

GeoDjango

Eine Einführung ins Web-Framework

Stephan Wagner

ITos GmbH, CH-9642 Ebnat-Kappel

FOSSGIS

13. Juni 2013



Preamble

GeoDjango kurz und bündig

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango

Eigenschaften
Installation
API

Würdigung

GeoDjango erweitert Django um:

- OGC-konforme Geometriefelder
- Object Request Broker (ORB) für GeoDaten:
PostgreSQL/PostGIS, SQLite/Spatialite, Oracle
- GDAL-, GEOS-, OGR-, und GeolP-Einbindung (ctypes)
- GeoEditor im Admin-Tool



Preamblel

GeoDjango kurz und bündig

GeoDjango

Preamblel

Django

Architektur
Eigenschaften

GeoDjango

Eigenschaften
Installation
API

Würdigung

GeoDjango erweitert Django um:

- OGC-konforme Geometriefelder
- Object Request Broker (ORB) für GeoDaten:
PostgreSQL/PostGIS, SQLite/SpatiaLite, Oracle
- GDAL-, GEOS-, OGR-, und GeoIP-Einbindung (ctypes)
- GeoEditor im Admin-Tool



Preamble

GeoDjango kurz und bündig

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango

Eigenschaften
Installation
API

Würdigung

GeoDjango erweitert Django um:

- OGC-konforme Geometriefelder
- Object Request Broker (ORB) für GeoDaten:
PostgreSQL/PostGIS, SQLite/SpatiaLite, Oracle
- GDAL-, GEOS-, OGR-, und GeoIP-Einbindung (ctypes)
- GeoEditor im Admin-Tool



Preamble

GeoDjango kurz und bündig

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango

Eigenschaften
Installation
API

Würdigung

GeoDjango erweitert Django um:

- OGC-konforme Geometriefelder
- Object Request Broker (ORB) für GeoDaten:
PostgreSQL/PostGIS, SQLite/SpatiaLite, Oracle
- GDAL-, GEOS-, OGR-, und GeoIP-Einbindung (ctypes)
- GeoEditor im Admin-Tool



Preamble

Django

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango

Eigenschaften
Installation
API

Würdigung

- Freies, quelloffenes Webapplikations-Framework
- Für **Linux**, Solaris, MacOS, MS-Windows
- Modell - View – Controller (MVC) Paradigma
- Geschrieben in *Python*
- Erste Veröffentlichung (BSD-Lizenz) im July 2005
- Heute unter dem Dach der *Django Software Foundation* (DSF)



Preamble

Django

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango

Eigenschaften
Installation
API

Würdigung

- Freies, quelloffenes Webapplikations-Framework
- Für **Linux**, Solaris, MacOS, MS-Windows
- Modell - View – Controller (MVC) Paradigma
- Geschrieben in *Python*
- Erste Veröffentlichung (BSD-Lizenz) im July 2005
- Heute unter dem Dach der *Django Software Foundation* (DSF)



Preamble

Django

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango

Eigenschaften
Installation
API

Würdigung

- Freies, quelloffenes Webapplikations-Framework
- Für **Linux**, Solaris, MacOS, MS-Windows
- Modell - View – Controller (MVC) Paradigma
- Geschrieben in *Python*
- Erste Veröffentlichung (BSD-Lizenz) im July 2005
- Heute unter dem Dach der *Django Software Foundation* (DSF)



Preamble

Django

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango

Eigenschaften
Installation
API

Würdigung

- Freies, quelloffenes Webapplikations-Framework
- Für **Linux**, Solaris, MacOS, MS-Windows
- Modell - View – Controller (MVC) Paradigma
- Geschrieben in *Python*
- Erste Veröffentlichung (BSD-Lizenz) im July 2005
- Heute unter dem Dach der *Django Software Foundation* (DSF)



Preamble

Django

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango

Eigenschaften
Installation
API

Würdigung

- Freies, quelloffenes Webapplikations-Framework
- Für **Linux**, Solaris, MacOS, MS-Windows
- Modell - View – Controller (MVC) Paradigma
- Geschrieben in *Python*
- Erste Veröffentlichung (BSD-Lizenz) im July 2005
- Heute unter dem Dach der *Django Software Foundation* (DSF)



Preamble

Dokumentation

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango

Eigenschaften
Installation
API

Würdigung

- **Django:**
 - Django Einstiegsseite: *djangoproject.org*
 - Django Tutorial in 4 Kapiteln
 - Django Dokumentation
- **GeoDjango:**
 - GeoDjango Tutorial
 - GeoDjango Dokumentation
- Verwendete Version beachten!



