



*software
projekte
prozesse

eck*cellent IT

* LINEAS Project Services heißt jetzt **eck*cellent IT** - besuchen Sie uns auf www.eckcellent-it.de



eck*cellent IT GmbH Standorte in Braunschweig und Dortmund / Germany:
Theodor-Heuss-Straße 2 38122 Braunschweig Phone +49 (0)531.70 22-20 00
Emil-Figge-Straße 76 44227 Dortmund Phone +49 (0)231.97 42-55 80



- * Thema: Entwicklung der Zeiterfassung „Theo“ mit Python und Django



Referent: Gerald Winter
Dipl.-Informatiker

Agenda

- Python
 - Überblick zu Python
 - Ziele, Paradigmen und Eigenschaften
 - Beispiel
 - Python in der Praxis
 - Ausblick Python
- Django
 - Überblick zu Django
 - Bestandteile von Django am Beispiel
 - Django in der Praxis
- Theo
 - Was ist Theo?
 - Verwendete Technologien
 - Erfahrungen und Vorteile mit Python & Django
- Zusammenfassung



Motivation

- Aufgabe: Schnelle Entwicklung eines Zeiterfassungssystems
- Problem: Entwicklung mit JEE meist aufwändig
 - Vermischung mehrerer Technologien
 - Java (Businesslogik)
 - XML (Konfiguration, Datenrepräsentation)
 - XSLT (View-Erzeugung)
 - SQL (Persistenzzugriff, Datenbankpflege)
 - Aufwändig erlernbar
 - Erhöhter Konfigurationsaufwand
 - Erhöhte Fehlerwahrscheinlichkeit in den Schnittstellen
- Python & Django versprechen:
 - Kurze Entwicklungszeiten
 - Abnahme und Automatisierung vieler typischer Aufgaben
- Schnelle Erlernbarkeit



Python



- interpretierte Sprache
- Open Source
 - C / C++
 - Auch für kommerzielle Nutzung
- Geschichte
 - Benannt nach „Monty Python's Flying Circus“
 - Entwickelt 1990 von Guido van Rossum in „Stichting Mathematisch Centrum“, Niederlande
 - 2001: Gründung „Python Software Foundation“
 - Jan. 2008: Aktuelle Version 2.5.1
 - Aug. 2008: geplanter Release 3.0.0 (Python 3000)



Ziele der Sprache

- Übersichtlicher Code
- Schnelle Erlernbarkeit
- Einbindungsmöglichkeit von Modulen anderer Sprachen
- Kurze Entwicklungszeiten
- Umfangreiche Standard-Bibliothek
- Verfügbarkeit auf vielen Plattformen
 - Windows, Linux, Unix, Mac, Amiga, ...
- Weitreichende Anwendungsgebiete
 - Web and Internet Development
 - Spiele und 3D-Grafik
 - ...



Sprachparadigmen

- Objektorientierte Programmierung
 - Garbage Collection
 - Mehrfachvererbung
- Funktionale Programmierung
- Strukturierte Programmierung
- Modularisierbarkeit durch Packages / Imports
- Dynamische Typverwaltung
- Exception-Handling



Spracheigenschaften

- Grunddatentypen
 - Ganzzahlen
 - Fließkommazahlen
 - Komplexe Zahlen
 - Strings
 - Listen
 - Tupel
 - Set
 - Dictionary
- Einrücken statt Klammern
- Diverse Kurzschreibweisen

```
year, month, day = getCurrentDateData()
```

```
substring = string[3:]
```



Python Code-Beispiel

➤ Der Quicksort-Algorithmus in Python

```
def quicksort(liste):  
    if len(liste) <= 1:  
        return liste  
    pivotelement = liste[0]  
    links = [element for element in liste[1:] if element < pivotelement]  
    rechts = [element for element in liste[1:] if element >= pivotelement]  
    return quicksort(links) + [pivotelement] + quicksort(rechts)
```

Quelle: Wikipedia



Python in der Praxis

- Dezember 2007 Python beliebter als Perl und C#
 - Laut „The TIOBE Programming Community“
- „The Computer Language Benchmarks Game“
 - Python oft schneller als Perl
 - Python deutlich langsamer als Java, C, C++



