

# Django

## Eine praktische Einführung

Ernesto Rico-Schmidt

e.rico.schmidt@gmail.com  
denklab.org

Grazer Linxstage 2009  
April 2009

# Inhalt

## Einführung

# Inhalt

## Einführung

## Django

- Geschichte und Zukunft

- Voraussetzungen

- Installation

- Konfiguration

# Inhalt

## Einführung

## Django

- Geschichte und Zukunft

- Voraussetzungen

- Installation

- Konfiguration

## Entwicklung

- Modelle

- Administrationsoberfläche

- Views

- Templates

# Einführung

## Django

*“Django ist ein Python-Web-Framework für die schnelle Entwicklung von sauberen, pragmatischen Lösungen. ”*

- ▶ Installation und Konfiguration
- ▶ Modelle
- ▶ Administrationsoberfläche
- ▶ Views
- ▶ Templates
- ▶ URL-Konfiguration

# Einführung

## Django

*“Django ist ein Python-Web-Framework für die schnelle Entwicklung von sauberen, pragmatischen Lösungen. ”*

- ▶ Installation und Konfiguration
- ▶ Modelle
- ▶ Administrationsoberfläche
- ▶ Views
- ▶ Templates
- ▶ URL-Konfiguration

*“Python on a Plane”*

# Einführung

bib

- ▶ Persönliche Bibliothek
- ▶ Autoren, Bücher und Plätze
- ▶ ein Buch kann einen oder mehreren Autoren haben
- ▶ ein Buch kann nur an einem Platz sein
- ▶ ein Platz kann mehrere Bücher haben
- ▶ Navigation und Suche.

# Django

## Geschichte und Zukunft

- ▶ ursprünglich von Adrian Holovaty, Simon Wilson, Jacob Kaplan-Moss und Wilson Miner für World-Online in 2003 entwickelt.
- ▶ offen (BSD-Lizenz) seit 2005
- ▶ Version 1.0 in September 2008
- ▶ Version 1.1 in April 2009 (?)
- ▶ Beispiele: <http://www.djangosites.org>



# Django

- ▶ Objektrelationales Mapping (ORM)
- ▶ Automatische Administrationsoberfläche
- ▶ Elegante URL-Konfiguration
- ▶ Templates
- ▶ Views (generische)
- ▶ Cache
- ▶ Internationalisierung

# Django

`django.contrib`

- ▶ `admin`
- ▶ `auth`
- ▶ `comments`
- ▶ `databrowse`
- ▶ `flatpages`
- ▶ `formtools`
- ▶ `localflavor`
- ▶ `markup`
- ▶ `sites`
- ▶ `syndication`

# Django

## Voraussetzungen

- ▶ Python 2.3
- ▶ Apache 2.x, mod\_python 3.x (WSGI, FastCGI, SCGI)
- ▶ PostgreSQL (psycopg)
- ▶ MySQL (MySQLdb)
- ▶ SQLite (pysqlite2, in Python 2.5)
- ▶ Oracle (cx\_oracle)

# Django

## Installation

```
$ cd ~/Development
$ tar xzf ~/Downloads/Django-1.0.2-final.tar.gz
$ cd Django-1.0.2
$ python setup.py install --prefix=$HOME
$ export PYTHONPATH=~/.lib/python2.5/site-packages\
:~/Projects
$ cd ~/Projects
$ django-admin.py startproject glt09
$ cd glt09
$ python manage.py startapp bib
$ python manage.py syncdb
$ python manage.py runserver
```

# Django

Konfiguration: settings.py

```
import os

DATABASE_ENGINE = 'sqlite3'
DATABASE_NAME = os.path.join(os.path.dirname(__file__),
    'glt09.data')
TIME_ZONE = 'Europe/Vienna'
LANGUAGE_CODE = 'de'
TEMPLATE_DIRS = (os.path.join(os.path.dirname(__file__),
    'templates'),)
INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'glt09.bib',
)
```

# Modelle: Author

bib/models.py

```
class Author(models.Model):
    first_name = models.CharField(max_length=128)
    last_name = models.CharField(max_length=128)
    notes = models.TextField(blank=True)

    class Meta:
        ordering = ('last_name', 'first_name')

    def __unicode__(self):
        return '%s, %s' % (self.last_name,
                           self.first_name)

    @models.permalink
    def get_absolute_url(self):
        return ('author-detail', [self.id])
```

