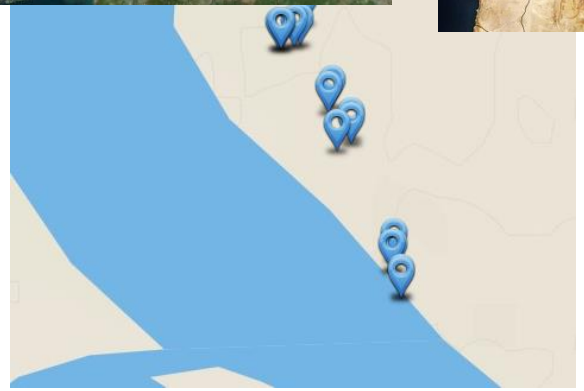
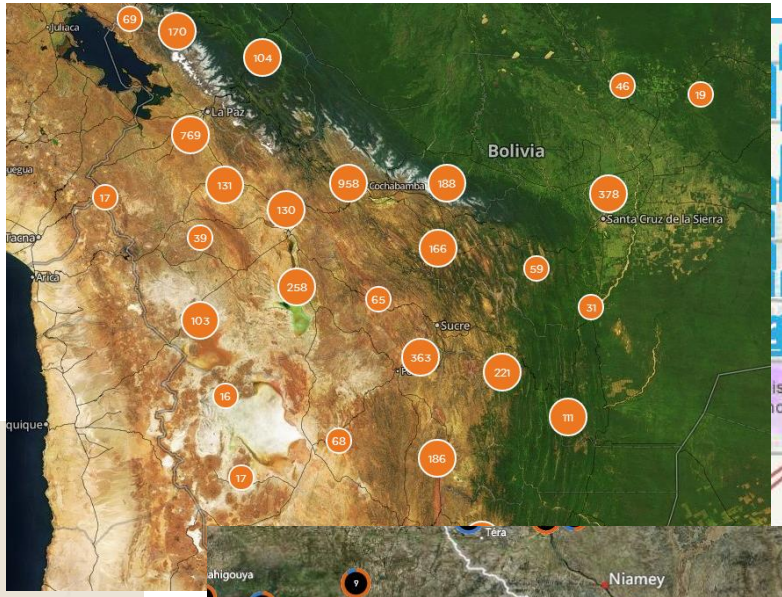
Replacing ArcGIS Server with Node.js and PostGIS