Ausblick für Python

- Python 3000
 - Release voraussichtlich August 2008
 - Inkompatibel zu 2.x-Versionen
 - Änderungen in Syntax, Bibliothek, Datentypen
- PyPy
 - Python-Interpreter, der in Python geschrieben ist
 - Just in Time-Compiler für höhere Geschwindigkeiten
 - Benutzerdefinierte Python-Versionen
 - Geplant: Erkenntnisse aus dem JIT-Compiler „Psyco“ integrieren



Django

- High-Level Python Web Framework
- Philosophien
 - Lose Kopplung
 - So wenig Code wie möglich
 - Schnelle Entwicklung
 - DRY-Prinzip (Don't repeat yourself)
 - Logik von Präsentation trennen
- MVC-Konzept
 - MTV = Model, Template, View
 - Model-Erstellung in Python (ORM)
 - Oberflächen-Generierung durch Template-Engine
 - Business-Logik (Views) in Python



Django

➤ Geschichte

- 2003: Adrian Holovaty und Simon Willison von World Online trennen sich von PHP und beginnen Web-Entwicklung mit Python
- 2005: World-Online gibt das erstellte Framework als Open Source frei

➤ Hauptbestandteile

- Automatische konfigurierbare Objektrelationale Abbildung (ORM)
- Automatische konfigurierbare Administrationsoberfläche
- Vereinfachtes URL-Design
- Template-Engine für Oberflächen
- Cache-Framework
- Internationalisierungsfunktionalitäten



Objektrelationale Abbildung

- Erstellung des Datenmodells in Python
- Synchronisation mit der Datenbank
- API zur Suche und Modifikation der Daten

```
class HolidayRule(models.Model):  
    """Feiertagsregeln"""  
    region = models.CharField(maxlength=10)  
    remark = models.CharField(maxlength=100, blank=True,  
                              null=True)  
    holiday = models.ForeignKey(Holiday, edit_inline=True)
```

```
ruleList = HolidayRule.objects.filter(holiday=currentHoliday)
```



Administrationsoberfläche

- Ebenfalls über das Model konfigurierbar
- Screenshot:

The screenshot shows the Django administration interface for editing a holiday rule. The page title is "Django Verwaltung" and the user is logged in as "Gerald". The breadcrumb trail is "Start > Feiertagsregeln > DE: 2. Weihnachtstag". The main heading is "Feiertagsregel ändern" with a "Geschichte" button. The form contains the following fields:

- Region:** DE
- Kommentar:** (empty text area)
- Feiertag:** 2. Weihnachtstag (selected in a dropdown menu)

The dropdown menu for "Feiertag" is open, showing a list of German holidays. The selected item is "2. Weihnachtstag". Other visible items include "Heiligabend", "Neujahr", "Tag der deutschen Einheit", "Ostersonntag", "Rosenmontag", "Aschermittwoch", "Christi Himmelfahrt", "Pfingstsonntag", "Fronleichnam", "Ostermontag", "Karfreitag", "Pfingstmontag", "Maifeiertag", "Reformationstag", "Allerheiligen", "1. Weihnachtstag", "Buß- und Betttag", and "Mariä Himmelfahrt".

At the bottom of the form, there are three buttons: "Löschen" (with a red 'x' icon), "Sichern und neu hinzufügen", and "Sichern und weiter bearbeiten". A blue "Sichern" button is also visible on the right side of the form.



URL-Design

➤ URL-Funktions-Mapping durch reguläre Ausdrücke

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

urlpatterns = patterns('mops.trs.views',

    (r'^overview/tu(?P<timeuserid>\d+)/(?P<year>\d+)-(?P<month>\d+)/$',
      'monthOverview'),
    (r'^export/(?P<year>\d+)-(?P<month>\d+)/$',
      'selectExportType'),
    (r'^export/(?P<year>\d+)-(?P<month>\d+)/month/$',
      'selectReportTimeUser'),
    (r'^export/(?P<year>\d+)-(?P<month>\d+)/month/tu(?P<timeuserid>\d+)/$',
      'selectReportTemplate'),
    ...
)
```



View-Erstellung (Business-Logik)

- Angesiedelt zwischen Datenbank und Oberfläche
- Parametrisierter Funktionsaufruf abhängig von URL
- Initiiert Oberflächen-Rendering via Template-Engine
- Übergabe der darzustellenden Daten als „Context“

