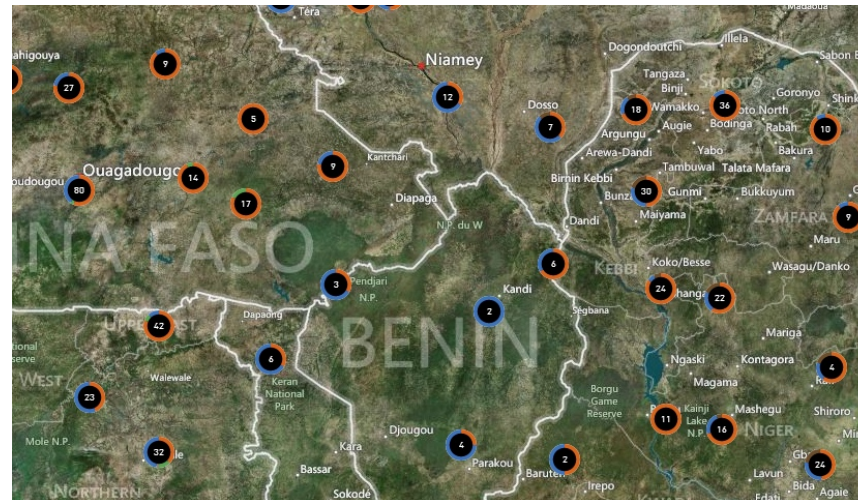
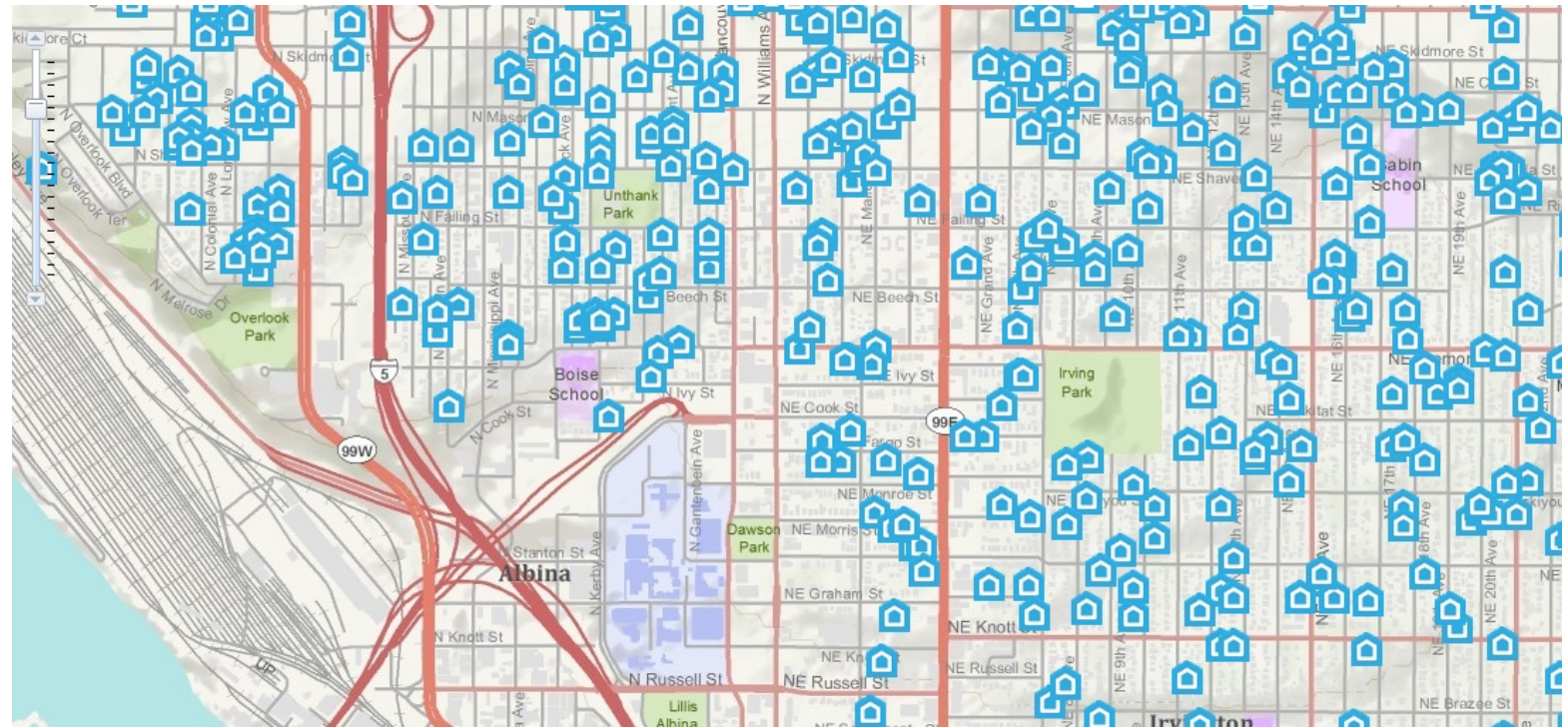
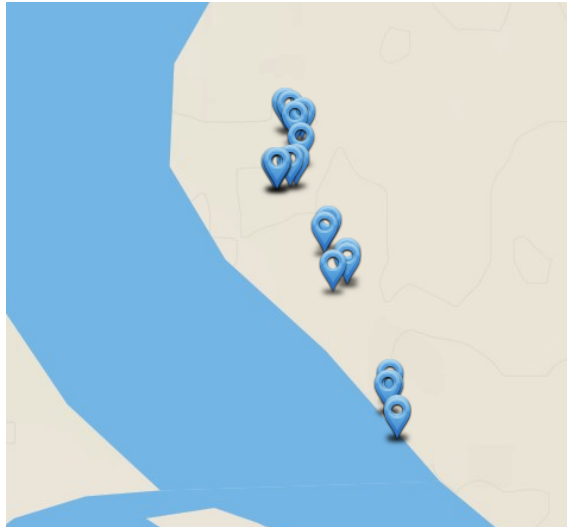


