Prepared by OpenGeo (Jeffrey Johnson, Ian Schneider and Mike Pumphrey)

Training Syllabus for Advanced GeoNode Software Developers

Morning sessions: 9:00 am - 12:00 pm. 10 min Break at 10:30 am
Afternoon sessions: 1:00 pm - **4:00 pm**. 10 min Break at 2:30 pm

Advanced Developer Training - 4 days
Code Sprint - 2 Days

Level ADVANCED

Audience

- Software developers with at least some background in Geospatial software and web technologies
- Software developers wanting to adopt, extend, and/or integrate GeoNode as part of a spatial data infrastructure

Objectives:

General

- At the end of the training, students will be able to develop and debug mapping and non-mapping web based apps on top of GeoNode.

Specific

1. Students will be able to install GeoNode.
2. Students will be familiar with the software components and APIs that comprise GeoNode
3. Students will be able to setup a project that builds off of a base GeoNode install.
4. Students will be able to customize the theme on a GeoNode instance (colors, logos, etc)
5. Students will be familiarized with the way the request/response cycle works in Django and the different files involved in the process (urls.py, views.py, models.py).
6. Students will know which backend services are running, the ports, how to stop / restart them and where to look for the logs, configuration files and data dirs for: Postgres, GeoServer and Django/Apache/mod_wsgi.
7. Students will be able to add new pages to their GeoNode, like /about or other static pages.
8. Students will be able to create and deploy small django apps, for example a polls app.
9. Students will be able to create a small map based app that makes use of GeoNode's HTTP api endpoints and python based api (Django ORM).
10. Students will be able to interact with GeoNode using the django shell, learn to import models and utility functions to perform administrative tasks.
11. Students will know how to find help from the broader GeoNode community, and familiarized with asking questions on the IRC channel, on the mailing list, the location of the demo and continuous integration servers to try to reproduce bugs and creating issue reports on github.

Combined Group A/B Objectives:

a. Be familiar with GeoNode and other open source tools as a means to build spatial data infrastructure
b. Identify potential problems/needs GeoNode will be capable of solving
c. Identify any issues that prevent GeoNode from being used
d. Define and implement a solution of one of the problems that either:
    1. Extends GeoNode with specific functionality
    2. Integrates GeoNode with existing an application or system
    3. Enhances existing GeoNode functionality

Background Material / Preparation

- **Each participant should come prepared data from their own country or area of interest**
- Basic Web based GIS Concepts
  - OGC Services
    - http://www.opengeospatial.org/
    - http://en.wikipedia.org/wiki/Open_Geospatial_Consortium
  - Web Application Architecture
    - http://en.wikipedia.org/wiki/Web_application
    - http://www.w3.org/2001/tag/2010/05/WebApps.html
    - http://www.amazon.com/Web-Application-Architecture-Principles-Protocols/dp/047051860X
  - AJAX and REST
    - http://en.wikipedia.org/wiki/Ajax_(programming)
    - http://en.wikipedia.org/wiki/Representational_state_transfer
- OpenGeo Suite
  - http://workshops.opengeo.org/suiteintro/
  - http://suite.opengeo.org/opengeo-docs/
- GeoServer Administration
  - http://suite.opengeo.org/opengeo-docs/geoserver/
  - https://docs.google.com/a/opengeo.org/presentation/d/15fvUDYg0TO6WGFQIMLM2J1qiTVBYpfjCp0aQBDT0GrM/edit
  - http://suite.opengeo.org/docs/sysadmin/index.html#sysadmin
- PostgreSQL and PostGIS Administration
  - http://workshops.opengeo.org/postgis-intro/
  - http://workshops.opengeo.org/postgis-spatialdbtips/
- Core development tools and libraries
  - python
    - http://docs.python.org/2/tutorial/
    - http://software-carpentry.org/4_0/python/
    - http://www.learnpython.org/
    - http://learnpythonthehardway.org/book/
  - django
    - https://docs.djangoproject.com/en/dev/intro/tutorial01/
    - https://code.djangoproject.com/wiki/Tutorials
  - javascript
    - http://www.crockford.com/javascript/inheritance.html
    - http://geoext.org/tutorials/quickstart.html
  - jquery
    - http://www.w3schools.com/jquery/default.asp
    - http://docs.jquery.com/Tutorials:Getting_Started_with_jQuery
    - http://www.jquery-tutorial.net/
  - bootstrap
    - http://twitter.github.com/bootstrap/
    - http://www.w3resource.com/twitter-bootstrap/tutorial.php
  - geotools/geoscript/geoserver
    - http://docs.geotools.org/stable/tutorials/feature/csv2shp.html
    - http://geoscript.org/tutorials/index.html
    - http://docs.geotools.org/stable/tutorials/
    - https://github.com/dwins/gsconfig.py/blob/master/README.rst

