Tutorial: GeoServer Scripting with Python and RESTConfig

MODULE 1: Installing GeoServer

OpenGeo
www.opengeo.org

GeoServer Configuration

- How is it organized?
  - Hierarchy!
  - Workspace -> Store -> Resource
  - Layer <-> Resource
  - Layer -> (many) Styles
  - LayerGroup -> Layers + Styles
  - Store == DataStore or CoverageStore
  - Resource == Coverage or FeatureType

GeoServer Configuration

- What are the parts?
  - **Workspaces:** published namespace, organization
  - **Stores:** connection parameters, db pooling, etc.
  - **Resources:** individual datasets
  - **Layer:** styles, service details
  - **Style:** just an SLD file

REST API

- Uses that hierarchy!
- GET in XML, JSON
- POST in the same formats

Quick n' Dirty

- HTTP GET
- Edit
- HTTP POST (don't forget content-type!)

Why gsconfig.py, then?

- Edit multiple items
- Inspect the content programmatically

Installation

- For basic use:

```
pip install
  https://github.com/dwins/tarball/master
```

- For hacking:

```
git clone
  git://github.com/dwins/gsconfig.py
cd gsconfig.py
python setup.py develop
```

OpenGeo
www.opengeo.org

Examples!

- Querying
  - All layers using a particular style
  - All attributes used in a layer's styles?
  - Do all resources in a workspace use the same projection?

Examples:

- Writing:
  - Import a Shapefile into a PostGIS database
  - Avoid reusing styles between layers by copying
  - Ensure all layers in a workspace advertise the same native bounding box


LEDGE
WEST
TILLIES
RIDG


LEDGE
WEST
TILLIES
RIDG

OpenGeo

www.opengeo.org

OpenGeo
www.opengeo.org