Preamble

Dokumentation

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango

Eigenschaften
Installation
API

Würdigung

- **Django:**
 - Django Einstiegsseite: *djangoproject.org*
 - Django Tutorial in 4 Kapiteln
 - Django Dokumentation
- **GeoDjango:**
 - GeoDjango Tutorial
 - GeoDjango Dokumentation
- Verwendete Version beachten!



Preamble

Dokumentation

GeoDjango

Preamble

Django

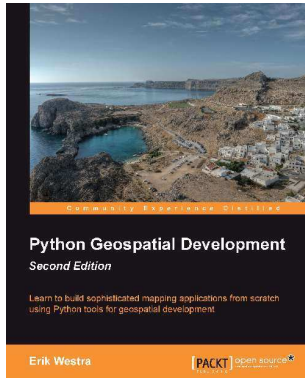
Architektur
Eigenschaften

GeoDjango

Eigenschaften
Installation
API

Würdigung

- **Django:**
 - Django Einstiegsseite: *djangoproject.org*
 - Django Tutorial in 4 Kapiteln
 - Django Dokumentation
- **GeoDjango:**
 - GeoDjango Tutorial
 - GeoDjango Dokumentation
- Verwendete Version beachten!



- 1 **Python Geospatial Development** 2nd edition, Erik Westra, Packt Publishing 2013,
- 2 **Beginning Geodjango**, Stuart Marsh, Apress 2010

- Stephan Wagner
- Geschäftsführer, Mitinhaber *ITos GmbH*
- Web-(GIS)-Lösungen
- PostgreSQL - Django (SQL, Python, HTML, CSS, JS)
- ⇒ System-Architektur und -Administration



- Stephan Wagner
- Geschäftsführer, Mitinhaber *ITos GmbH*
- Web-(GIS)-Lösungen
- PostgreSQL - Django (SQL, Python, HTML, CSS, JS)
- ⇒ System-Architektur und -Administration



- Stephan Wagner
- Geschäftsführer, Mitinhaber *ITos GmbH*
- Web-(GIS)-Lösungen
- PostgreSQL - Django (SQL, Python, HTML, CSS, JS)
- ⇒ System-Architektur und -Administration





Preamble

GeoDjango by ITos

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango

Eigenschaften
Installation
API

Würdigung

- Klassisches Kartenwerk
- Geo-Lokalisation (OSM/Google) → Geodatenanalyse (PostGIS, pandas)
- PostGIS-Raster als Matrizenspeicher (Modellrechnung)



Preamble

GeoDjango by ITos

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango

Eigenschaften
Installation
API

Würdigung

- Klassisches Kartenwerk
- Geo-Lokalisation (OSM/Google) → Geodatenanalyse (PostGIS, pandas)
- PostGIS-Raster als Matrizenspeicher (Modellrechnung)



Preamble

GeoDjango by ITos

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango

Eigenschaften
Installation
API

Würdigung

- Klassisches Kartenwerk
- Geo-Lokalisation (OSM/Google) → Geodatenanalyse (PostGIS, pandas)
- PostGIS-Raster als Matrizenspeicher (Modellrechnung)



Django

Bekannte, djangobasierte Webseiten

GeoDjango

Preamblel

Django

Architektur
Eigenschaften

GeoDjango
Eigenschaften
Installation
API

Würdigung

- lawrence.com (GEO)
- newyorktimes.com (GEO)
- guardian.co.uk
- nationalgeographic.com (GEO)
- mercedesbenz.com (GEO)
- orange.ch
- mozilla.org
- instagram.com



Django

Installation

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango

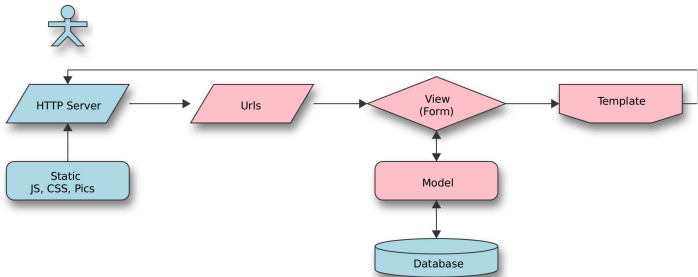
Eigenschaften
Installation
API

Würdigung

- Python *virtualenv*, *pip*
- Entwicklungs-, Test- und Produktiv-Umgebung
- Deploying **Automatisieren!**
- ChangeLog **lesen!**

