

Python Webframeworks Django

Yunus Sari – Muhammedhizir Kanyildiz

Inhaltsverzeichnis

1	Was ist Django?	1
1.1	Erklärung.....	1
1.2	Vorteile.....	2
1.3	Nachteile.....	2
1.4	Beispiele	3
2	Installation.....	5
3	Konfiguration.....	8
3.1	Erstellung eines Projektes	8
3.2	Ausführen des Django-Servers	8
3.3	Klonen eines Projektes.....	8
4	Anwendungsfall.....	Error! Bookmark not defined.
5	Datensätze	Error! Bookmark not defined.
6	Anlegen/Bearbeiten/Löschen von Datensätzen	Error! Bookmark not defined.
7	Schwierigkeiten	10
7.1	Environment Variable	Error! Bookmark not defined.
8	Aufwandschätzung.....	12

1 Was ist Django?

1.1 Erklärung

Django ist ein in Python geschriebenes Full Stack Framework, dass die schnelle Entwicklung von Web-Applikationen ermöglicht. Dabei wird Wert auf sauberen Code und die Wiederverwendbarkeit der einzelnen Komponenten gelegt. Der Quellcode und die umfangreiche Dokumentation sind unter einer BSD-Lizenz lizenziert. Die Django Software Foundation stellt die Weiterentwicklung von Django sicher.

Django unterstützt durch seine Architektur und Werkzeuge eine schnelle Entwicklung (Rapid Development) von Websites und neuen Komponenten.

Loose Coupling, also die lose Koppelung der verschiedenen Teile des Frameworks und der Applikation, stehen bei Django stark im Vordergrund. Dadurch soll die Qualität und Wiederverwendbarkeit des Codes erhöht werden. [1]

1.2 Model-Template-View

Django ist nach dem Model-Template-View (MTV) Muster aufgebaut. MTV orientiert sich am bekannten Model-View-Controller Muster (MVC).

Der in Django enthaltene Object Relational Mapper (ORM) überträgt die Models in Datenbankstrukturen und führt alle Operationen in der Datenbank durch. Es können alle gängigen Datenbanken benutzt werden. Alle Models werden in Python geschrieben.

Die Template-Engine unterstützt die Vererbung von Templates und bietet umfangreiche Filter und Templates. Diese können auch selbst erweitert werden. Der View holt die Daten, zum Beispiel mit Hilfe des Object Relational Mappers. Es können aber auch anderen Datenquellen genutzt werden. Die URLConf steuert das Routing. Mit Hilfe von regulären Ausdrücken wird der Request dem richtigen View zugewiesen. [2]

1.3 Vorteile

Alle großen Python-Web-Frameworks bieten sich dann an, wenn eine Anwendung in relativ kurzer Zeit, möglichst sauber implementiert werden soll. Auf dieses Gebiet ist bereits Python ausgerichtet. Unterstützt wird dies durch die bereits standardmäßig in Python-Interpretern enthaltenen Module. Viele von ihnen erfüllen bereits wichtige Aufgaben die im Internet benötigt werden und sind mit wenig Einarbeitungszeit leicht zu verwenden.

Auch ist Python nicht proprietär, wird also nicht von einem Unternehmen entwickelt, welches seine Interessen bestmöglich durchsetzen will. Sinnvolle Erweiterungen der Sprache werden ständig durchgeführt und erst nach ausgiebiger Diskussion übernommen. Hinzu kommt die Plattformunabhängigkeit der Sprache, so dass die Web-Frameworks auf nahezu jedem größeren Betriebssystem laufen.

Für Entwickler ist natürlich eine besonders gute Dokumentation sehr wichtig. Zu allen drei vorgestellten Web-Frameworks ist eine solche, auch in sehr detaillierter Form, vorhanden. Die Dokumentationen beziehen sich primär auf das Entwickeln neuer Anwendungen und gehen weniger auf konkrete Details des Systems ein. Dies erleichtert vor allem das Erstellen von Applikationen.

