

django

The Web framework for perfectionists with deadlines.

Arne Brodowski
PHP Unconference 2009

Themen

- Was ist Django?
- Request und Response
- ORM (Models)
- Forms
- Templates
- Projekt-Struktur

Was ist Django?

- MVC Framework für Webapplikationen
- Python
- Lawrence Journal-World
- BSD Lizenz (seit Juli 2005)
- Django Software Foundation (seit Juni 2008)

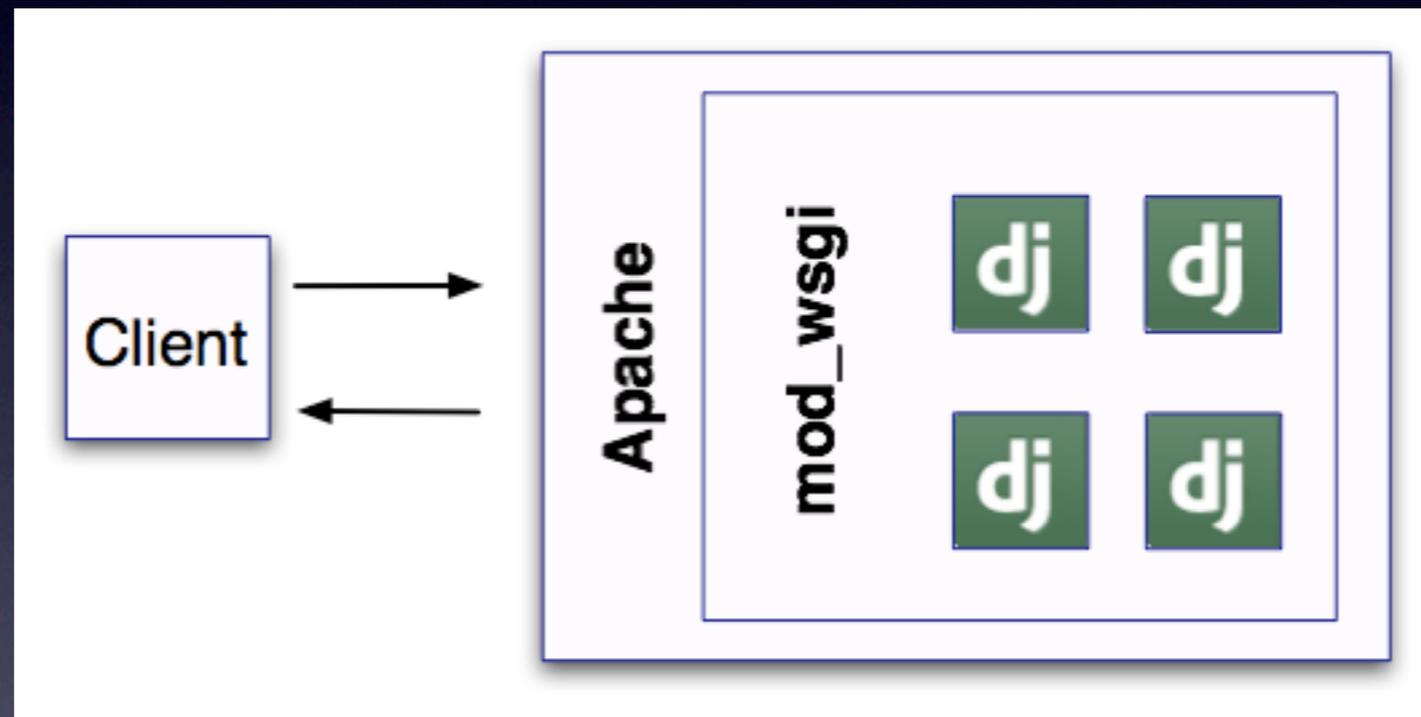
Kern-Komponenten

- ORM
- URL-Dispatcher
- Template-System
- Forms
- i18n, L10n
- Caching
- Middlewares
- Signal-Dispatcher
- Serializer (Json, XML)
- Dev-Server
- Template-Helpers
- Test-Framework