- ① **M:** Objektrelationaler Mapper
- ② **V:** Template-Engine
- ③ **C:** Restful URL-Drehscheibe, basierend auf regulären Ausdrücken

	MVC	Django
1.	<i>Model</i>	<i>Model</i>
2.	<i>View</i>	<i>Template</i>
3.	<i>Controller</i>	View





Django

Haupt-Features

GeoDjango

Preamblel

Django

Architektur

Eigenschaften

GeoDjango

Eigenschaften

Installation

API

Würdigung

- Eigenständiger http-Server für die Entwicklung
- **Serialisation und Validierung von Formularen (HTML-Form ⇔ Datenbank)**
- Internationalisierung (gettext)
- Lesen/Schreiben von XML- und JSON-Repräsentationen der Modell-Instanzen
- Erweiterbar über Middleware Klassen ⇒



Django

Haupt-Features

GeoDjango

Preamblel

Django

Architektur Eigenschaften

GeoDjango

Eigenschaften Installation API

Würdigung

- Eigenständiger http-Server für die Entwicklung
- **Serialisation und Validierung von Formularen (HTML-Form ⇔ Datenbank)**
- Internationalisierung (gettext)
- Lesen/Schreiben von XML- und JSON-Representationen der Modell-Instanzen
- Erweiterbar über Middleware Klassen ⇒



Django

Haupt-Features

GeoDjango

Preamblel

Django

Architektur Eigenschaften

GeoDjango

Eigenschaften Installation API

Würdigung

- Eigenständiger http-Server für die Entwicklung
- **Serialisation und Validierung von Formularen (HTML-Form ⇔ Datenbank)**
- Internationalisierung (gettext)
- Lesen/Schreiben von XML- und JSON-Representationen der Modell-Instanzen
- Erweiterbar über Middleware Klassen ⇒



Django

Haupt-Features

GeoDjango

Preamblel

Django

Architektur Eigenschaften

GeoDjango

Eigenschaften Installation API

Würdigung

- Eigenständiger http-Server für die Entwicklung
- **Serialisation und Validierung von Formularen (HTML-Form ⇔ Datenbank)**
- Internationalisierung (gettext)
- Lesen/Schreiben von XML- und JSON-Repräsentationen der Modell-Instanzen
- Erweiterbar über Middleware Klassen ⇒



Django

Haupt-Features

GeoDjango

Preamblel

Django

Architektur
Eigenschaften

GeoDjango
Eigenschaften
Installation
API

Würdigung

- Eigenständiger http-Server für die Entwicklung
- **Serialisation und Validierung von Formularen (HTML-Form ⇔ Datenbank)**
- Internationalisierung (gettext)
- Lesen/Schreiben von XML- und JSON-Representationen der Modell-Instanzen
- Erweiterbar über Middleware Klassen ⇒



Django

Integrierte Erweiterungen

GeoDjango

Preamblel

Django

Architektur
Eigenschaften

GeoDjango
Eigenschaften
Installation
API

Würdigung

- Erweiterbares Authentifizierungs-System
- **Aministrations-Interface**
- Werkzeuge um RSS- und Atom-Feeds zu erzeugen
- Flexibles Kommentarsystem
- Site-Framework: eine Django-Installation – mehrere, unabhängige Webseiten
- Mehrere Schutzmechanismen gegen böse Buben (default:on)
- **Framework für GIS-Applikationen:** *GeoDjango*



Django

Integrierte Erweiterungen

GeoDjango

Preamblel

Django

Architektur
Eigenschaften

GeoDjango
Eigenschaften
Installation
API

Würdigung

- Erweiterbares Authentifizierungs-System
- **Aministrations-Interface**
- Werkzeuge um RSS- und Atom-Feeds zu erzeugen
- Flexibles Kommentarsystem
- Site-Framework: eine Django-Installation – mehrere, unabhängige Webseiten
- Mehrere Schutzmechanismen gegen böse Buben (default:on)
- Framework für GIS-Applikationen: *GeoDjango*