Ein besonders wichtiger Punkt ist die große Auswahl an verschiedenen Web-Frameworks. Diese ermöglicht es, das für sich passende zu wählen und zu verwenden. Außerdem bestehen für die größeren Frameworks bereits eine Reihe von vorgefertigten Anwendungen, wie Shops und Gästebücher, die zur Verwendung nur noch eingebunden werden müssen. [3]

1.4 Nachteile

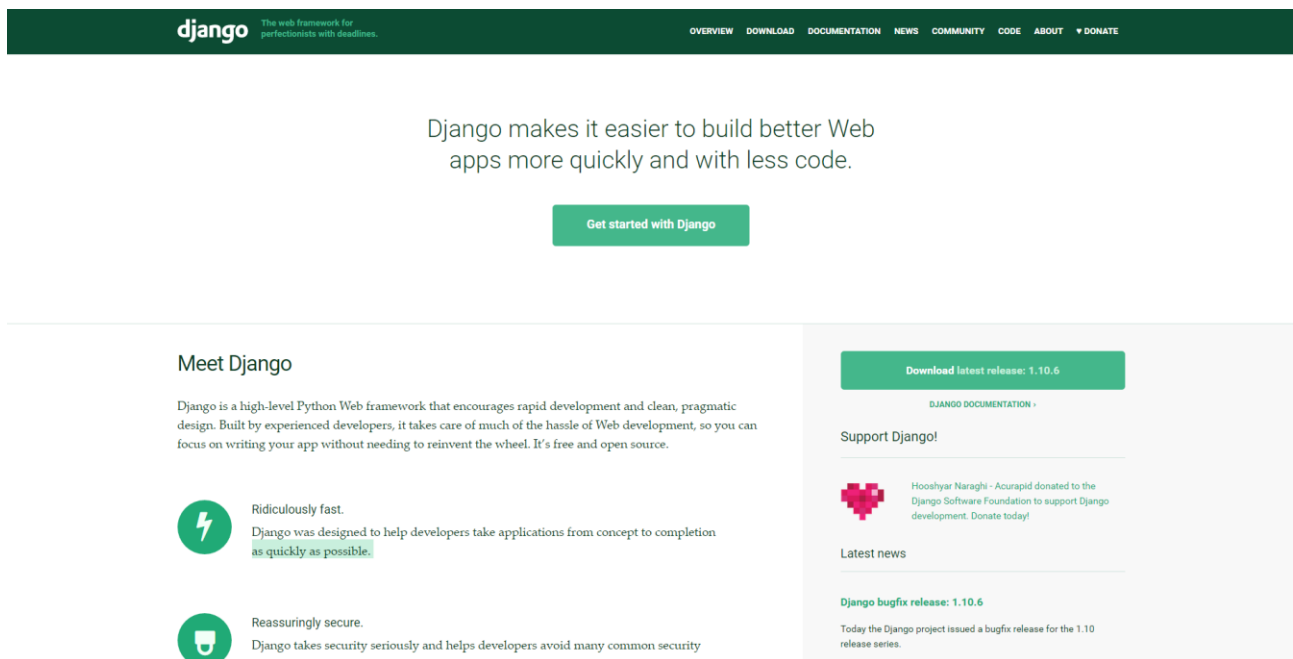
Aber auch das Einbringen von Python, und Anwendungen die in dieser Sprache geschrieben sind, in ein Unternehmen wurde bis vor kurzem erschwert, da nahezu kein Marketing auf Seiten der Python Software Foundation und auch nicht bei den Web-Frameworks bestand. Dies ist aber oftmals ein entscheidender Grund für einen Manager ein bestimmtes Produkt zu verwenden oder nicht, da für diesen Sprach-Konzepte von Programmiersprachen wenig interessant sind.