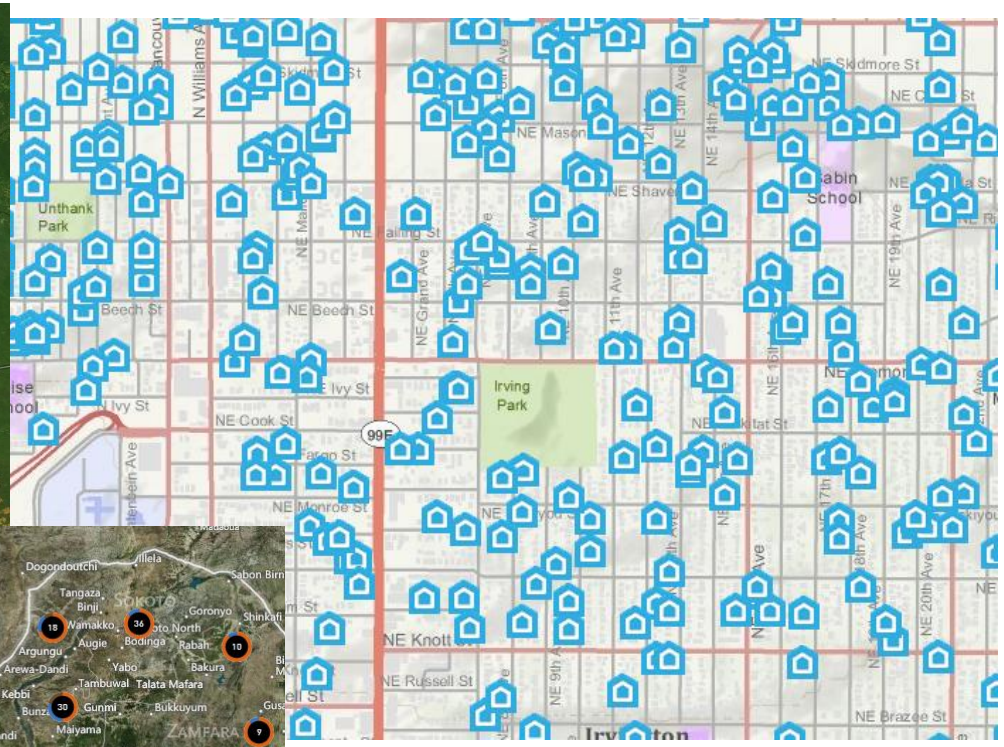
Ryan Whitley

SpatialDev℠

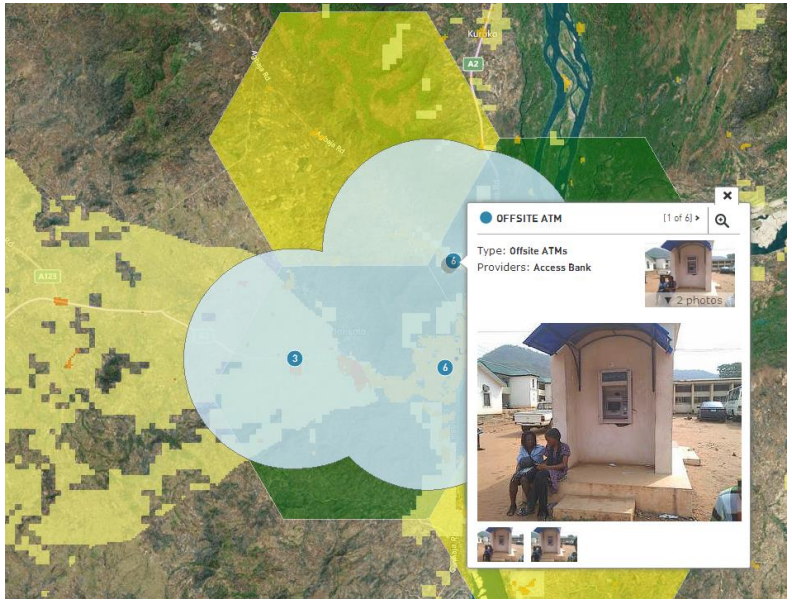
Web app development - Points on Maps

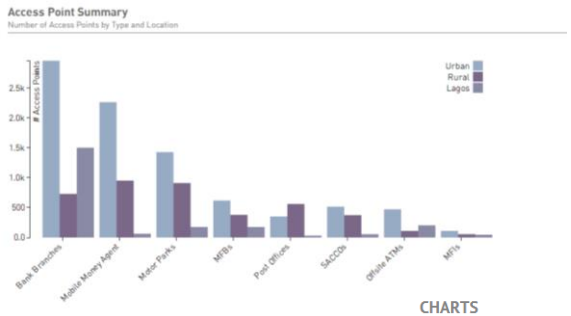