- geopython
  - http://pycsw.org/docs/documentation.html
  - http://geopython.github.com/OWSLib/
  - https://github.com/sgillies/Fiona
- gdal/ogr
  - http://www.gdal.org/gdal_utilities.html
  - http://www.gdal.org/ogr_utilities.html

Day 1 - Monday 18th February (Joint A/B Session)

- Opening Remarks
- Self Introduction by participants
- Training Overview and Goals and Joint Group A/B Objectives Review

Objectives:

- Install GeoNode
- Become familiar with GeoNode components and APIs
- Learn backend services and how to start/stop, etc.
- Load (some) sample data into GeoNode
- (Potentially) Begin setup of a custom project that builds off of a base GeoNode
- (Advanced) Explore using the django shell

Morning Session

- GeoNode Technical Overview
    - Toolset
        - Django
        - jQuery/Bootstrap
        - GeoServer
        - pycsw
    - Standards
        - OGC Standards
        - Web Standards
    - Architecture
        - GeoNode Architecture
        - Django Architecture
        - GeoServer APIs
- Linux Admin Overview:
    - Basic Shell tools (ssh, sudo, apt etc)
- Begin GeoNode Installation (reference Administrators Workshop)
    - Available GIS Datasets;
    - gisdata repo (San Andres)

Afternoon Session Lab activities

The goal of the afternoon session is to work with a provisioned GeoNode installation as an administrator (using virtualboxes on a mac mini server) and for advanced students to get a GeoNode development environment setup from scratch and work through the manual deploy process. All students should examine the various deploy tools and how to manage the various services on an installation.

- GeoNode Deploy Tools and Management
    - Using a provisioned virtualbox (vagrant)
    - Installing via packages
    - Installing remotely via fabric
    - Controlling basic services
        - apache

        - tomcat
        - postgres
    - Log Inspection and Troubleshooting
        - Debugging on server and client (Optional for very Advanced Students)
    - Advanced Configuration and Tuning (Optional for very Advanced Students)
        - GeoServer PostGIS & Apache
        - Server Monitoring Tools
    - Django Admin
- Manual GeoNode Development Environment Setup (For more advanced Students)
    - Basic tools installation
        - apt-get update and install
        - git
        - python
        - virtualenv tools/wrapper
    - git clone
    - pip install GeoNode
    - paver setup
    - paver start
    - Manual Deployment
        - Apache
        - Tomcat
        - PostgreSQL and PostGIS
        - mod_wsgi
- If time allows, begin setup of custom project
- Additional time after 3:30 pm for individual assistance

Day 2 - Tuesday 19th February

Objectives:

- Complete setup of a project that builds off of a base GeoNode install
- Customize theme of a GeoNode install
- Introduction to request/response cycle: urls.py
- Add new static pages to GeoNode
- (Potentially) Add a dynamic page to GeoNode (just to excercise model/view/template)
- (Advanced) Override/extend existing template
- (Advanced) Explore using the django shell