Auch ist Python ursprünglich nicht primär für das Web ausgelegt, bietet aber schon länger sehr gute Module für diese Aufgabe. Daher konzentriert man sich lediglich auf den logischen Teil von Web-Anwendungen und bietet die Gestaltung der Seiten über die bewährte Technik der Templates an. [3]

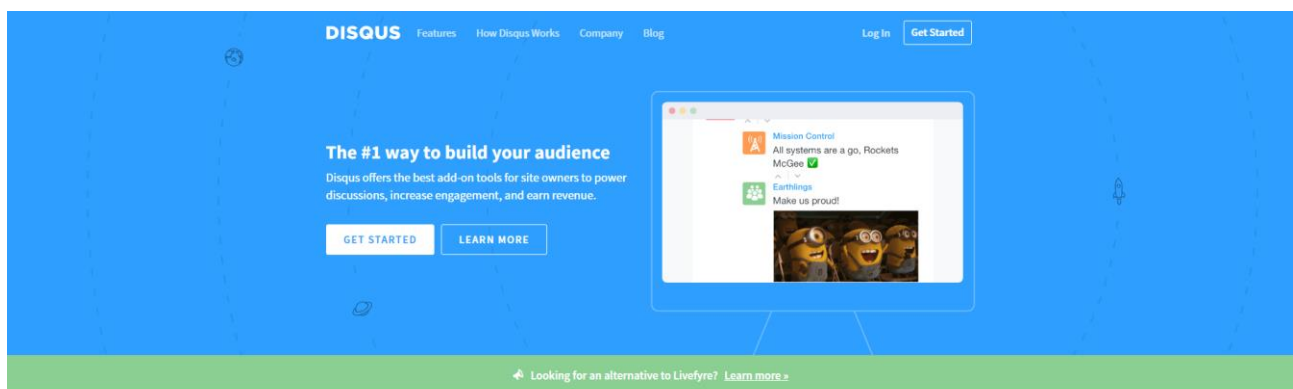
1.5 Beispiele

Beispiele von Webseiten, die Django benutzen:

