



OPEN SOURCES #14 - GEODJANGO



GEODJANGO: A SYNERGY OF PYTHON, GEOSPATIAL, AND THE WEB

GeoDjango is a set of extensions to Django, the popular open source web content framework, allowing for easy portrayal of geospatial data inside your web portal.

A Geographic Web Framework

GeoDjango grants spatial faculties to Django, a high level open source web framework written in Python. Django was developed out of necessity in the fast paced online-newspaper world where short turn-around times and precise web content are top priorities. Creating HTML content dynamically with a server-side scripting language is not a novel idea but the Django creators needed to be able to create new content types rapidly, and with high standards. "For Perfectionists with Deadlines" is the crafty Django slogan that sums up this powerful flavor of web development. The goal of GeoDjango is to make the development of GIS web applications as quick, flexible, and elegant as possible and has been dubbed the tool of choice for "(Neo) Geographers with Deadlines".

Seamless Python API

The core of GeoDjango is an intuitive Python API to several of today's most important open source geospatial packages for data conversion, manipulation, and storage: GDAL, GEOS, and PostGIS. If you are a veteran of these tools then you'll find GeoDjango to be a great way to harness their power and increase your efficiency by letting GeoDjango abstract away the details. And if you are new to GIS and Python programming, GeoDjango is a smart place to start as the wiki, documentation, and code structure all exhibit best practices and good learning examples for making python your geo-language of choice.

GeoDjango provides a seamless Python API through two major features: a ctypes interface to the rich vector libraries of GEOS and GDAL and an extension of the Django Object Relational Mapper (ORM) to support the spatial functions and operators of PostGIS/PostgreSQL, Oracle,

and MySQL. This architecture allows you to create, query, analyze, and summarize spatial and attribute data on-the-fly using pure Python from anywhere within your Django site or even outside of a web context in standalone scripts.

Your Geodata on MTV

Django uses a MTV design pattern (Model, Template, View) similar in concept to Ruby on Rails. The idea is full separation of code concerns into modules that define your database (Models), display and format your data based on variables and inheritance of common html across pages (Templates), or respond with the proper data queries and forms based on specific web requests (Views). Whether you are designing your first basic Django site or managing a handful in production you will benefit from this approach which encourages modular, clean code and allows teams of multiple programmers and designers to work on discrete functionality at once.

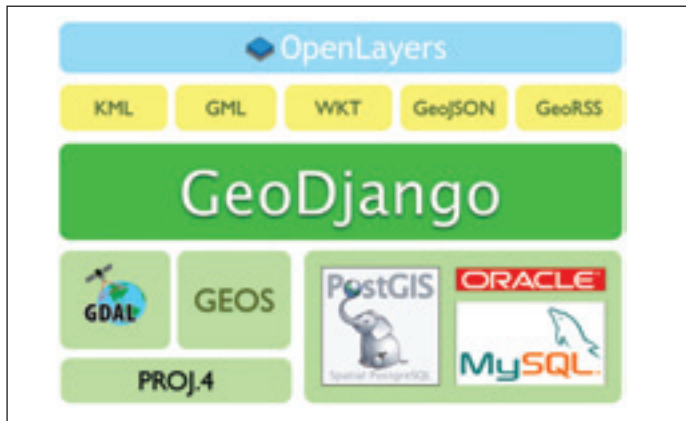
Django Goodness Meets GeoDjango

In addition, for nearly each unique feature which makes Django stand out as a top web framework, GeoDjango adds a thoughtful geo-hook.

The Django models that allow you to define your relational database schema as simple Python classes are extended by GeoDjango enabling all data types described by the OpenGIS Simple Features Specification and custom methods to process spatial data automatically when a record is created or updated.

The Django model API is enhanced by GeoDjango to allow for easy data import into your database from any OGR supported vector format, as well as the automatic generation of Django models through the introspection of all data layers, field types, and features.

The Django template system used to build all HTML content is augmented by GeoDjango with built-in KML generation templates and



Django in the web stack

javascript objects allowing you to easily embed dynamic Google or OpenLayer's maps within your site's pages.

The automatic Django Administration interface that creates an instant, secure web panel to create and edit data is enabled with a customizable OpenLayers vector editing map interface for each spatial data type featuring automatic reprojection, geometry validation, and OpenStreetMap base layers.

The Django Fixtures system works with spatial data allowing for the serialization, backup, and syncing of all database tables across servers and sites using XML, JSON, or YAML formats.