Morning Session

- Review of Key concepts from Day 1
    - Architecture
    - Standards
    - Deploy and Admin
- Overview of GeoNode Development Prerequisites
    - Server side programming:
        - Basic Shell tools (ssh, sudo, apt etc) review
        - Python
            - virtualenv
            - pip
            - etc (shell, ipython, pdb etc)
        - Django
            - A look at the HTTP request / response cycle (browser request -> urls.py -> views.py -> template.html -> browser rendering)
            - Management commands (createsuperuser, importlayers, shell)
            - Django Admin Web Interface
            - Template tags
- GeoNode Project Layout Review (with goal of learning where to put stuff)
    - Core modules quick intro
        - security
        - layers
        - maps
        - search
        - catalogue
        - geoserver
    - Static resources and how they are resolved

Afternoon Session Lab activities

- Setup new project
- Perform some basic customization
    - CSS
    - Image
- Add static page to GeoNode
    - Extend base template
    - Reference new static resource

- Add dynamic page to GeoNode
    - Simple view
    - Class-based view
    - Parameters via URL
- Override or extend existing template
    - TODO - identify and verify template block for this activity

Day 3 - Wednesday 20th February

Objectives:

- Meet again with Group B to refine problem/solution??
- Create and deploy a small django app: model, view, templates (TODO - define the app)
- Enhance the app using GeoNode HTTP API and Python API (TODO - define the enhancement)
- Introduction to GeoExt and client-side development
- (Advanced) Explore using the django shell

Morning Session:

- Review
- Introduction to app development
  - Creation, Layout, etc.
- Introduction to Django ORM (generally and also cover our example app's model)
- Introduction to GeoNode HTTP API (TODO - beyond search, what?)
- Introduction to GeoNode core models
  - ResourceBase
  - Layer
  - Map
- Simple GeoExt example using WMS from a layer
- Putting it all together - walk through example *inasafe*
  https://github.com/inasafe/safe-GeoNode/blob/master/safe_GeoNode/templates/safe/safe.html
  https://github.com/inasafe/safe-GeoNode/blob/master/safe_GeoNode/static/safe/js/safe.js
  https://github.com/inasafe/safe-GeoNode/blob/master/safe_GeoNode/views.py
  https://github.com/inasafe/safe-GeoNode/blob/master/safe_GeoNode/models.py

Afternoon Session:

- Create the app and hook into custom project
- Enhance the example app
  - Django model/template side of things
  - GeoExt additions

Day 4 - Thursday 21st February

Objectives:

- Introduction to the GeoNode community and resources
- Debugging techniques on client and server
- Learning how to find out what went wrong when things break
- Wildcard! (something the class wants to dig deeper into)

Morning Session

- GeoNode Community
    - Resources
        - IRC, lists, demo, integration servers, github
    - Processes
        - Testing, Roadmap, GNIP
    - Creating (quality) issues
        - Identify the problem and try to reproduce
        - Discuss on IRC or mailing list
        - Create issue (and continue commenting)
- Debugging
    - Client
        - Techniques
        - Firebug
            - Firefox
- Wildcard
    - Client libraries?
    - Developer setup?
    - Other ideas accepted!

Afternoon Session Lab activities

It is expected that at this point, some participants will still need to spend time catching up on previous days afternoon exercises and some will be ready to explore advanced debugging and deployment techniques or continue working on their downstream project. As such it the exercises are much less structured than previous days. Potential activities:

- Explore Advanced development tools and techniques
    - Python
    - Javascript
- Testing
    - Unit tests for new project
    - For bug reproduction
- Additional time after 3:30 pm for individual assistance;

Day 5 - Friday 22nd February

Meet again with Group B to refine problem/solution??

During the 2 days of the code sprint, the participants will apply the techniques learned in the first 4 days and practiced in the exercises to continue working on their own downstream GeoNode based projects with the goal of getting to a minimum viable product that they are excited to show off to the rest of the class as well as to the wider world.