- <https://www.djangoproject.com/start/overview/>



- <https://disqus.com/>



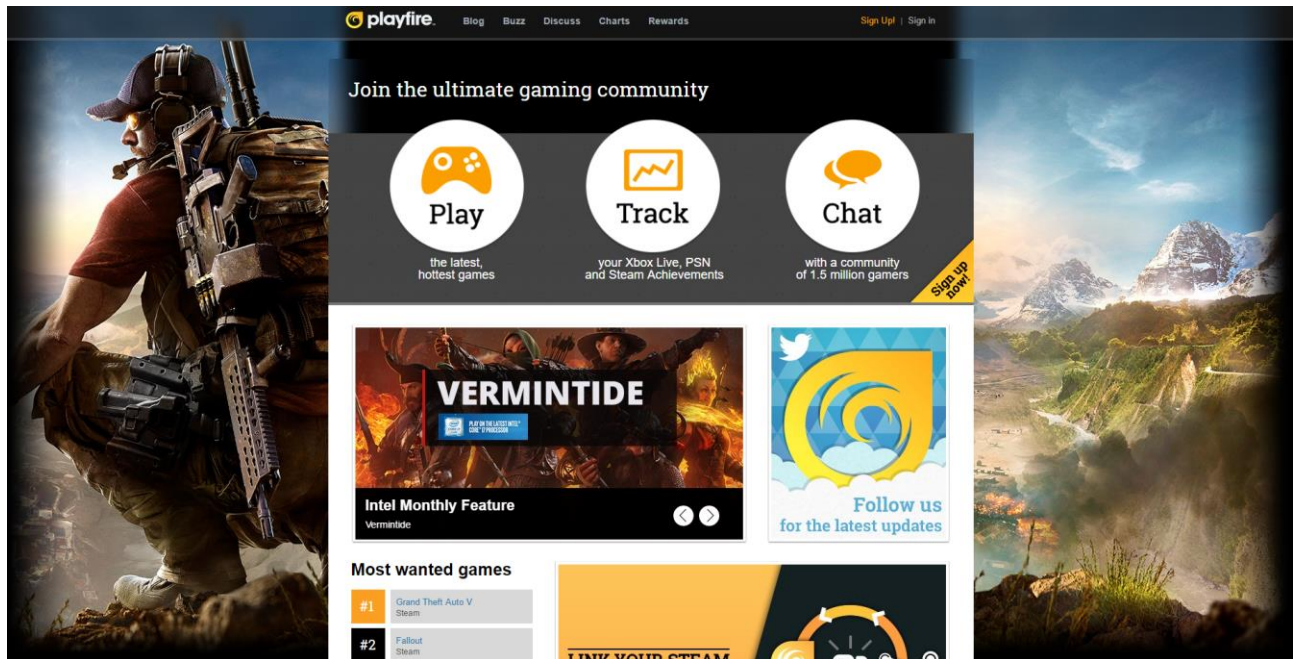
Spark engagement with comments

Engage by Disqus is the world's most popular comment system. It makes communities easier for publishers to manage, and readers love using it.

[LEARN ABOUT ENGAGE](#)



- <https://www.playfire.com/>



- <https://www.washingtonpost.com/>



2 Installation

Vor der Installation wurde ein Update durchgeführt.

```
OK http://ftp.at.debian.org jessie Release.gpg
OK http://ftp.at.debian.org jessie-updates/main Sources
OK http://security.debian.org jessie/updates/main Sources
Holen: 1 http://ftp.at.debian.org jessie-updates/main amd64 Packages/DiffIndex [
6.916 B]
OK http://security.debian.org jessie/updates/main amd64 Packages
Ign http://pkg.jenkins.io binary/ InRelease
Holen: 2 http://ftp.at.debian.org jessie-updates/main Translation-en/DiffIndex [
2.704 B]
OK http://security.debian.org jessie/updates/main Translation-en
OK http://ftp.at.debian.org jessie Release
OK http://ftp.at.debian.org jessie/main Sources
OK http://pkg.jenkins.io binary/ Release.gpg
OK http://ftp.at.debian.org jessie/main amd64 Packages
OK http://ftp.at.debian.org jessie/main Translation-de
OK http://ftp.at.debian.org jessie/main Translation-en
OK http://pkg.jenkins.io binary/ Release
OK http://pkg.jenkins.io binary/ Packages
Ign http://pkg.jenkins.io binary/ Translation-de_AT
Ign http://pkg.jenkins.io binary/ Translation-de
Ign http://pkg.jenkins.io binary/ Translation-en
Es wurden 9.620 B in 3 s geholt (2.658 B/s).
Paketlisten werden gelesen... Fertig
root@sew:~# sudo apt-get update
```

```
Ign http://pkg.jenkins.io binary/ Translation-de_AT
Ign http://pkg.jenkins.io binary/ Translation-de
Ign http://pkg.jenkins.io binary/ Translation-en
Es wurden 9.620 B in 3 s geholt (2.658 B/s).
Paketlisten werden gelesen... Fertig
root@sew:~# sudo apt-get update
Ign http://ftp.at.debian.org jessie InRelease
OK http://ftp.at.debian.org jessie-updates InRelease
OK http://security.debian.org jessie/updates InRelease
OK http://ftp.at.debian.org jessie Release.gpg
OK http://ftp.at.debian.org jessie Release
OK http://ftp.at.debian.org jessie-updates/main Sources
Holen: 1 http://ftp.at.debian.org jessie-updates/main amd64 Packages/DiffIndex [
6.916 B]
OK http://security.debian.org jessie/updates/main Sources
Holen: 2 http://ftp.at.debian.org jessie-updates/main Translation-en/DiffIndex [
2.704 B]
OK http://ftp.at.debian.org jessie/main Sources
OK http://security.debian.org jessie/updates/main amd64 Packages
OK http://ftp.at.debian.org jessie/main amd64 Packages
OK http://ftp.at.debian.org jessie/main Translation-de
OK http://security.debian.org jessie/updates/main Translation-en
OK http://ftp.at.debian.org jessie/main Translation-en
Ign http://pkg.jenkins.io binary/ InRelease
```