But It Gets Better Still

GeoDjango is a young project and is on the edge of many exciting features that are still in development. Most notable is the addition of a geo-syndication module that will automatically publish your data as GeoRSS and Atom feeds, and a KML sitemap generator to enable the full indexing of your public geodata served from GeoDjango. A mapping framework based on MapServer or Mapnik is also planned and will round out the suite of tools.

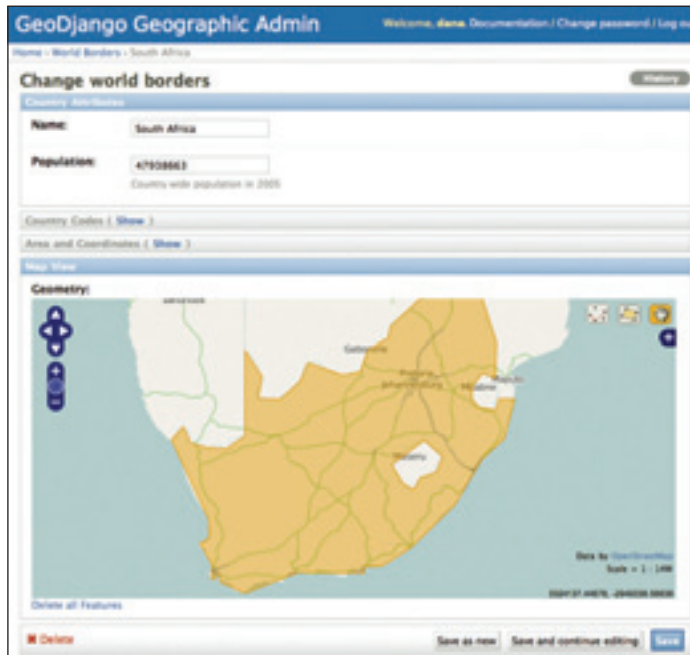
Okay, Enough! How do I try it?

GeoDjango lives as a "contrib" package inside the django code tree and needs no additional installation beyond the installation of Django with the standard python setup.py install command which copies the Django source into your Python site-packages directory. Since Django and all contributed packages are pure Python no compilation is needed. Once you have installed your database of choice with the appropriate driver you will be ready to run through the wealth of great tutorials on the Django site, which will introduce you to creating a new Django "project" folder for your website code, designing a Django "app" to store your application-specific code, and enabling the contributed packages like the Admin panel, Flatpages (a content management system), or the Databrowse (RESTful urls to your data models by record, field name, and shared attributes).

To enable GeoDjango at any time add django.contrib.gis to your Django settings file after installing the required linked libraries (GDAL, GEOS, Proj4). If you are using PostgreSQL you'll need to create a fresh PostGIS-enabled database. Then define your Django models based on sample spatial data you'd like to import (or auto-generate them using the GeoDjango ogrinspect utility). Next you will run the Django syncdb command to automatically create your database schema, sequences, constraints, and indexes and then you will be ready to import your data using the GeoDjango "layermapping" utility. At this point you can fire up the development server on your local machine and immediately be able to add and edit spatial data in the Admin panel, and the possibilities for custom web pages to show off your data will become clearer.

Looking Forward

Frameworks targeted at rapid application development are changing the way both users and developers experience the web. Reusable, modular, integrated, community-driven, and open source, projects like Django, Rails,



GeoDjango Geographic Administration screen



Houston Crime Map created using GeoDjango

and Drupal are fast becoming the best way to design and deploy dynamic sites. Developing sites with Django is already enjoyable and GeoDjango makes sharing and displaying spatial data on the web a true joy. From web mashups utilizing the integrated OpenLayers client and RESTful urls to fast development of complex spatial models and queries, GeoDjango is poised to grow as key toolset in the already rich python/geospatial world. GeoDjango is drawing both seasoned geospatial experts to web development and web wunderkinds to the world of geospatial.

For some examples of what you can do with GeoDjango, check out these sites: www.houstoncrimemaps.com; www.backyardpost.com; code.google.com/p/geodjango-basic-apps/; www.terrადue.com/duetopia.

Links

- Wiki: <http://code.djangoproject.com/wiki/GeoDjango>
- Homepage: <http://geodjango.org>
- Django Project: www.djangoproject.com
- Mailing List: <http://groups.google.com/group/django-users>
- IRC: #geodjango on freenode

Article by Dane Springmeyer, a freelance geospatial application developer; an experienced practitioner of applied modeling, planning, and cartography for the field of conservation science, he enjoys building dynamic websites based upon open source web stacks including OpenLayers, PostGIS, Mapnik, and now, GeoDjango.