They will be free to work in groups or alone to accomplish their own goals based on their own requirements and expertise. It is hoped that the more advanced students will be able to help the ones that need it and that the whole group can build some camaraderie in support of the goal of eventually federating their projects together in support of the shared regional goals. Several key GeoNode developers (Jeff, Ariel, TomK? others?) will be participating remotely (via IRC and Google+ Hangout) to assist and provide guidance.

Day 6 - Saturday 23rd February

Continue work on downstream projects with help from instructor, more advanced participants, and remote participants (Jeff Ariel et al)

Each individual project (single participant or group) should produce a screencast (using http://www.screenr.com/) to present their project at the end of the code sprint. These should be presented to the group on this final day, and will be featured on the main GeoNode blog as well as on the Collab 4 Dev site. Screenr is limited to a 5 minute presentation which will help to focus on the most important and distinguishing features of each project so that these can be evaluated by the group and by the wider world.

The proposed categories:

- Best theming and customization
- Best use of external module or other integration
- Best core GeoNode contribution

Annex 1: High level conversation with Ian:

[11:26] <ingenieroariel> I think we can meet while the other meeting is happening
[11:27] <ischneider> sure, it's going somewhat slowly
[11:28] <ingenieroariel> yep, glad it is chat
[11:28] <ingenieroariel> so, I added an objectives section on top of the document
[11:28] <ingenieroariel> please take a quick look and let me know if it seems doable in a week
[11:29] <ischneider> looking
[11:30] <ischneider> definitely doable
[11:31] <ischneider> inline with what seem to be 4 potential tasks for the dev/user projects - 1) theming, 2) new pages using GeoNode models, 3) new app/pages, 4) 2 or 3 mixed with some geoext
[11:32] <ingenieroariel> correct
[11:32] <ingenieroariel> so, I have been getting push by other people in my team
[11:32] <ingenieroariel> to have them replicate the app here:
[11:32] <ingenieroariel> http://203.77.224.78/safe/
[11:32] <ingenieroariel> because it involves code from inasafe (our risk calculator)
[11:33] <ingenieroariel> it's a simple django app:
[11:33] <ingenieroariel> urls: https://github.com/inasafe/safe-GeoNode/blob/master/safe_GeoNode/urls.py
[11:34] <ingenieroariel> with just 3 models:
https://github.com/inasafe/safe-GeoNode/blob/master/safe_GeoNode/models.py
[11:34] <ischneider> nice
[11:34] <ingenieroariel> one template:
https://github.com/inasafe/safe-GeoNode/blob/master/safe_GeoNode/templates/safe/safe.html
[11:34] <ingenieroariel> and one js and css file:
https://github.com/inasafe/safe-GeoNode/blob/master/safe_GeoNode/static/safe/js/safe.js
[11:34] <ingenieroariel> very flat, functional style (not a lot of OO)
[11:35] <ischneider> depending upon the dev's intended goal, maybe even possible to stub out some of the implementation with a mock
[11:35] <ingenieroariel> and small customization to bootstrap classes:
https://github.com/inasafe/safe-GeoNode/blob/master/safe_GeoNode/static/safe/css/safe.css
[11:35] <ischneider> a great example to start work with
[11:35] <ingenieroariel> yeah, I think it is okay if we give them something that starts 50% of the way there
[11:35] <ingenieroariel> and we guide them through the completion
[11:36] <ingenieroariel> I put on purpose the polls app in the previous objective
[11:36] <ingenieroariel> since there is already a django tutorial for that
[11:36] <ingenieroariel> and by the time they come to this one they will know about the structure
[11:36] <ingenieroariel> now, what I have not done is thought through how this can be achieved on each day
[11:37] <ischneider> yes on the first 5 of the last sentences :)
[11:38] <ischneider> for the agenda, it seems a big goal is to have an agenda that includes the user's group
[11:38] <ischneider> (and step in if you know more than me)
[11:38] <ischneider> my thought was to have everyone meet early to discuss the problems they need to solve
[11:39] <ischneider> then start learning about GeoNode and iterate at least one more time, preferably 2 during the week
[11:39] <ischneider> to refine the needs and solution
[11:41] <ingenieroariel> that sounds good
[11:41] <ingenieroariel> we do want them to go home feeling they designed a solution to an existing problem
[11:41] <ingenieroariel> contingency planning is a good example
[11:42] <ingenieroariel> what if there was a tsunami here? how many buildings would have to be closed? are