PGRestAPI

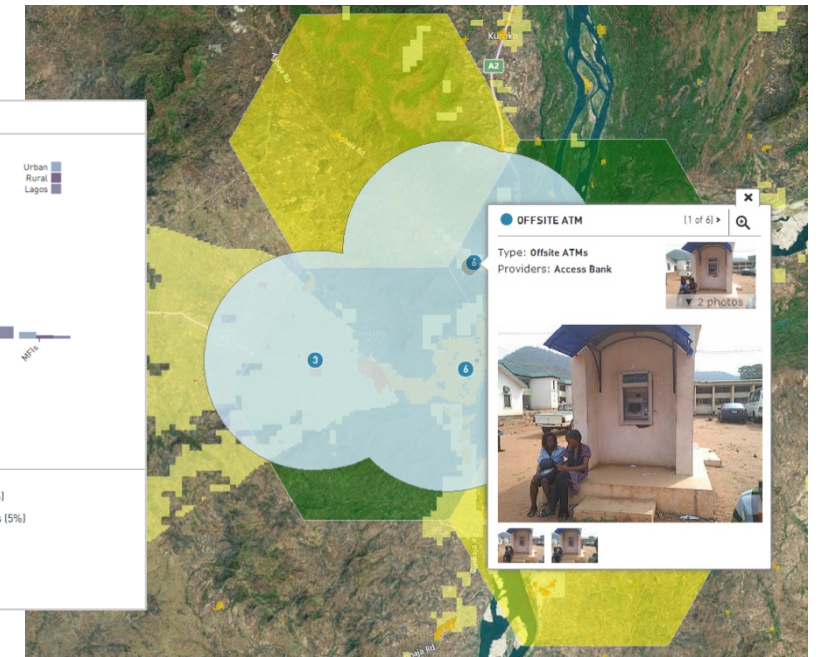
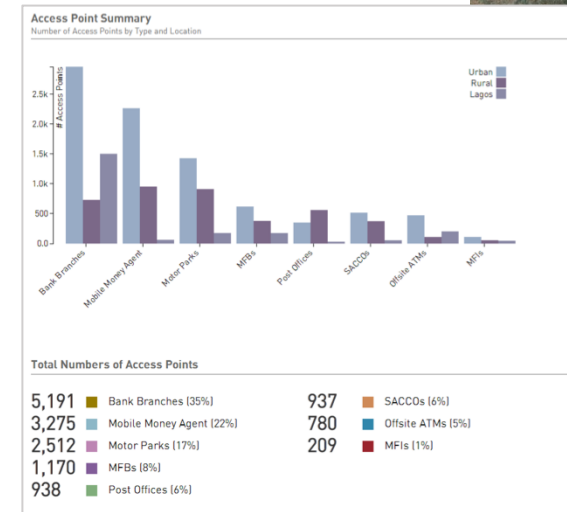
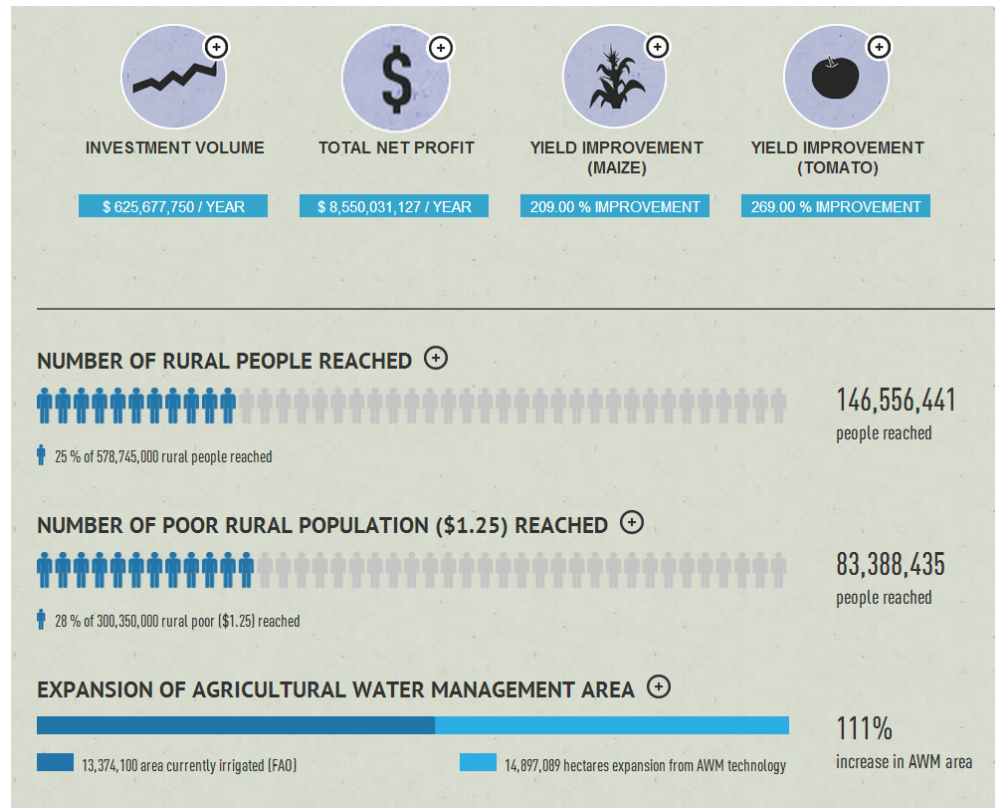
The location sensation that's sweepin' the nation....