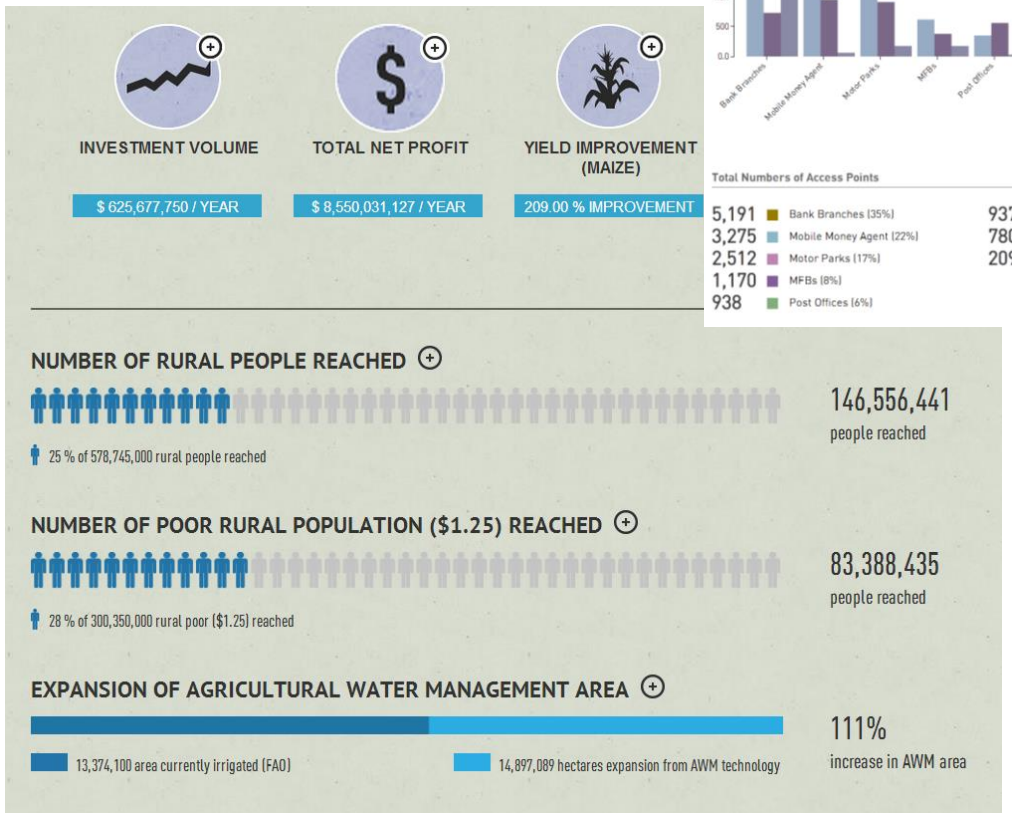




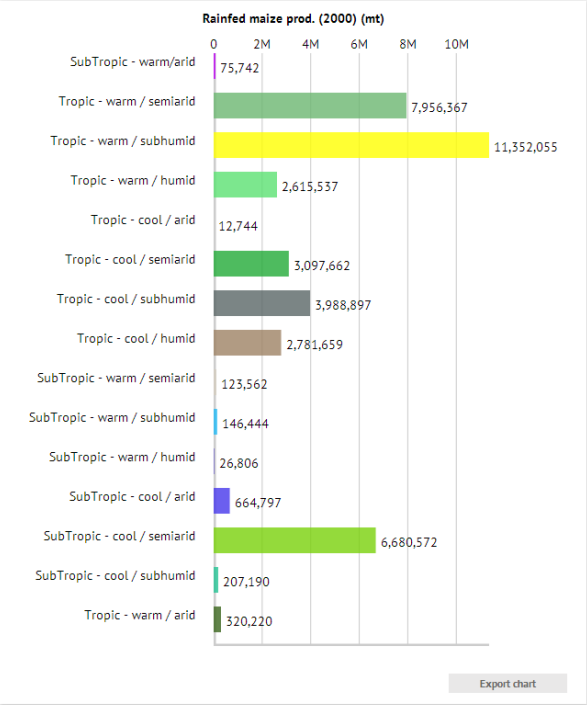
OK – maybe a little more…

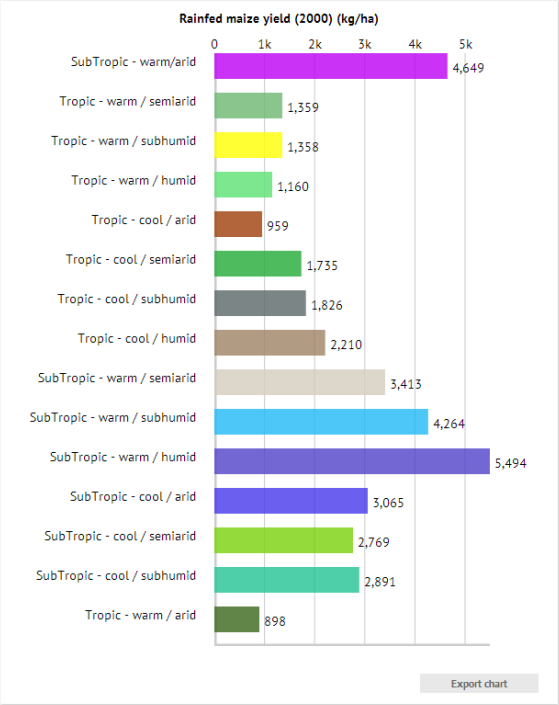






CHARTS





What features of ArcGIS Server are we using most?