Nach diesem Schritt wurde Pip installiert

```
Es wurden 9.620 B in 3 s geholt (2.658 B/s).
Paketlisten werden gelesen... Fertig
root@sew:~# sudo apt-get update
Ign http://ftp.at.debian.org jessie InRelease
OK http://ftp.at.debian.org jessie-updates InRelease
OK http://security.debian.org jessie/updates InRelease
OK http://ftp.at.debian.org jessie Release.gpg
OK http://ftp.at.debian.org jessie Release
OK http://ftp.at.debian.org jessie-updates/main Sources
Holen: 1 http://ftp.at.debian.org jessie-updates/main amd64 Packages/DiffIndex [
6.916 B]
OK http://security.debian.org jessie/updates/main Sources
Holen: 2 http://ftp.at.debian.org jessie-updates/main Translation-en/DiffIndex [
2.704 B]
OK http://ftp.at.debian.org jessie/main Sources
OK http://security.debian.org jessie/updates/main amd64 Packages
OK http://ftp.at.debian.org jessie/main amd64 Packages
OK http://ftp.at.debian.org jessie/main Translation-de
OK http://security.debian.org jessie/updates/main Translation-en
OK http://ftp.at.debian.org jessie/main Translation-en
Ign http://pkg.jenkins.io binary/ InRelease
OK http://pkg.jenkins.io binary/ Release.gpg
OK http://pkg.jenkins.io binary/ Release
OK http://pkg.jenkins.io binary/ Packages
Ign http://pkg.jenkins.io binary/ Translation-de_AT
Ign http://pkg.jenkins.io binary/ Translation-de
Ign http://pkg.jenkins.io binary/ Translation-en
Es wurden 9.620 B in 3 s geholt (3.191 B/s).
Paketlisten werden gelesen... Fertig
root@sew:~# apt-get install python3-pip
OK http://ftp.at.debian.org jessie Release.gpg
OK http://ftp.at.debian.org jessie Release
OK http://ftp.at.debian.org jessie-updates/main Sources
Holen: 1 http://ftp.at.debian.org jessie-updates/main amd64 Packages/DiffIndex [
6.916 B]
OK http://security.debian.org jessie/updates/main Sources
Holen: 2 http://ftp.at.debian.org jessie-updates/main Translation-en/DiffIndex [
2.704 B]
OK http://ftp.at.debian.org jessie/main Sources
OK http://security.debian.org jessie/updates/main amd64 Packages
OK http://ftp.at.debian.org jessie/main amd64 Packages
OK http://ftp.at.debian.org jessie/main Translation-de
OK http://security.debian.org jessie/updates/main Translation-en
OK http://ftp.at.debian.org jessie/main Translation-en
Ign http://pkg.jenkins.io binary/ InRelease
OK http://pkg.jenkins.io binary/ Release.gpg
OK http://pkg.jenkins.io binary/ Release
OK http://pkg.jenkins.io binary/ Packages
Ign http://pkg.jenkins.io binary/ Translation-de_AT
Ign http://pkg.jenkins.io binary/ Translation-de
Ign http://pkg.jenkins.io binary/ Translation-en
Es wurden 9.620 B in 3 s geholt (3.191 B/s).
Paketlisten werden gelesen... Fertig
root@sew:~# apt-get install python3-pip
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut.
Statusinformationen werden eingelesen.... Fertig
python3-pip ist schon die neueste Version.
0 aktualisiert, 0 neu installiert, 0 zu entfernen und 0 nicht aktualisiert.
root@sew:~#
```

