

YAFOWIL

(Yet Another FOrm Widget Library)

Webformulare ohne Kopfschmerzen

Jens W. Klein <jk@kleinundpartner.at>

23.02.2012

Der Leidensdruck

- „I feel like a stupid beginner while doing a simple z3cform“
- Forms werden doppelt verarbeitet: Validierung und Wertumwandlung
- Ich will keine Interfaces und Klassen schreiben müssen um Formulare zu bauen.
- Ich will Formulare zur Laufzeit verändern können.
- Ich will Formulare in Pyramid und Plone gleich erstellen können.

Designkriterien

- Form-Widgets sind eine Datenstruktur
- Formulare werden deklarativ erstellt.
- Formulare sind nicht zwingend Python Code
- Validierung und Wertumwandlung sind Eins
- Widgets sollen flexibel zusammengebaut werden und einfach funktional erweitert werden können
- Man muss kein Python Guru sein um Forms erstellen zu können.

```
from yafowil import factory

form = factory('form', name='helloworld',
               props={'action': url})

form['person'] = factory('#stringfield',
                         props={'label': 'Who?',
                                'value': '',
                                'required': True})

form['submit'] = factory('field:submit',
                        props={'label': 'greet',
                               'action': 'save',
                               'handler': store,
                               'next': next})
```

```
factory: form
name: soundcloudsettings
props:
  action: expr:context.action
  class: edit-form enableUnloadProtection
widgets:
- client_id:
  factory: field:label:error:text
  value: expr:context.settings.client_id
  props:
    label: Client Id
    required: Client Id needed.
```

- client_secret:
 - factory: field:label:error:text
 - value:
 - expr:context.settings.client_secret
 - props:
 - label: Client Secret
 - required: Client Secret needed.
- save:
 - factory: submit
 - props:
 - action: save
 - expression: True
 - handler: context.save
 - next: context.next
 - label: Save

```
import yafowil.loader

def store(widget, data):
    with open('storage.txt', 'a') as f:
        f.write(data['person'])

def next(widget, data):
    return 'http://some.where.org/nextpage'
```

```
from yafowil import Controller  
  
controller = Controller(form, request)  
  
if not controller.next:  
    return redirect(controller.next)  
  
return controller.rendered
```


Prinzipien

- Runtime rules, static is subordinate,
- Don't mess with a framework,
- Keep it simple and pythonic,
- No fights with storage,
- Use chains and trees as structures.

- **Software-Paket**
<http://pypi.python.org/pypi/yafowil>
- **Dokumentation**
<http://yafowil.info>
- **Beispiel: Plone Custom Search**
<http://plone.org/documentation/kb/build-a-custom-search-form-with-yafowil>