- Querying the REST API (30%)
- Dynamic Map Tiles (30 %)
- Geoprocessing Services (15%)
- Hair-Pulling, inexplicable Errors (“Error: 999999”) (15%)
- Static Map Caching (7 %)
- Feature Editing (2 %)
- Printing (1%)

ArcGIS REST Services Directory

Home > services > Bangladesh_MapService_Dev (MapServer)

JSON | SOAP

Bangladesh_MapService_Dev (MapServer)

View In: ArcGIS JavaScript ArcGIS.com Map Google Earth ArcMap ArcGIS Explorer

View Footprint In: ArcGIS.com Map

Service Description: Bangladesh Map Service

Map Name: Layers

Legend

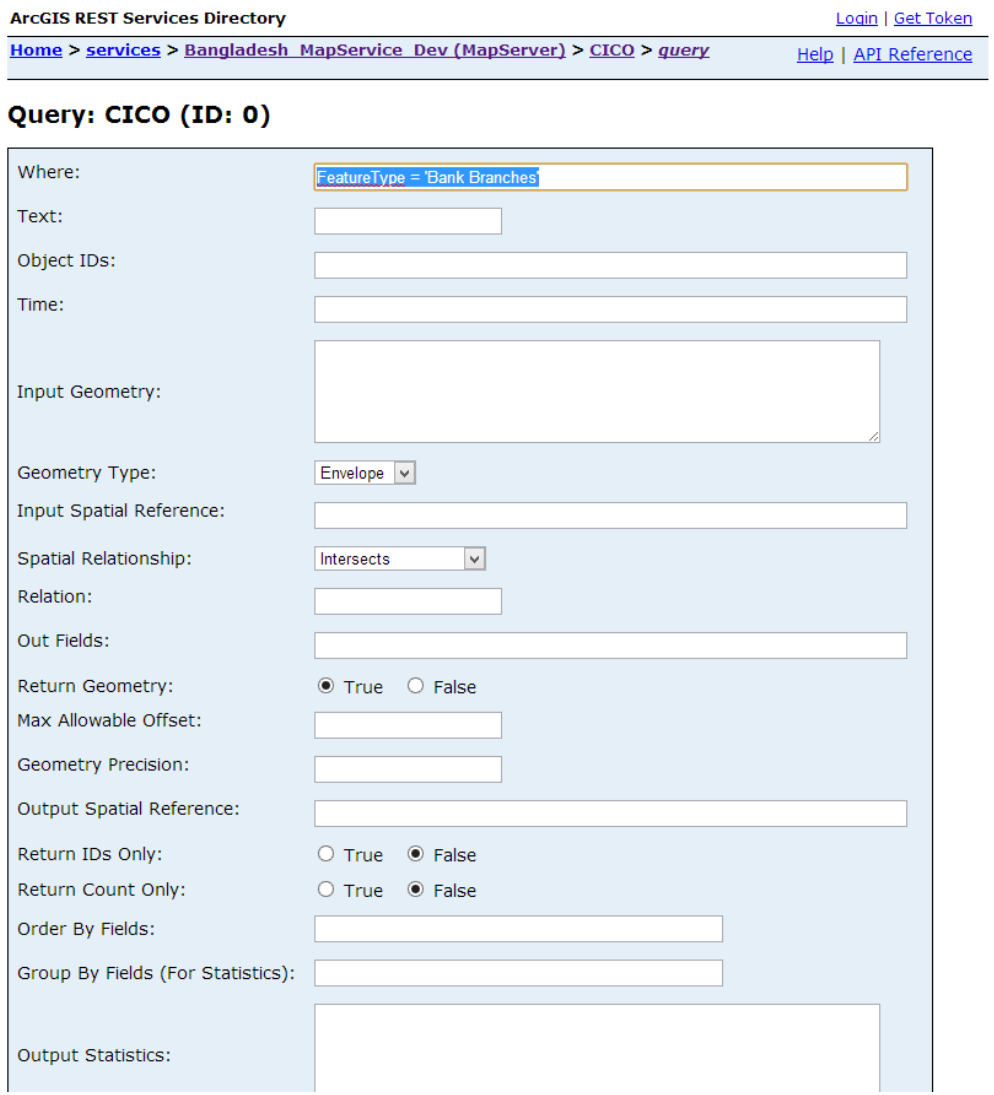
All Layers and Tables

Layers:

- CICO (0)
- Financial Access Points (1)
  - Offsite ATMs (2)
  - Bank Branches (3)
  - MFIs (4)
  - SACCOs (5)
  - Mobile Money Agent (6)
  - Post Offices (7)
- Population Classes (8)
  - Pop_1_10 (9)
  - Pop_11_50 (10)
  - Pop_51_100 (11)
  - Pop_101_500 (12)
  - Pop_501_1000 (13)
- Cell Coverage (14)
- Urban Areas (15)
- Districts (16)
- Reservoirs (17)

Querying the REST API

- Pass in ‘where’ clauses
- Spatial intersects
- Specify output fields
- Return Geometry (or not)
- Reproject coordinates
- Summary statistics



Maybe we can do that with Node.js and PostGIS??