Django

Integrierte Erweiterungen

GeoDjango

Preamblel

Django

Architektur
Eigenschaften

GeoDjango
Eigenschaften
Installation
API

Würdigung

- Erweiterbares Authentifizierungs-System
- **Aministrations-Interface**
- Werkzeuge um RSS- und Atom-Feeds zu erzeugen
- Flexibles Kommentarsystem
- Site-Framework: eine Django-Installation – mehrere, unabhängige Webseiten
- Mehrere Schutzmechanismen gegen böse Buben (default:on)
- Framework für GIS-Applikationen: *GeoDjango*



Django

Integrierte Erweiterungen

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango
Eigenschaften
Installation
API

Würdigung

- Erweiterbares Authentifizierungs-System
- **Aministrations-Interface**
- Werkzeuge um RSS- und Atom-Feeds zu erzeugen
- Flexibles Kommentarsystem
- Site-Framework: eine Django-Installation – mehrere, unabhängige Webseiten
- Mehrere Schutzmechanismen gegen böse Buben (default:on)
- Framework für GIS-Applikationen: *GeoDjango*



Django

Integrierte Erweiterungen

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango
Eigenschaften
Installation
API

Würdigung

- Erweiterbares Authentifizierungs-System
- **Aministrations-Interface**
- Werkzeuge um RSS- und Atom-Feeds zu erzeugen
- Flexibles Kommentarsystem
- Site-Framework: eine Django-Installation – mehrere, unabhängige Webseiten
- Mehrere Schutzmechanismen gegen böse Buben (default:on)
- Framework für GIS-Applikationen: *GeoDjango*



Django

Integrierte Erweiterungen

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango
Eigenschaften
Installation
API

Würdigung

- Erweiterbares Authentifizierungs-System
- **Aministrations-Interface**
- Werkzeuge um RSS- und Atom-Feeds zu erzeugen
- Flexibles Kommentarsystem
- Site-Framework: eine Django-Installation – mehrere, unabhängige Webseiten
- Mehrere Schutzmechanismen gegen böse Buben (default:on)
- Framework für GIS-Applikationen: *GeoDjango*



Django

Integrierte Erweiterungen

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango
Eigenschaften
Installation
API

Würdigung

- Erweiterbares Authentifizierungs-System
- **Aministrations-Interface**
- Werkzeuge um RSS- und Atom-Feeds zu erzeugen
- Flexibles Kommentarsystem
- Site-Framework: eine Django-Installation – mehrere, unabhängige Webseiten
- Mehrere Schutzmechanismen gegen böse Buben (default:on)
- **Framework für GIS-Applikationen:** *GeoDjango*



GeoDjango

Bekannte GeoDjango-Webseiten

GeoDjango

Preamblel

Django

Architektur
Eigenschaften

GeoDjango

Eigenschaften
Installation
API

Würdigung

- houstoncrimemaps.com
- gentoo-users.org
- oktravel.com.au
- burningman.com
- backyardpost.com
- Tampa Bay Times
- sosma.org.br
- osfmedicalgroup.org
- UNI Oxford UK



GeoDjango

Eigenschaften

GeoDjango

Preamblel

Django

Architektur
Eigenschaften

GeoDjango
Eigenschaften
Installation
API

Würdigung

GeoDjango erweitert Django um:

- OGC-konforme Geometriefelder
- Object Request Broker (ORB) für GeoDaten:
PostgreSQL/PostGIS, SQLite/SpatiaLite, Oracle
- GDAL-, GEOS-, OGR-, und GeoIP-Einbindung (ctypes)
- GeoEditor im Admin-Tool



GeoDjango

Aktuelle (Geo-)Django-Versionen

GeoDjango

Preamblel

Django

Architektur
Eigenschaften

GeoDjango

Eigenschaften
Installation
API

Würdigung

1 Django:

- Django 1.4 unterstützt Python 2.5, 2.6, 2.7
- **Django 1.5 unterstützt Python 2.6, 2.7; experimentel 3.x**
- Django 1.6 wird Python 2.6?, 2.7, 3.x unterstützen

2 GeoDjango:

- GeoDjango in Django 1.4 mit PostGIS 1.5.x
- GeoDjango in Django 1.5 mit PostGIS 2.x (GDAL \geq 1.5)



1 Django:

- Django 1.4 unterstützt Python 2.5, 2.6, 2.7
- **Django 1.5 unterstützt Python 2.6, 2.7; experimentel 3.x**
- Django 1.6 wird Python 2.6?, 2.7, 3.x unterstützen