Django wurde installiert.

```
OK http://ftp.at.debian.org jessie Release.gpg
OK http://ftp.at.debian.org jessie Release
OK http://ftp.at.debian.org jessie-updates/main Sources
Holen: 1 http://ftp.at.debian.org jessie-updates/main amd64 Packages/DiffIndex [
6.916 B]
OK http://security.debian.org jessie/updates/main Sources
Holen: 2 http://ftp.at.debian.org jessie-updates/main Translation-en/DiffIndex [
2.704 B]
OK http://ftp.at.debian.org jessie/main Sources
OK http://security.debian.org jessie/updates/main amd64 Packages
OK http://ftp.at.debian.org jessie/main amd64 Packages
OK http://ftp.at.debian.org jessie/main Translation-de
OK http://security.debian.org jessie/updates/main Translation-en
OK http://ftp.at.debian.org jessie/main Translation-en
Ign http://pkg.jenkins.io binary/ InRelease
OK http://pkg.jenkins.io binary/ Release.gpg
OK http://pkg.jenkins.io binary/ Release
OK http://pkg.jenkins.io binary/ Packages
Ign http://pkg.jenkins.io binary/ Translation-de_AT
Ign http://pkg.jenkins.io binary/ Translation-de
Ign http://pkg.jenkins.io binary/ Translation-en
Es wurden 9.620 B in 3 s geholt (3.191 B/s).
Paketlisten werden gelesen... Fertig
root@sew:~# apt-get install python3-pip
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut.
Statusinformationen werden eingelesen.... Fertig
python3-pip ist schon die neueste Version.
0 aktualisiert, 0 neu installiert, 0 zu entfernen und 0 nicht aktualisiert.
root@sew:~# pip3 install django
```

```
Holen: 2 http://ftp.at.debian.org jessie-updates/main Translation-en/DiffIndex [
2.704 B]
OK http://ftp.at.debian.org jessie/main Sources
OK http://security.debian.org jessie/updates/main amd64 Packages
OK http://ftp.at.debian.org jessie/main amd64 Packages
OK http://ftp.at.debian.org jessie/main Translation-de
OK http://security.debian.org jessie/updates/main Translation-en
OK http://ftp.at.debian.org jessie/main Translation-en
Ign http://pkg.jenkins.io binary/ InRelease
OK http://pkg.jenkins.io binary/ Release.gpg
OK http://pkg.jenkins.io binary/ Release
OK http://pkg.jenkins.io binary/ Packages
Ign http://pkg.jenkins.io binary/ Translation-de_AT
Ign http://pkg.jenkins.io binary/ Translation-de
Ign http://pkg.jenkins.io binary/ Translation-en
Es wurden 9.620 B in 3 s geholt (3.191 B/s).
Paketlisten werden gelesen... Fertig
root@sew:~# apt-get install python3-pip
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut.
Statusinformationen werden eingelesen.... Fertig
python3-pip ist schon die neueste Version.
0 aktualisiert, 0 neu installiert, 0 zu entfernen und 0 nicht aktualisiert.
root@sew:~# pip3 install django
Downloading/unpacking django
  Downloading Django-1.10.6-py2.py3-none-any.whl (6.8MB): 6.8MB downloaded
Installing collected packages: django
Successfully installed django
Cleaning up...
root@sew:~#
```

Es wurde überprüft, ob Django installiert wurde

```
root@sew:~# django-admin --version
1.10.6
root@sew:~#
```

3 Konfiguration

3.1 Erstellung eines Projektes

```
root@sew:/home/sew/Dokumente# django-admin startproject A13Django
root@sew:/home/sew/Dokumente# cd A13Django/
root@sew:/home/sew/Dokumente/A13Django# ls
A13Django  manage.py
```

3.2 Ausführen des Django-Servers

```
root@sew:/home/sew/Dokumente/A13Django# python3 manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).

You have 13 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.

March 12, 2017 - 11:25:11
Django version 1.10.6, using settings 'A13Django.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

3.3 Klonen eines Projektes

Nach dem Klonen des Projektes wurde folgender Befehl ausgeführt.