# Modelle: Location

bib/models.py

```
class Location(models.Model):
    name = models.CharField(max_length=16)
    description = models.CharField(max_length=32)
    notes = models.TextField(blank=True)

    class Meta:
        ordering = ('name',)

    def __unicode__(self):
        return '%s' % self.name

    @models.permalink
    def get_absolute_url(self):
        return ('location-detail', [self.id])
```

# Modelle: Book I

bib/models.py

```
class Book(models.Model):
    authors = models.ManyToManyField(Author)
    is_editor = models.BooleanField(default=False)
    title = models.CharField(max_length=256)
    publisher = models.CharField(max_length=64)
    year = models.IntegerField()
    volume = models.CharField(max_length=8, blank=True)
    series = models.CharField(max_length=256, blank=True)
    address = models.CharField(max_length=64, blank=True)
    edition = models.CharField(max_length=8, blank=True)
    pages = models.CharField(max_length=8, blank=True)
    isbn = models.CharField('ISBN', max_length=16,
        blank=True)
    location = models.ForeignKey(Location)
    notes = models.TextField(blank=True)
```



# Modelle: Book II

`bib/models.py`

```
class Meta:
    ordering = ('title',)

def __unicode__(self):
    return '%s' % self.title

@models.permalink
def get_absolute_url(self):
    return ('book-detail', [self.id])
```

# Administrationsoberfläche

URL-Konfiguration: `urls.py`

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

admin.autodiscover()

urlpatterns = patterns('',
    ...
    (r'^admin/(.*)', admin.site.root),
    ...
)
```

# Administrationsoberfläche

Konfiguration: `settings.py`

```
INSTALLED_APPS = (  
    ...  
    'django.contrib.admin',  
    ...  
)
```

# Administrationsoberfläche

bib/admin.py

```
from django.contrib import admin
from models import Author, Book, Location

class AuthorAdmin(admin.ModelAdmin):
    list_display = ('last_name', 'first_name')
class BookAdmin(admin.ModelAdmin):
    list_display = ('title', 'location')
    list_filter = ('location', 'publisher')
    search_fields = ('title', 'series')
class LocationAdmin(admin.ModelAdmin):
    list_display = ('name', 'description')

admin.site.register(Author, AuthorAdmin)
admin.site.register(Book, BookAdmin)
admin.site.register(Location, LocationAdmin)
```

# Generische Views

- ▶ `django.views.generic.simple`
  - ▶ `direct_to_template`
  - ▶ `direct_to`
- ▶ `django.views.generic.date_based`
  - ▶ `archive_index`
  - ▶ `archive_year`
  - ▶ `archive_month`
  - ▶ `archive_week`
  - ▶ `archive_day`
  - ▶ `archive_today`
  - ▶ `object_detail`
- ▶ `django.views.generic.list_detail`
  - ▶ `object_list`
  - ▶ `object_detail`
- ▶ `django.views.generic.create_update`
  - ▶ `create_object`
  - ▶ `update_object`
  - ▶ `delete_object`

# Generische Views

URL-Konfiguration: `urls.py`

```
from django.conf.urls.defaults import *
from django.views.generic.list_view import *
from models import Author, Book, Location

authors = {'queryset': Author.objects.all(),}
books = {'queryset': Book.objects.all(),}
locations = {'queryset': Location.objects.all(),}

urlpatterns = patterns('',
    url(r'^bib/author/$', object_list, authors),
    url(r'^bib/author/(?P<object_id>\d+)/$', object_detail,
        authors, name='author-detail'),
    url(r'^bib/book/$', 'object_list', books),
    url(r'^bib/book/(?P<object_id>\d+)/$', object_detail,
        books, name='book-detail'),
    url(r'^bib/location/(?P<object_id>\d+)/$', object_detail,
        locations, name='location-detail'),)
```

# Suche

Views: `bib/views.py`

```
from django.shortcuts import render_to_response
from models import Author, Book

def search(request):
    authors_list =
        Author.objects.filter(last_name__contains=
            request.GET['query'])

    books_list =
        Book.objects.filter(title__contains=
            request.GET['query'])

    return render_to_response('bib/results.html',
        {'authors_list': authors_list,
         'books_list': books_list,})
```

# Suche

URL-Konfiguration: `urls.py`