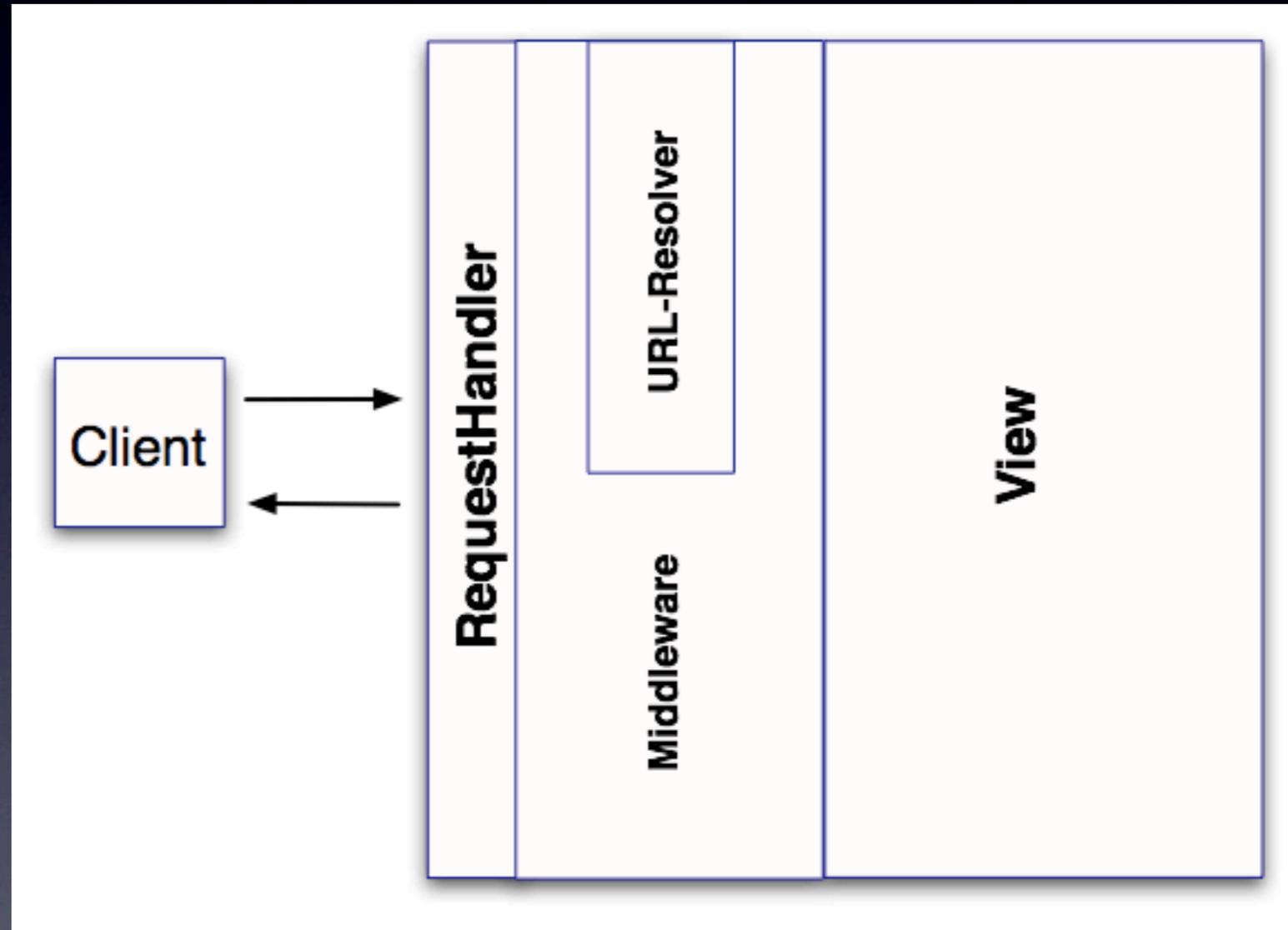
Included Batteries

- Sessions
- Authentication
- Feeds (RSS, Atom)
- Comments
- Sitemaps
- Markup (Textile, ReST, Markdown)
- GIS
- Admin (CRUD)
- Databrowse
- CSRF Protection
- Flatpages
- ContentType

Request - Response



Request - Response



URL-Resolver

```
from django.conf.urls.defaults import *

urlpatterns = patterns('django.views.generic',
    (r'^foo/', include('foo.urls')),
    (r'^foo/$', 'simple.direct_to_template', {'template': 'foo.html'}),

    url(r'^(?P<slug>[\w\-\.\,+\!]+)\.html$', 'list_detail.object_detail',
        {'queryset': BlogEntry.objects.all(), 'slug_field': 'slug',
         'template_name': 'blog/detail.html'},
        name="blog_detail"),

    url(r'^archives/(?P<year>[\d]{4})/(?P<month>[\d]{2}).html$',
        'date_based.archive_month', {...},
        name="blog_archive_month"),
)
```

Views

- Callables (Funktionen, Objekt-Methoden)
- Input: Request - Output: Response
- Input: Parameter aus der URL
- Generic-Views
- Decorators