```
root@sew:/home/sew/Dokumente/helloworld# pip install -r requirements.txt
Collecting Django==1.9.4 (from -r requirements.txt (line 3))
  Downloading Django-1.9.4-py2.py3-none-any.whl (6.6MB)
    100% |#####| 6.6MB 128kB/s
Installing collected packages: Django
Successfully installed Django-1.9.4
```

```
root@sew:/home/sew/Dokumente/helloworld# python manage.py migrate
Operations to perform:
  Apply all migrations: admin, contenttypes, sites, auth, sessions
Running migrations:
  Rendering model states... DONE
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying sessions.0001_initial... OK
  Applying sites.0001_initial... OK
  Applying sites.0002_alter_domain_unique... OK
```

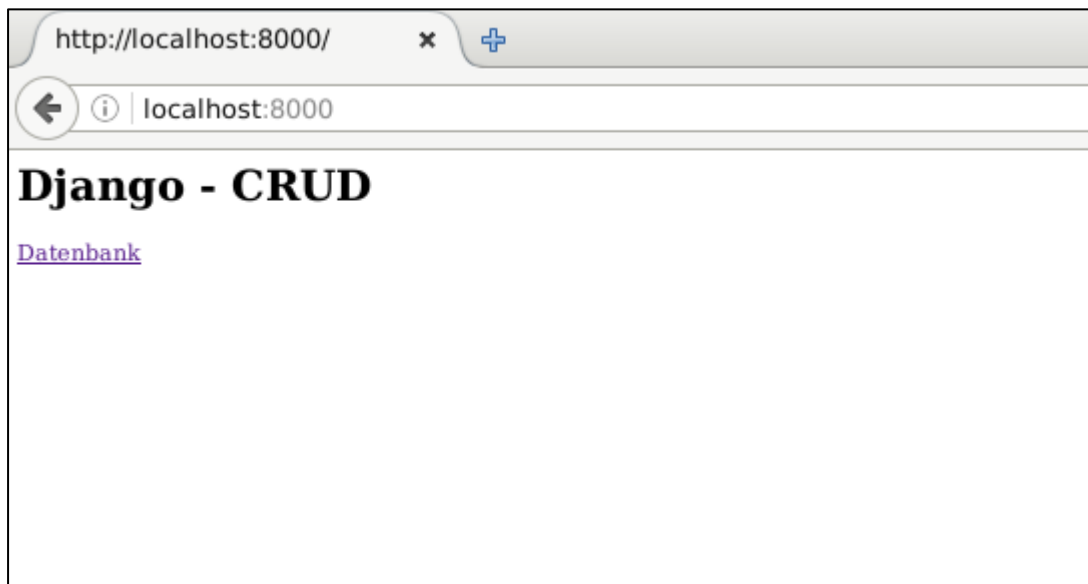
Danach kann der Django-Server ausgeführt werden.

```
root@sew:/home/sew/Dokumente/helloworld# python manage.py runserver
Performing system checks...

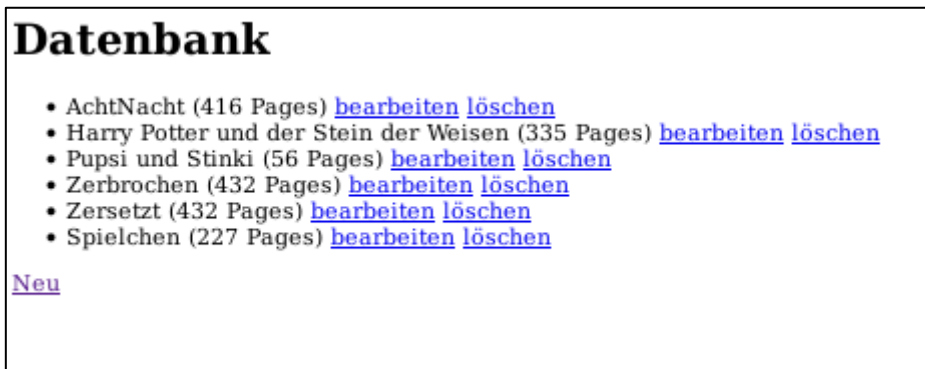
System check identified no issues (0 silenced).
March 12, 2017 - 08:40:21
Django version 1.9.4, using settings 'helloworld.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