2 GeoDjango:

- GeoDjango in Django 1.4 mit PostGIS 1.5.x
- GeoDjango in Django 1.5 mit PostGIS 2.x (GDAL \geq 1.5)

GeoDjango

GeoEditor im Admin-Interface (OpenLayers)

GeoDjango

Date: Today |

Drillsite:



- **Alternative:** QGIS, X2Go, Xfce



GeoDjango

Installation

GeoDjango

Preamblel

Django

Architektur Eigenschaften

GeoDjango Eigenschaften Installation API

Würdigung

- 1 Libraries und Erweiterungen
 - GEOS, Proj.4, PostGIS oder SpatiaLite, (GDAL, GeoIP)

- 2 Django **settings.py**

```
• DATABASES = {'ENGINE':  
                'django.contrib.gis.db.backends.postgis',  
                'NAME': '...',  
                'USER': '...',  
                'PASSWORD': '...',  
                'HOST': '...',  
                'PORT': '...',  
                'OPTIONS': {}},  
• GEOS_LIBRARY_PATH = '/usr/lib/libgeos.c.so',  
• ...  
• INSTALLED_APPS = (  
    'django.contrib.gis',  
    'django.contrib.gis.db.backends.postgis',  
    ...  
)
```

- 3 Django **models.py**

```
• from django.contrib.gis.db import models  
• Geodatenmodel:  
    class Geodatenmodel(models.Model):  
        geom = models.GeometryField(srid=4326,  
                                   geodetic=True)  
        name = models.CharField(max_length=100)  
        objects = models.GeoManager()
```

- 4 Django **view.py**

```
• from django.contrib.gis.geos import *
```



GeoDjango

Installation

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango
Eigenschaften
Installation
API

Würdigung

- 1 Libraries und Erweiterungen
 - GEOS, Proj.4, PostGIS oder SpatiaLite, (GDAL, GeoIP)
- 2 Django **settings.py**
 - `DATABASES = {'ENGINE':`
 - `'django.contrib.gis.db.backends.postgis',`
 - `GEOS_LIBRARY_PATH = '/usr/lib/libgeos_c.so',`
 - `...`
 - `INSTALLED_APPS = (`
 - `'django.contrib.gis',`
 - `'django.contrib.gis.db.backends.postgis',`
- 3 Django **models.py**
 - `from django.contrib.gis.db import models`
 - Geodatenmodel:
 - `class GeoModel(models.Model):`
 - `geometry = models.GeometryField(srid=4326)`
 - `def __str__(self):`
 - `return self.name`
 - `class Meta:`
 - `verbose_name = 'Geodatenmodell'`
- 4 Django **view.py**
 - `from django.contrib.gis.geos import *`



GeoDjango

Installation

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango
Eigenschaften
Installation
API

Würdigung

- 1 Libraries und Erweiterungen
 - GEOS, Proj.4, PostGIS oder SpatiaLite, (GDAL, GeoIP)
- 2 Django **settings.py**
 - `DATABASES = {'ENGINE':`
 - `'django.contrib.gis.db.backends.postgis',`
 - `GEOS_LIBRARY_PATH = '/usr/lib/libgeos_c.so',`
 - `...`
 - `INSTALLED_APPS = (`
 - `'django.contrib.gis',`
 - `'django.contrib.gis.db.backends.postgis',`
- 3 Django **models.py**
 - `from django.contrib.gis.db import models`
 - Geodatenmodel:
 - `mypoints = models.PointField(srid=21781,`
`spatial_index=True)`
 - `objects = models.GeoManager()`
- 4 Django **view.py**
 - `from django.contrib.gis.geos import *`



GeoDjango

Installation

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango
Eigenschaften
Installation
API

Würdigung

- ❶ Libraries und Erweiterungen
 - GEOS, Proj.4, PostGIS oder SpatiaLite, (GDAL, GeolIP)
- ❷ Django **settings.py**
 - `DATABASES = {'ENGINE':`
 - `'django.contrib.gis.db.backends.postgis',`
 - `GEOS_LIBRARY_PATH = '/usr/lib/libgeos_c.so',`
 - `...`
 - `INSTALLED_APPS = (`
 - `'django.contrib.gis',`
 - `'django.contrib.gis.db.backends.postgis',`
- ❸ Django **models.py**
 - `from django.contrib.gis.db import models`
 - Geodatenmodel:
 - `mypoints = models.PointField(srid=21781,`
`spatial_index=True)`
 - `objects = models.GeoManager()`
- ❹ Django **view.py**
 - `from django.contrib.gis.geos import *`