Getting Started: Ingredients

- Computer (Windows, Ubuntu 12 and 13, Mac)
- Node.js
- PostGreSQL 9.2+ and PostGIS 2.x
- Some data

- P.S. - This project just kind of materialized. I mostly wanted to play around with node, and this just sort of popped out.

List tables and views from PostGIS

- Started with a Node.js “Hello World”
- Next - seeing if I can get a list of PostGres tables and views to be spit out
- Build a dynamic REST API using tables and views as the primary building block

PGRestAPI

Home > Services > Table List

Table Listing

Name

bangladesh_cicos
bangladesh_coverage
bangladesh_urbanareas
nigeria_cicos
nigeria_coverage
nigeria_hexbins
nigeria_statecapitals
nigeria_states
nigeria_surveycoverage
nigeria_urbanareas
tanzania_cicos
tanzania_coverage
tanzania_distritcs
tanzania_hexbins
tanzania_urbanareas
uganda_cicos
uganda_coverage
uganda_districts
uganda_hexbins
uganda_popcarto
uganda_population_raster
uganda_poverty2dollarcarto
uganda_poverty_raster
uganda_states
uganda_urbanareas
vw_bangladesh_cicos
vw_nigeria_cicos
vw_tanzania_cicos
vw_uganda_cicos

Running on node with Express, Jade and Less

Table Details

- All tables have Query operations
- If spatial, then Dynamic Map Service and TopoJSON operations
- If raster, then raster operations (such as Zonal Statistics)

PGRestAPI

Home > Services > Table List > uganda_hexbins

Below is a list of columns and column data types for this table

Columns

gid (integer)
longitude (numeric)
latitude (numeric)
hexid (character varying)
cicocount (smallint)
popsum (numeric)
cicoscapit (numeric)
shape_leng (numeric)
shape_area (numeric)
geom (geometry)

SRID is 4326

Operations

Query Dynamic Map Service TopoJSON

Running on node with Express, Jade and Less

Table Querying

- Pass in ‘where’ clauses
- Spatial intersects
- Specify output fields
- Return Geometry (or not)
- Reproject coordinates
- Summary statistics

- Get Feature Envelopes!
- Get GeoJSON or esriJSON

PGRestAPI

Home > Services > Table List > uganda_hexbins > Query

Enter a where clause to retrieve a subset of features from this table

Where	where
Return Fields	field1,field2
Format	html
Return geometry	Yes ◉ No ○
Return feature envelope(s)	Yes ○ No ◉
Group By Fields	field2,field3
Output SRID	3857
Statistics definition	count:field1,sum:field2
Well Known Text (WKT) Geometry	POLYGON((85.341 25.811,85.523 25.838,85.665 25.719,85.479 25.598,85.278 25.687,85.341 25.811))
Feature limit	1000

Submit

Results

Running on node with Express, Jade and Less

- Querying the REST API ✅
- Dynamic Map Tiles
- Geoprocessing Services
- Static Map Caching
- Feature Editing
- Printing

Dynamic Map Tiles

- Nodetiles-core
- Nodetiles-postgis
- Dynamic Dynamic Map Service
- Uses CartoCSS

- Can be served based on PostGIS table, Shapefile, GeoJSON file or GeoJSON object

PGRestAPI

Home > Services > Table List > uganda_hexbins > Dynamic Map Service

Dynamic Map Service

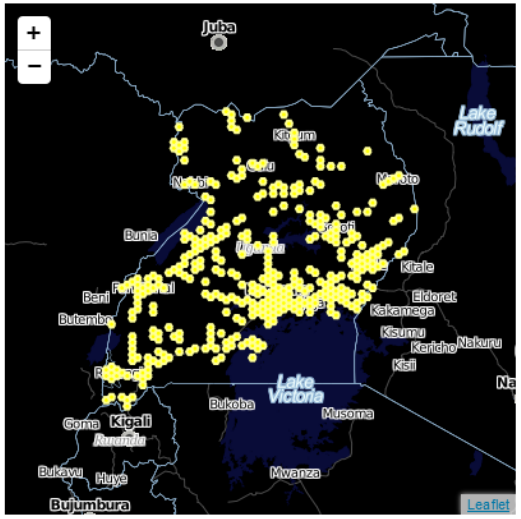
Leaflet Example

```
//Create a leaflet map
var map = L.map('map');

//Add the tiled layer
L.tileLayer('http://54.212.251.211/services/tables/uganda_hexbins/dynamicMap/{z}/{x}/{y}.png').addTo(map);
```

