id

# add a command

response.addAfter(selector, added_text)

# get XML serialization

return response()
```

# How ?

## Commands processor

- Asynchronous calls
- Parsing of AjaxResponse payload
- Select nodes target with CSSQuery
- Executes DOM updates

# Code : Commands Payload

```
<selectors>
  <selector>
    <value>#first_cp_top</value>
    <commands>
      <command>
        <name>addAfter</name>
        <data name="html">
          <div xmlns="http://www.w3.org/1999/xhtml">
            some html
          </div>
        </data>
      </command>
    </commands>
  </selector>
</selectors>
```

# References

- **Azax repository**

<https://svn.nuxeo.org/pub/z3lab/azax/trunk>

- **CompositePack repository**

<https://svn.plone.org/svn/collective/CompositePack/trunk>

# Bern Ajax Sprint

- November Thu 17, Fri 18, Sat 19 2005
- Hosted by 4teamwork

# Questions

Thanks !!