there schools there?
[11:42] <ischneider> having a fall-back to learn from would be good if a specific problem seems to much to tackle in the week
[11:48] <ingenieroariel> right, it is really hard to get them to specify a solvable problem
[11:49] <ingenieroariel> when I have tried they always went back to 'things I would like to do if I had this data that I do not have'
[11:49] <ischneider> ha
[11:49] <ingenieroariel> instead of 'things I would like to do with the data I have available'
[11:49] <ingenieroariel> so they ask for things like LIDAR
[11:49] <ingenieroariel> image tracing
[11:49] <ingenieroariel> alerts via SMS
[11:49] <ischneider> it's true that so often for geospatial that things quickly devolve into a data gathering, QA/QC exercise
[11:49] <ingenieroariel> yep, curation of crowsourced data
[11:50] <ingenieroariel> so, I think it is more effective if we just select the problem and let them apply it to their local island
[11:50] <ischneider> not having data/process is a good reason to mock something up
[11:50] <ingenieroariel> i.e. we invent a big hurricane coming to all islands
[11:50] <ingenieroariel> and an expected flood map
[11:51] <ingenieroariel> we start by uploading that dataset to GeoNode, getting OSM data (adding data to OSM if it does not exist)
[11:51] <ingenieroariel> about building
[11:51] <ingenieroariel> s
[11:51] <ingenieroariel> getting a shapefile back from the hot export tool, or by using qgis
[11:51] <ingenieroariel> and uploading it to GeoNode
[11:51] <ingenieroariel> all that can be done by users
[11:51] <ingenieroariel> and qgis and osm are in scope
[11:51] <ingenieroariel> we can just tell bishwa we need that and they'll take care of it
[11:51] <ischneider> i think i'm understanding - without requiring live hurricane data or the real-time service/processes
[11:52] <ingenieroariel> right
[11:52] <ingenieroariel> so, then our developers
[11:52] <ingenieroariel> will use that data in the GeoNode
[11:52] <ingenieroariel> to create a contingency planning app
[11:52] <ischneider> that's a good point, needing qgis on the client machines in the lab
[11:53] * ingenieroariel thinks we are getting somewhere
[11:53] <ischneider> agreed
[11:55] <ingenieroariel> okay, so how do we move forward? do I let you edit the document to reflect what we talked about (since you will be the one leading the training)
[11:55] <ingenieroariel> or would you rather have me work a bit more on an actual agenda?
[11:56] <ischneider> both can happen in parallel?
[11:57] <ingenieroariel> :)

Annex 2: Stuff that was removed or broken up:

[Note from Ian - I just cut and pasted this stuff out from various sections]

- GeoNode Core Python Dependencies Review
  - owslib

        - pycsw
        - gsconfig
- GeoNode Core Python Module Review
    - Layers
    - Maps
    - Catalogue
    - Search
    - Security
    - geoserver
- GeoNode Contrib Python Modules Review
    - Documents
    - Groups
    - Social
    - Printing
    - Portals
- GeoServer Integration Review
    - GeoServer Catalog and Data Directory Overview
    - gsconfig
    - GeoNode geoserver module (Java and Python Modules)
    - Use of OGC Services in GeoExplorer and Site
- GeoNode Javascript Module Review
    - Dependencies (look at bower component.json)
    - GeoExplorer
        - OpenLayers, ExtJS, GeoExt, GXP, Suite SDK
    - Core Modules
        - core
        - search
        - upload
        - portals
- GeoNode Templates Review
    - Base Templates
    - Core Modules
    - Contrib Modules
- GeoNode's APIs
    - OGS OWS (WMS, WFS, WCS, CSW, TMS, KML, GSR etc)
    - GeoServer REST API
    - GeoServer Import and Print API
    - GeoNode Ad Hoc API
    -