Views

```
from django.shortcuts import render_to_response, get_object_or_404
from django.contrib.auth.decorators import login_required
from django.template import loader, RequestContext
from myapp.models import BlogEntry

@login_required
def detail(request, object_id):
    obj = get_object_or_404(BlogEntry, pk=object_id)
    return render_to_response('object_template.html', {'object': obj})

def index(request):
    """ ohne shortcuts """
    object_list = BlogEntry.objects.all()
    template = loader.get_template('object_list.html')
    context = RequestContext({'object_list': object_list,})
    return HttpResponse(template.render(context))
```

ORM

```
from django.db import models
from django.utils.translation import ugettext_lazy as _

class BlogEntry(models.Model):
    """
    A simple blog entry.
    """
    user = models.ForeignKey(User)
    title = models.CharField(max_length=255)
    slug = models.CharField(max_length=255, unique=True)
    content = models.TextField()
    published = models.DateTimeField(blank=True, null=True, editable=False)

    class Meta:
        verbose_name = _(u'blog entry')
        verbose_name_plural = _(u'blog entries')
        ordering = ('published',)
```

Forms

```
from django import forms
from myapp.models import BlogEntry

class BlogForm(forms.ModelForm):
    class Meta:
        model = BlogEntry
        fields = ('title', 'content',)
```

Forms in Views

```
from django.http import HttpResponseRedirect
from django.shortcuts import render_to_response, get_object_or_404
from myapp.forms import BlogEntryForm
from myapp.models import BlogEntry

def edit(request, object_id):
    obj = get_object_or_404(BlogEntry, pk=object_id)

    if request.method == 'POST':
        form = BlogEntryForm(request.POST, instance=obj)
        if form.is_valid():
            obj = form.save()
            return HttpResponseRedirect('../')

    else:
        form = BlogEntryForm(instance=obj)

    return render_to_response('blog_edit.html', {'form': form, 'obj': obj})
```

Templates

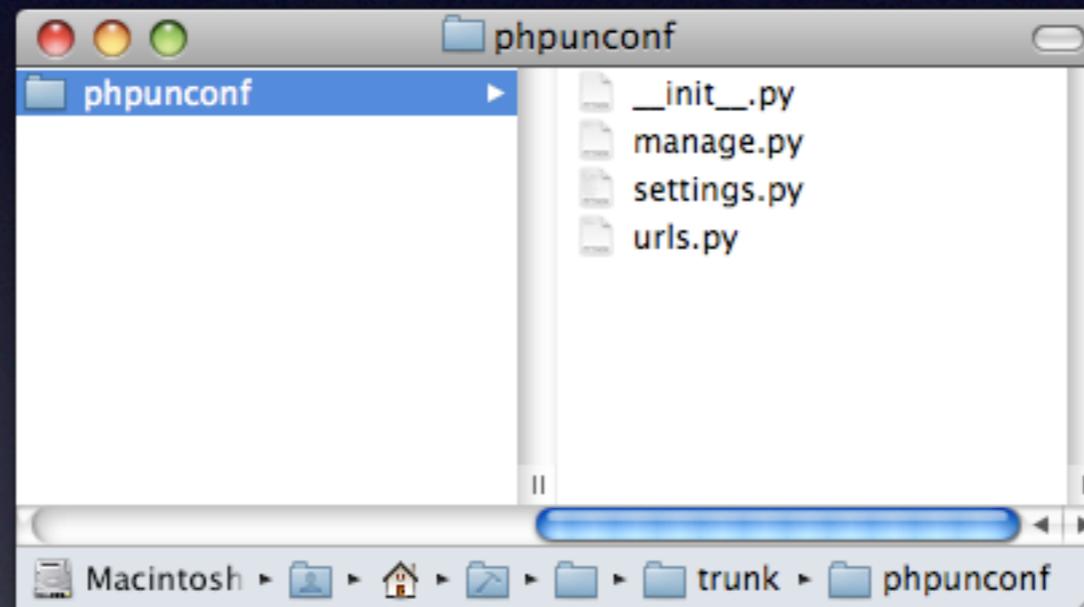
```
{% extends "base.html" %}
{% load i18n %}

{% block extrahead %}{% block.super %}
<script type="text/javascript" charset="utf-8">
    function focus_input(id){document.getElementById(id).focus();}
</script>
{% endblock %}

{% block content %}
<h1>{% trans "Search" %}</h1>
<form action="" method="get">
    {% form.as_p %}
    <input type="submit" value="{% trans "Search" %}"/></p>
</form>
{% if query %}
<p>Die Suche nach <strong>{{ query }}</strong>
    ergab {{ hits|length }} Treffer:</p>
    {% if hits %}
        <ul>{% for hit in hits %}
            <li><a href="{{ hit.path }}">{{ hit.title|safe }}</a></li>
        {% endfor %}</ul>
    {% endif %}
{% endif %}
{% endblock %}
```