```
from bib.views import search

urlpatterns = patterns('',
    ...
    (r'^bib/search/$', search),
    ...
)
```



# Basis-Template

base.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <title>GLT'09: bib</title>
</head>
<body>
<form action="/bib/search/" method="GET">
<input name="query" type="text">
<input type="submit">
</form>
<hr>
{% block main %}
{% endblock %}
</body>
```

# Templates

bib/author\_list.html

```
{% extends "base.html" %}
{% block main %}
<h1>Authors</h1>
<ul>
{% for author in object_list %}
    <li>
        <a href="{{ author.get_absolute_url }}">
            {{ author.last_name }}, {{ author.first_name }}</a>
            {{ author.book_set.count }} book(s)
        </li>
{% endfor %}
</ul>
{% endblock %}
```

# Templates

`bib/author_detail.html`

```
{% extends "base.html" %}
{% block main %}
<h1>Author: {{ object.last_name }},
    {{ object.first_name }}</h1>
{% if object.book_set %}
<h2>Book(s)</h2>
<ul>
    {% for book in object.book_set.all %}
    <li>
        <a href="{{ book.get_absolute_url }}">
            {{ book.title }}</a>, {{ book.publisher }},
            {{ book.year }}</li>
    {% endfor %}
</ul>
{% endif %}
{% endblock %}
```

# Templates

bib/book\_list.html

```
{% extends "base.html" %}
{% block main %}
<h1>Books</h1>
<ul>
{% for book in object_list %}
  <li>
    <a href="{{ book.get_absolute_url }}">
      {{ book.title }}</a> by
    {% for author in book.author_set.all %}
      <a href="{{ author.get_absolute_url }}">
        {{ author.last_name }}, {{ author.first_name }}</a>
    {% endfor %}
  </li>
{% endfor %}
</ul>
{% endblock %}
```

# Templates

bib/book\_detail.html

```
{% extends "base.html" %}
{% block main %}
<h1>Book: {{ object.title }}</h1>
<p>
{% for author in object.author_set.all %}
    <a href="{{ author.get_absolute_url }}">
        {{ author.last_name }}, {{ author.first_name }}</a>
{% endfor %}
<p>{{ object.publisher }}, {{ object.year }}
<p>
Location:
<a href="{{ object.location.get_absolute_url }}">
{{ object.location.name }}</a>
{% endblock %}
```

# Templates

`bib/location_detail.html`

```
{% extends "base.html" %}
{% block main %}
<h1>Location: {{ object.name }}</h1>
<p>
{{ object.description }}
<h2>Book(s)</h2>
<ul>
{% for book in object.book_set.all %}
    <li>
        <a href="{{ book.get_absolute_url }}">
            {{ book.title }}</a>,
        {% for author in book.authors.all %}
            {{ author.last_name }}, {{ author.first_name }}
        {% endfor %}
        {{ book.publisher }}, {{ book.year }}
    </li>
{% endfor %}
```

# Suchergebnisse I

bib/results.html

```
{% extends "base.html" %}
{% block main %}
<h1>Search results</h1>
{% if authors_list %}
<h2>Author(s)</h2>
<ul>
  {% for author in authors_list %}
    <li>
      <a href="{ { author.get_absolute_url } }">
        { { author.last_name } },
        { { author.first_name } }</a>
    </li>
  {% endfor %}
</ul>
{% endif %}
```

# Suchergebnisse II

[bib/results.html](#)

```
{% if books_list %}
<h2>Book(s)</h2>
<ul>
  {% for book in books_list %}
    <li>
      <a href="{{ book.get_absolute_url }}">
        {{ book.title }}</a>
        {% for author in book.authors.all %}
          {{ author.last_name }}, {{ author.first_name }}
        {% endfor %}
        {{ book.publisher }}, {{ book.year }}
      </li>
    {% endfor %}
  </ul>
{% endif %}
{% endblock %}
```



# Ausblick

*“Django ist ein Python-Web-Framework für die schnelle Entwicklung von sauberen, pragmatischen Lösungen. ”*

- ▶ Installation und Konfiguration
- ▶ Modelle
- ▶ Administrationsoberfläche
- ▶ Views
- ▶ Templates
- ▶ URL-Konfiguration

Quellen für mehr Information:

- ▶ <http://www.djangoproject.com>
- ▶ <http://code.djangoproject.com>
- ▶ <http://www.djangobook.com>