Afternoon Session Lab activities

The goal of this session is to continue to both continue to work with students who may need help with the previous day's exercises, but begin to explore more advanced techniques for those that are ready. The more advanced students in the class will go over how to fix a bug or add a new feature to GeoNode via the Pull Request process and will be free to move on to the exercises scheduled for the following (3rd or 4th) days if they choose.

- Walk through github pull request for either a real issue in github or new feature as identified by class (For more advanced students)

        - Identify goal
        - Clone instructor repo
        - Branch
        - Complete fix/feature
        - Test!
        - Pull Request
- Review prior topics and revisit areas of concern/doubt (For less advanced other students)
- Potentially use API to experiment and learn more (Time and skill permitting)
- Advanced students can start on Day 3 Exercises

- Additional time after 3:30 pm for individual assistance;

Morning Session

- Review of Existing Downstream Projects
    - Worldmap
    - MapStory
    - Risiko/SAFE
    - Others
- Downstream Project Setup Overview
    - django-startproject using a template
    - GeoNode's Templates
        - base
        - social
    - Version Control (github)
    - Deploying your Project
        - Amazon, OpenShift and Heroku (For very Advanced Developers)
    - Staying in Synch with Mainline GeoNode
- Avenues for Customizing and extending with a downstream project
    - CSS (bootswatch)
    - Site Javascript and static media
    - Templates
    - Adding additional django modules
    - Custom GeoExplorer
        - Suite SDK
        - Deploying
- Avenues for Integration with External Services (For very Advanced Developers)
    - External OGC Services (owslib and pycsw)
    - Google Earth
    - ArcGIS
    - qGIS
    - MapBox
    - OpenStreetMap
    - CartoDB
    - Wordpress
    - Drupal

Afternoon Session Lab activities

The goal of the afternoon session is to work with a provisioned GeoNode installation to setup a downstream GeoNode project, put it under version control on github, stay in sync with mainline GeoNode and deploy it to the local install and investigate deploying to other infrastructure. The students will use this project to customize their GeoNode, extend from it and and integrate with external services based on their own requirements and expertise.

Students who are still exploring basic development techniques can do via the simple method and focus on theming, branding and data import while more advanced students can explore all the the various options for customization. Its hoped that the more advanced can also begin to pair up with the less advanced and help them through some of the setup and with their questions and issues. **Every** student should participate in setting up a downstream project whether it is individually or as part of a group.

- Setup a Downstream Project
    - startproject (with templates?)
    - commit to github
    - deploy to local instance
    - deploy to remote instance (for very advanced Developers)
- Use the downstream project to customize, extend and integrate with external services
    - Theme and Branding Customization
    - Base Templates
    - Custom Static Pages
    - Add blog or static pages module
    - Integrate with Mezzanine CMS (for very advanced Developers)
    - Integrate with other Django Modules
    - Setup a Custom Suite SDK App (for very advanced Developers)
        - Debugging
    - Add Custom site javascript (for very advanced developers)
        - Debugging
    - Integrate with External Services (for very advanced Developers)
- Additional time after 3:30 pm for individual assistance;

- GeoNode's Build, Testing and Packaging Infrastructure
    - Continuous Integration
        - Jenkins
        - Travis CI
    - unit tests
    - integration tests
    - javascript tests
    - packaging
    - Automated Deploys
    - geohubot :)
    -

- Development Process
    - Discuss and author a feature request in GitHub leading toward setting up a GNIP with a core contributor.
- Explore Continuous Integration Infrastructure
- Interact with geohubot in the channel