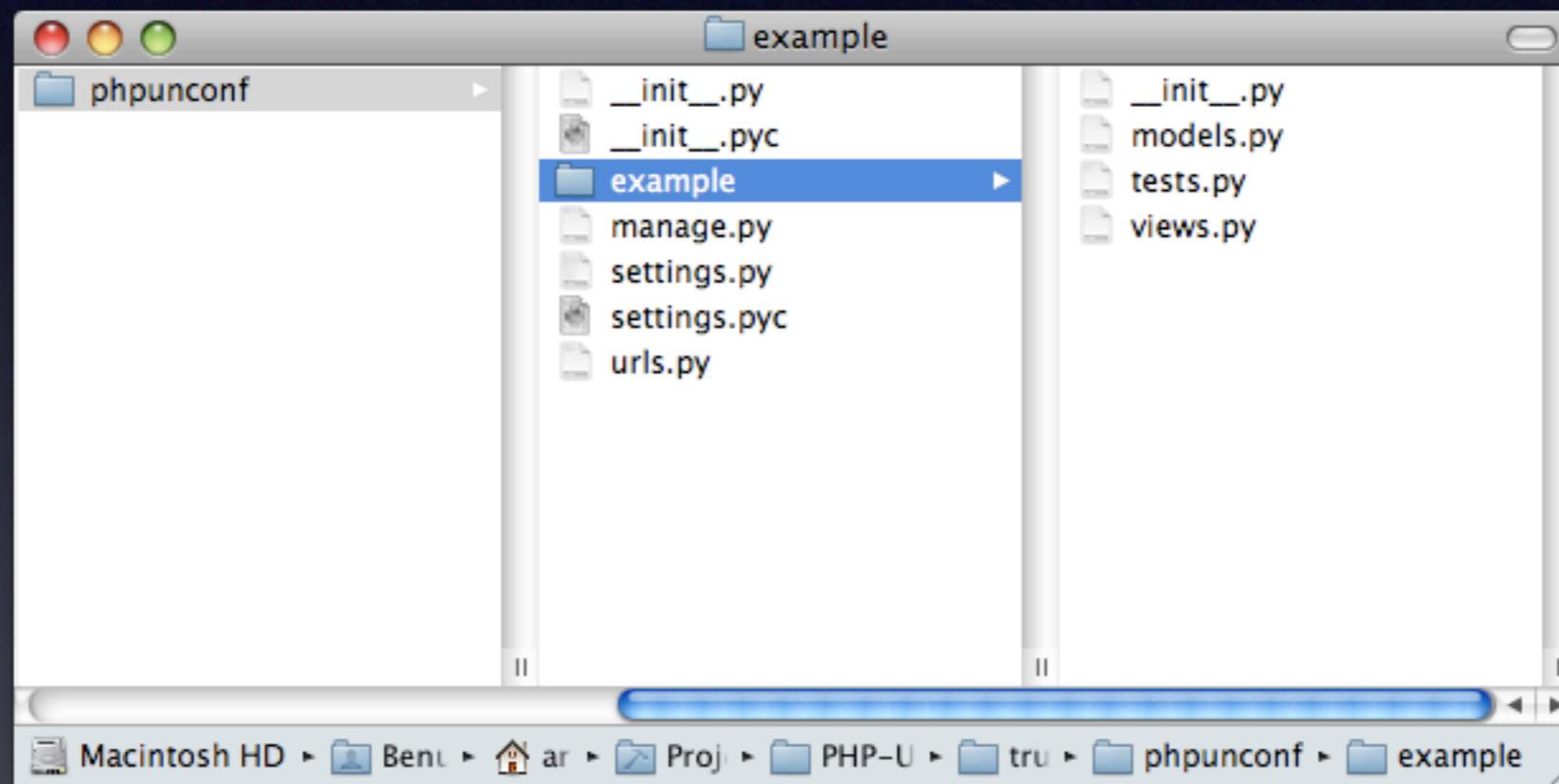
Projekte

django-admin.py startproject phpunconf



Apps

manage.py startapp example



django

The Web framework for perfectionists with deadlines.

Slides: <http://files.arnebrodowski.de/>