GeoDjango

Meistverwendete Komponenten

GeoDjango

Preamblel

Django

Architektur
Eigenschaften

GeoDjango
Eigenschaften
Installation
API

Würdigung

- **Betriebssystem:** *GNU-Linux:*
- **Webserver:**
 - *NGINX*
 - *WSGI: Gunicorn (Python), uWSGI (C)*
- **Deploying:** *fabric*
- **Cache:** *redis*
- **JS:** *jQuery, OpenLayers (Swisstopo GeoAdmin API)*
- **CSS:** *pyScss (Preprozessor), webassets (merge, compress)*
- **Datenbank:** *PostgreSQL, PostGIS*
- **DB-Daten & -Schema migrieren:** *South (Extension)*



GeoDjango

Meistverwendete Komponenten

GeoDjango

Preamblel

Django

Architektur
Eigenschaften

GeoDjango
Eigenschaften
Installation
API

Würdigung

- **Betriebssystem:** *GNU-Linux:*
- **Webserver:**
 - 1 *NGINX*
 - 2 *WSGI: Gunicorn (Python), uWSGI (C)*
- **Deploying:** *fabric*
- **Cache:** *redis*
- **JS:** *jQuery, OpenLayers (Swisstopo GeoAdmin API)*
- **CSS:** *pyScss (Preprozessor), webassets (merge, compress)*
- **Datenbank:** *PostgreSQL, PostGIS*
- **DB-Daten & -Schema migrieren:** *South (Extension)*



GeoDjango

Meistverwendete Komponenten

GeoDjango

Preamblel

Django

Architektur
Eigenschaften

GeoDjango
Eigenschaften
Installation
API

Würdigung

- **Betriebssystem:** *GNU-Linux:*
- **Webserver:**
 - 1 *NGINX*
 - 2 *WSGI: Gunicorn (Python), uWSGI (C)*
- **Deploying:** *fabric*
- **Cache:** *redis*
- **JS:** *jQuery, OpenLayers (Swisstopo GeoAdmin API)*
- **CSS:** *pyScss (Preprozessor), webassets (merge, compress)*
- **Datenbank:** *PostgreSQL, PostGIS*
- **DB-Daten & -Schema migrieren:** *South (Extension)*



GeoDjango

Meistverwendete Komponenten

GeoDjango

Preamblel

Django

Architektur
Eigenschaften

GeoDjango
Eigenschaften
Installation
API

Würdigung

- **Betriebssystem:** *GNU-Linux:*
- **Webserver:**
 - ① *NGINX*
 - ② *WSGI: Gunicorn (Python), uWSGI (C)*
- **Deploying:** *fabric*
- **Cache:** *redis*
- **JS:** *jQuery, OpenLayers (Swisstopo GeoAdmin API)*
- **CSS:** *pyScss (Preprozessor), webassets (merge, compress)*
- **Datenbank:** *PostgreSQL, PostGIS*
- **DB-Daten & -Schema migrieren:** *South (Extension)*



GeoDjango

Meistverwendete Komponenten

GeoDjango

Preamblel

Django

Architektur
Eigenschaften

GeoDjango
Eigenschaften
Installation
API

Würdigung

- **Betriebssystem:** *GNU-Linux:*
- **Webserver:**
 - ① *NGINX*
 - ② *WSGI: Gunicorn (Python), uWSGI (C)*
- **Deploying:** *fabric*
- **Cache:** *redis*
- **JS:** *jQuery, OpenLayers (Swisstopo GeoAdmin API)*
- **CSS:** *pyScss (Preprozessor), webassets (merge, compress)*
- **Datenbank:** *PostgreSQL, PostGIS*
- **DB-Daten & -Schema migrieren:** *South (Extension)*



GeoDjango

Meistverwendete Komponenten

GeoDjango

Preamblel

Django

Architektur
Eigenschaften

GeoDjango
Eigenschaften
Installation
API

Würdigung

- **Betriebssystem:** *GNU-Linux:*
- **Webserver:**
 - 1 *NGINX*
 - 2 *WSGI: Gunicorn (Python), uWSGI (C)*
- **Deploying:** *fabric*
- **Cache:** *redis*
- **JS:** *jQuery, OpenLayers (Swisstopo GeoAdmin API)*
- **CSS:** *pyScss (Preprozessor), webassets (merge, compress)*
- **Datenbank:** *PostgreSQL, PostGIS*
- **DB-Daten & -Schema migrieren:** *South (Extension)*

