reahl™

the Python-only web framework

Iwan Vosloo
About

15 Python

Financial industry / in-house development

26,600 loc
(40% tests)
the Python-only web framework

- Background
- What it is
- How it works
- Why you should care
- Strategy
- Status
- Questions
Background

Web frameworks

1. GET /some/url
2. map url to some code
3. read from a database
4. merge data with template

HTML Template

<h1>HTML</h1>

```python
def func(code code co):
    code code code
code code
```

<body>
</body>
</html>

respond final HTML
Background

Web frameworks

- re-use templates
- css frameworks
- js libraries
Background

Technology focus

- Javascript (js libraries)
- CSS (other frontend libraries)
- HTML (template language)
- HTTP (routing)
What it is

Goal focus

**Address Book**
John: john@example.org

**View:** /

**View:** /add

Name:
Email:

**Button**
Add Address

Save

**how**

Javascript (js libraries)
CSS (other frontend libraries)
HTML (template language)
HTTP (routing)

EuroPython 2015
What it is

Pages are widgets
What it is

Composing widgets

```python
panel1 = Div(view)
panel1.add_child(P(view, text='a paragraph with text'))
```

EuroPython 2015
What it is

Widgets are not HTML
What it is
Widgets are not HTML

sliding_panel = SlidingPanel(view)

panel1 = sliding_panel.add_panel(Div(view))
panel1.add_child(P(view, text='a paragraph with text'))

panel2 = sliding_panel.add_panel(Div(view))
panel2.add_child(P(view, text='a different paragraph'))
What it is

Widgets are not HTML
How it works

Behind the SlidingPanel

GET /some/url?index=2

HTML
(showing second panel)
How it works

Behind the SlidingPanel

HTTP (routing)
HTML (template language)
Javascript (js libraries)
CSS (other frontend libraries)
Extra URLs & Query string arguments
Server-side logic

JavasScript (js libraries)
CSS (other frontend libraries)
HTML (template language)
HTTP (routing)
How it works

Widget lifetime

Address Book
John: john@example.org

Add Address

AddressBookPanel
How it works

Widget lifetime

class AddressBookPanel(Div):
    def __init__(self, view):
        super(AddressBookPanel, self).__init__(view)

        self.add_child(H(view, 1, text='Address Book'))

        for address in Session.query(Address).all():
            self.add_child(AddressBox(view, address))
How it works

Defining a User Interface

Address Book
John: john@example.org

Add Address

View: /

Name:

Email:

Save

View: /add

EuroPython 2015
How it works
Defining a UserInterface

class AddressBookUI(\texttt{UserInterface}):

    def assemble(self):

        addresses = self.\texttt{define_view}('\\/', \texttt{title}='Addresses')
        addresses.\texttt{set_page}(\texttt{HomePage}.\texttt{factory}())

        add = self.\texttt{define_view}('\\/add', \texttt{title}='Add an address')
        add.\texttt{set_page}(\texttt{AddPage}.\texttt{factory}())

        self.\texttt{define_transition}(\texttt{Address}.events.save, add, addresses)
Why you should care
Why you should care

Knowledge & focus

Widget

Button

Event

Action

Field

Layout

CheckboxInput

TabbedPanel

ValidationConstraint

Menu

forget technological details
Why you should care
Re-use

Widget

Button

Event

Menu

TextInput

Layout

ValidationConstraint

concurrency
double submit
security issues
quirks

access control

i18n

graceful degradation

accessability

forget technological details
Why you should care

Complicated requirements

<table>
<thead>
<tr>
<th>Name</th>
<th>Twitter</th>
<th>No</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Adam Byrtek</td>
<td>@adamb...</td>
<td>1</td>
<td>Edit</td>
</tr>
<tr>
<td>Adam Forsyth</td>
<td></td>
<td>2</td>
<td>Edit</td>
</tr>
<tr>
<td>Adriano Petrich</td>
<td>@fractal</td>
<td>3</td>
<td>Edit</td>
</tr>
</tbody>
</table>
Strategy

“No-html” frameworks

main stream

reahl™

no-html
Strategy

Maintain web semantics

- Tabbed browsing
- Bookmarks
- Back button
- ...
Strategy

Support web ideals

- JavaScript optional
- Device independence
- Accessibility
- ...

EuroPython 2015
Strategy

Web-grown API
Strategy

Aim: higher level issues

Address Book
John: john@example.org

Add Address
Strategy

Leverage existing wisdom
Status

Our journey

Idealistic

Impossible

Foundation

Risk

Concrete

More flexibility

Bells & whistles

Interest & community

EuroPython 2015
Dream
You’re invited

1. Look at examples
2. Join mailing list(s)
3. Install
4. Follow tutorial
5. Help

spread the word
infra-structure
evolve core
grow up
Thanks

www.reahl.org

On groups.google.com:
- reahl-announce
- reahl-discuss

@reahl

iwan@reahl.org