4 Beispiel

Startseite



Datenbank



Neuer Eintrag

Name:

Pages:

Bearbeiten

Name:

Pages:

• Sherlock Holmes - Das Tal des Grauens (400 Pages) [bearbeiten](#) [löschen](#)

Löschen

Willst du "Sherlock Holmes - Das Tal des Grauens" wirklich löschen?

5 Models erstellen

Models definieren, wie Daten gespeichert werden. Üblicherweise repräsentiert ein Model eine Tabelle in der Datenbank und besteht aus Feldern, Metadaten und Methoden. Durch diese Informationen kann Django automatisch eine Schnittstelle zur Datenbank generieren, die es erlaubt, objektorientiert auf diese zuzugreifen. Man spricht hierbei auch von Object-Relational-Mapping (ORM).

6 Felder

Felder werden als Klassenattribute definiert und entsprechen den Spalten einer Tabelle. Django bietet verschiedene Typen von Feldern, um die Daten in einem möglichst sinnvollen Format abzulegen. So entspricht beispielsweise CharField einer Spalte vom Typ VARCHAR in einer SQL-Datenbank. Eine Liste der Feldtypen kannst du in der Django-Dokumentation finden.

```
class Tag(models.Model):
    name = models.CharField(max_length=50, unique=True)

class Bookmark(models.Model):
    url = models.URLField()
    title = models.CharField('title', max_length=255)
    description = models.TextField('description', blank=True)
    is_public = models.BooleanField('public', default=True)
    date_created = models.DateTimeField('date created')
    date_updated = models.DateTimeField('date updated')
    owner = models.ForeignKey(User, verbose_name='owner',
                              related_name='bookmarks')
    tags = models.ManyToManyField(Tag, blank=True)
```

7 Metadaten

Models können Metadaten enthalten, welche die Darstellung oder das Verhalten beeinflussen. Sie werden in der internen Klasse Meta definiert. Im Beispiel werden der Name für die Darstellung (singular und plural) sowie die Standardsortierung festgelegt.

```
class Tag(models.Model):
    name = models.CharField(max_length=50, unique=True)
```

```
class Meta:
    verbose_name = 'tag'
    verbose_name_plural = 'tags'
    ordering = ['name']
```

```
class Bookmark(models.Model):
    url = models.URLField()
    title = models.CharField('title', max_length=255)
    description = models.TextField('description', blank=True)
    is_public = models.BooleanField('public', default=True)
```

8 Methoden

You can now add Model functionality Methods for actions that apply to a single record. For instance, it is usual to create a human readable form of the record with the method `__str__`.

```
class Tag(models.Model):
    name = models.CharField(max_length=50, unique=True)

    class Meta:
        verbose_name = 'tag'
        verbose_name_plural = 'tags'
        ordering = ['name']

    def __str__(self):
        return self.name
```

9 Aufwandschätzung

	Geschätzter Aufwand	Tatsächlicher Aufwand
Installation & Konfiguration	2 h	2,5 h
Anwendungsfall	2 h	2 h
Datensätze	1 h	0,5 h
Anlegen/Bearbeiten/Löschen von Datensätzen	4 h	2 h
Protokollieren	1 h	1 h
Summe	10 h	8 h

10 Quellenverzeichnis

[1] drweb.de

<https://www.drweb.de/magazin/framework-django/>

[2] django-workshop.de

<http://www.django-workshop.de/einfuehrung.html>

[3] t3n.de

<http://t3n.de/news/django-python-framework-schnelle-entwicklungsprozesse-335967/>