- **Betriebssystem:** *GNU-Linux:*
- **Webserver:**
 - 1 *NGINX*
 - 2 *WSGI: Gunicorn (Python), uWSGI (C)*
- **Deploying:** *fabric*
- **Cache:** *redis*
- **JS:** *jQuery, OpenLayers* (Swisstopo GeoAdmin API)
- **CSS:** *pyScss* (Preprozessor), *webassets* (merge, compress)
- **Datenbank:** *PostgreSQL, PostGIS*
- **DB-Daten & -Schema migrieren:** *South (Extension)*



GeoDjango

Meistverwendete Komponenten

GeoDjango

Preamblel

Django

Architektur
Eigenschaften

GeoDjango
Eigenschaften
Installation
API

Würdigung

- **Betriebssystem:** *GNU-Linux:*
- **Webserver:**
 - ① *NGINX*
 - ② *WSGI: Gunicorn (Python), uWSGI (C)*
- **Deploying:** *fabric*
- **Cache:** *redis*
- **JS:** *jQuery, OpenLayers* (Swisstopo GeoAdmin API)
- **CSS:** *pyScss* (Preprozessor), *webassets* (merge, compress)
- **Datenbank:** *PostgreSQL, PostGIS*
- **DB-Daten & -Schema migrieren:** *South (Extension)*



GeoDjango

Application Programming Interface

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango
Eigenschaften
Installation
API

Würdigung

- **GeoDjango Model API:** Geometrie-Felder, -Optionen, -Koordinatensystem, -Index.
- **GeoDjango Database API:** Anfragen an PostGIS, Spatialite, Oracle, (MySQL)
- **GeoDjango Forms API:** Visuelles Anzeigen und Editieren von Geometrien (Default: OpenLayers)
- **GeoQuerySet API:** Abfrage über die Geometrien
- **Measurement API:** Distanzen und Flächen
- **GEOS API:** *Geometry Engine - Open Source*
- **GDAL/OGR API:** *Geospatial Data Abstraction Library*, aktuell ohne Raster
- **GeoIP API:** Georeferenzierung über Netzwerkadresse



API

Object Request Broker (Model)

GeoDjango

Preamblel

Django

Architektur
Eigenschaften

GeoDjango

Eigenschaften
Installation
API

Würdigung

- Django-Modell ~ Schnittmenge der unterstützen Datenbanken wird abgebildet
- Massnahmen:
 - SQLalchemy
 - Middleware
 - raw query: `City.objects.raw('SELECT id, name, asText(point) from city')`
 - VIEW, RULE, FUNCTION, TRIGGER
- PostgreSQL SCHEMA:
 - Nur ein SCHEMA verwenden
 - Extension: *django-appschema*
 - SCHEMA Path: `DATABASES = { ... 'OPTIONS': {'options': '-c search_path=app,postgis'}`



API

Object Request Broker (Model)

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango
Eigenschaften
Installation
API

Würdigung

- Django-Modell ~ Schnittmenge der unterstützen Datenbanken wird abgebildet
- Massnahmen:
 - SQLAlchemy
 - Middleware
 - raw query: `City.objects.raw('SELECT id, name, asText(point) from city')`
 - VIEW, RULE, FUNCTION, TRIGGER
- PostgreSQL SCHEMA:
 - Nur ein SCHEMA verwenden
 - Extension: *django-appschema*
 - SCHEMA Path: *DATABASES = { ... 'OPTIONS': {'options': '-c search_path=app,postgis'}*



API

Object Request Broker (Model)

GeoDjango

Preamblel

Django

Architektur
Eigenschaften

GeoDjango

Eigenschaften
Installation
API

Würdigung

- Django-Modell ~ Schnittmenge der unterstützen Datenbanken wird abgebildet
- Massnahmen:
 - SQLAlchemy
 - Middleware
 - raw query: `City.objects.raw('SELECT id, name, asText(point) from city')`
 - VIEW, RULE, FUNCTION, TRIGGER
- PostgreSQL SCHEMA:
 - Nur ein SCHEMA verwenden
 - Extension: *django-appschema*
 - SCHEMA Path: *DATABASES = { ... 'OPTIONS': {'options': '-c search_path=app,postgis'}*