```
def selectReportTemplate(request, year, month, timeuserid):  
    templates = getTemplates(ReportTemplate.Type.MONTH)  
    context = {}  
    context['templates'] = templates  
    context['request'] = request  
    context['month'] = month  
    context['year'] = year  
    return render_to_response("trs/reportSelectTemplate.html", context)
```



Template-Engine

- Keine HTML-Abhängigkeit
- Zugriff auf „Context“-Objekt aus der Business-Logik
- Template-Vererbung
- Erweiterbar durch eigene Tags in Python

```
{% load i18n %}
{% load layout_extras %}
{% load trs_extras %}
{% extends "base_theo.html" %}

{% block subtitle %}
    {% trans "trs.common.selectTemplate" %}
{% endblock %}

{%block navigation %}
    {%monthYearNavigation request month year %}
{%endblock%}

{% block content %}
    <ul>
        {% for template in templates %}
            <li>
                <a href="{{request.path}}{{template.key}}"
                    type= "{{template.mimetype}}"
                    {{template.name}}
                </a>
            </li>
        {% endfor %}
    </ul>
{% endblock %}
```



Django in der Praxis

- www.djangosites.org listet 997 Django-Websites
- Ned Batchelder von www.tabblo.com (Eigentum von Hewlett-Packard) schrieb:
 - „Before I joined Tabblo, Antonio Rodriguez (Tabblo's founder/CTO) did an evaluation of Rails and Django, and found that both provided a great quick-out-of-the-blocks rapid development environment. In comparing the two, he found that Django had a greater technical depth that would make it easier to build a robust, scalable site. Also, Django's Python foundation meant that we'd have all the richness of the Python ecosystem to support our work. This has definitely been proven out as we've built Tabblo.“
 - Quelle: <http://www.djangobook.com/en/1.0/appendixA/>



Theo

- Webanwendung
 - Mitarbeiter, Organisations- und Projektmanagementsystem (M.O.P.S.)
 - Modularer Aufbau
 - Core
 - Zeiterfassungssystem
 - Urlaubsverwaltung
 - Projektmanagement
 - Mitarbeiter-Profil-Management
 - CRM
 - ...
 - Ablösung des vorherigen Zeiterfassungssystems



Verwendete Technologien

- Django
- Python
- ExtJS / Javascript
- HTML
- Excel (Reports)
- Trac
- OpenOffice
- SVN
- Dia
- ...



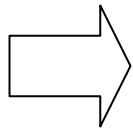
Vorführung Theo

- Erstellung der Persistenz:
 - Eingabe: `python manage.py syncdb`
 - Datenbank-Datei für SQLite wird automatisch erstellt
 - Initialeinträge werden vorgenommen (Fixtures)
- Start des Servers:
 - Eingabe: `python manage.py runserver 8080`
- Aufruf von Theo im Browser
 - Eingabe: `http://localhost:8080/mops/`



Vorteile durch Verwendung von Django

- Automatische Datenbank-Erstellung / -Verwaltung
- Schnelle Testbarkeit / Rollbacks durch SQLite-DB
- Templates erlauben
 - Leichte und schnelle Oberflächenerstellung
 - Saubere Trennung zwischen Oberfläche und Logik
- Frühe Prototypen durch Django Auto-Forms
- Leichte Konfiguration direkt in Python-Code
 - Server-Einstellungen
 - URLs
- Testdatenerzeugung über Django-Admin / Fixtures



Entwicklungszeit Core, Zeiterfassung,
Urlaubsmanagement ca. 60 MT



Zusammenfassung

- Python ist übersichtlich, schnell erlernbar und vielseitig einsetzbar
- Django ist ein auf Python basierendes umfangreiches MVC-Framework für Web-Anwendung
- Beides soll gemeinsam eine schnelle Entwicklung von Web-Anwendung ermöglichen
- Django hat sich bei der Entwicklung eines neuen Zeiterfassungssystems bewährt



Referenzen

> Python

- > <http://www.python.org>
- > http://de.wikipedia.org/wiki/Python_%28Programmiersprache%29
- > <http://www.tiobe.com/tpci.htm>
- > <http://shootout.alioth.debian.org/debian/python.php>
- > <http://psyco.sourceforge.net/introduction.html>
- > <http://codespeak.net/pypy/>

> Django

- > <http://www.djangoproject.com/>
- > <http://www.djangosites.org/>
- > <http://www.djangobook.com/>





*software
projekte
prozesse

eck*cellent IT

* Vielen Dank für Ihre Aufmerksamkeit!