Get more help with Leaflet

Map Preview



Running on node with Express, Jade and Less

- Querying the REST API ✅
- Dynamic Map Tiles ✅
- Geoprocessing Services
- Static Map Caching
- Feature Editing
- Printing

Geoprocessing framework

- Add your own PostGIS Logic into a javascript file
- Specify inputs
- Drop file in the GP Folder
- Get a dynamic REST endpoint that will execute your logic

- * soon will be using Nodetiles to render in-memory GeoJSON to tiles (so you can have images returned from Geoprocessing operations)

- Querying the REST API ✅
- Dynamic Map Tiles ✅
- Geoprocessing Services ✅
- Static Map Caching
- Feature Editing
- Printing

Cached Map Tiles

- Uses a TileStream server
- Use TileMill to connect to PostGIS (or your own local data)
- Style
- Export MBTiles
- Copy to TileStream server
- Get a cached tile server! - http://54.212.254.185:8888

- Querying the REST API ✓
- Dynamic Map Tiles ✓
- Geoprocessing Services ✓
- Static Map Caching ✓
- Feature Editing
- Printing

We’re on a roll!

Feature Editing

- Not yet.

- Querying the REST API ✅
- Dynamic Map Tiles ✅
- Geoprocessing Services ✅
- Static Map Caching ✅
- Feature Editing ❌
- Printing

Printing (split into new project – phantasm)

- Show it (http://services.spatialdev.com/print)
- Print whatever user is looking at (if it's in the DOM)
- Create your own print layouts in HTML

- Querying the REST API ✅
- Dynamic Map Tiles ✅
- Geoprocessing Services ✅
- Static Map Caching ✅
- Feature Editing ❌
- Printing ✅

Create TopoJSON files

- Uses TopoJSON module
- Hits its own Table Query endpoint and gets GeoJSON
- Converts that to TopoJSON file

PGRestAPI

Home > Services > Table List > uganda_hexbins > TopoJSON

TopoJSON files for this table

Name

topo_uganda_hexbins.json

Make a TopoJSON file for this entire dataset

TopoJSON File name uganda_hexbins

Submit

Running on node with Express, Jade and Less

- Querying the REST API ✓
- Dynamic Map Tiles ✓
- Geoprocessing Services ✓
- Static Map Caching ✓
- Feature Editing ✗
- Printing ✓

For custom views of data – use PostGres views.

- For real.
- In practice, I've been using views to do complex joins and things like aggregations that can then be served as a spatial or non-spatial view

Recap: How'd we do?

- Querying the REST API ✅
- Dynamic Map Tiles ✅
- Geoprocessing Services ✅
- Static Map Caching ✅
- Feature Editing ❌
- Printing ✅
- + More!

What?

- McDonald’s
- Big Mac
- Golden Arches
- Two all-beef patties, special sauce, lettuce cheese, pickles onion on a sesame seed bun

- McDowell’s
- Big Mic
- Golden Arcs
- “My buns have no seeds”

Feature Roadmap – Stuff to figure out

- Improve Dynamic Map Service performance
- On the fly CartoCSS Dynamic Tile rendering
- JavaScript API
- Simple Feature Creation/Editing (…like creating features from a mobile device)
- GeoJSON.io support
- General Performance Enhancements
- Security (https and/or authentication)
- Printing – Post rendering operations (clip, rotate, scale, etc.)
- More basic Geometry operations
- More raster operations
- More Documentation
- Licensing?????

Questions? Thoughts?

- Tell me if this is already being done
- Tell me how to make it better
- Tell me if you'd like to contribute

- https://github.com/spatialdev/PGRestAPI
- Ryan Whitley - @apollolm on Twitter
- www.spatialdev.com

SpatialDev℠