API

Object Request Broker (Model)

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango

Eigenschaften
Installation
API

Würdigung

- Django-Modell ~ Schnittmenge der unterstützen Datenbanken wird abgebildet
- Massnahmen:
 - SQLAlchemy
 - Middleware
 - raw query: `City.objects.raw('SELECT id, name, asText(point) from city')`
 - VIEW, RULE, FUNCTION, TRIGGER
- PostgreSQL SCHEMA:
 - Nur ein SCHEMA verwenden
 - Extension: *django-appschema*
 - SCHEMA Path: *DATABASES = { ... 'OPTIONS': {'options': '-c search_path=app,postgis'}*



API

Object Request Broker (Model)

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango

Eigenschaften
Installation
API

Würdigung

- Django-Modell ~ Schnittmenge der unterstützen Datenbanken wird abgebildet
- Massnahmen:
 - SQLAlchemy
 - Middleware
 - raw query: `City.objects.raw('SELECT id, name, asText(point) from city')`
 - VIEW, RULE, FUNCTION, TRIGGER
- PostgreSQL SCHEMA:
 - Nur ein SCHEMA verwenden
 - Extension: *django-appschema*
 - SCHEMA Path: `DATABASES = { ... 'OPTIONS': {'options': '-c search_path=app,postgis'}`

- Django-Modell ~ Schnittmenge der unterstützen Datenbanken wird abgebildet
- Massnahmen:
 - SQLAlchemy
 - Middleware
 - raw query: `City.objects.raw('SELECT id, name, asText(point) from city')`
 - VIEW, RULE, FUNCTION, TRIGGER
- PostgreSQL SCHEMA:
 - Nur ein SCHEMA verwenden
 - Extension: *django-appschema*
 - SCHEMA Path: *DATABASES = { ... 'OPTIONS': { 'options': '-c search_path=app,postgis' }*

- Django-Modell ~ Schnittmenge der unterstützen Datenbanken wird abgebildet
- Massnahmen:
 - SQLAlchemy
 - Middleware
 - raw query: `City.objects.raw('SELECT id, name, asText(point) from city')`
 - VIEW, RULE, FUNCTION, TRIGGER
- PostgreSQL SCHEMA:
 - Nur ein SCHEMA verwenden
 - Extension: *django-appschema*
 - SCHEMA Path: *DATABASES = { ... 'OPTIONS': { 'options': '-c search_path=app,postgis' }*

- Django-Modell ~ Schnittmenge der unterstützen Datenbanken wird abgebildet
- Massnahmen:
 - SQLAlchemy
 - Middleware
 - raw query: `City.objects.raw('SELECT id, name, asText(point) from city')`
 - VIEW, RULE, FUNCTION, TRIGGER
- PostgreSQL SCHEMA:
 - Nur ein SCHEMA verwenden
 - Extension: *django-appschema*
 - SCHEMA Path: `DATABASES = { ... 'OPTIONS': { 'options': '-c search_path=app,postgis' }`

- Dokumentation & Tutorial
 - Admin-Interface
 - Django erweitert sich nahtlos um GEO
 - Form
 - Extensions (200 GEO), Library bis Applikation, bsp. django-cms

- Dokumentation & Tutorial
- Admin-Interface
- Django erweitert sich nahtlos um GEO
- Form
- Extensions (200 GEO), Library bis Applikation, bsp. django-cms

- Dokumentation & Tutorial
- Admin-Interface
- Django erweitert sich nahtlos um GEO
- Form
- Extensions (200 GEO), Library bis Applikation, bsp. django-cms

- Dokumentation & Tutorial
- Admin-Interface
- Django erweitert sich nahtlos um GEO
- Form
- Extensions (200 GEO), Library bis Applikation, bsp. django-cms

- Dokumentation & Tutorial
- Admin-Interface
- Django erweitert sich nahtlos um GEO
- Form
- Extensions (200 GEO), Library bis Applikation, bsp. django-cms

Es folgen ...

GeoDjango

Preamble

Django

Architektur
Eigenschaften

GeoDjango

Eigenschaften
Installation
API

Würdigung

- **5.001:** Im- und Export von dxf-Daten mit OGR (Harald Schwenk)
- **3.008:** Effizientes Mappen von Hausnummern (Markus Semm)
- **3.010:** Das SHOGun-WebGIS Framework (Till Adams)