<https://github.com/spatialdev/PGRestAPI/>

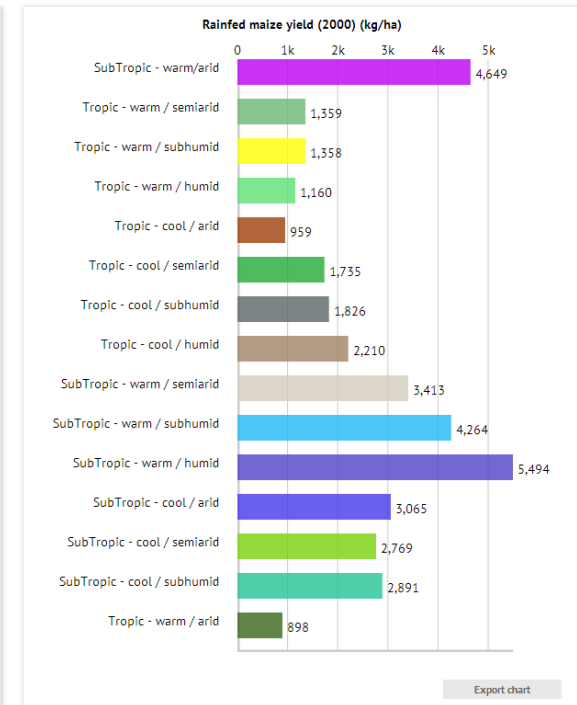
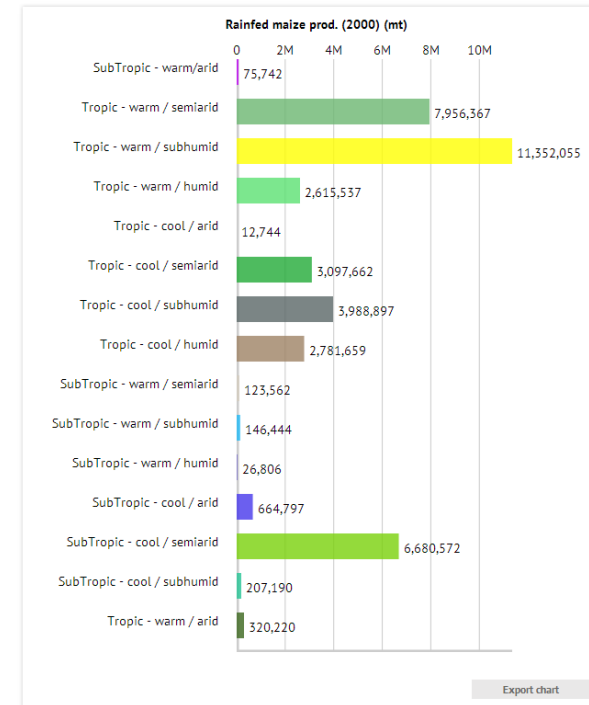
Web maps...



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

ArcGIS REST Services Directory

[Login](#) | [Get Token](#)

[Home](#) > [services](#) > [Bangladesh_MapService_Dev \(MapServer\)](#) > [CICO](#) > [query](#)

[Help](#) | [API Reference](#)

Query: CICO (ID: 0)

Where:	<input type="text" value="FeatureType = 'Bank Branches'"/>
Text:	<input type="text"/>
Object IDs:	<input type="text"/>
Time:	<input type="text"/>
Input Geometry:	<input type="text"/>
Geometry Type:	<input type="text" value="Envelope"/>
Input Spatial Reference:	<input type="text"/>
Spatial Relationship:	<input type="text" value="Intersects"/>
Relation:	<input type="text"/>
Out Fields:	<input type="text"/>
Return Geometry:	<input checked="" type="radio"/> True <input type="radio"/> False
Max Allowable Offset:	<input type="text"/>
Geometry Precision:	<input type="text"/>
Output Spatial Reference:	<input type="text"/>
Return IDs Only:	<input type="radio"/> True <input checked="" type="radio"/> False
Return Count Only:	<input type="radio"/> True <input checked="" type="radio"/> False
Order By Fields:	<input type="text"/>
Group By Fields (For Statistics):	<input type="text"/>
Output Statistics:	<input type="text"/>

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

Getting Started: Ingredients

- Computer (Windows, Ubuntu, Mac)
- Node.js
- PostgreSQL and PostGIS
- Mapnik
- Some data



[SERVICES](#) [TABLE LISTING](#) [GEOPROCESSING](#) [UTILITIES](#)

SERVICES LISTING

A list of services available

[PostGres Table Endpoints](#)

[Geoprocessing](#)

[Utilities](#)

[Static Vector Tile Services](#)

[Static Image Tile Services](#)

Code Sample

Here's how to hit this endpoint programmatically to get a list of these services.

POST

```
1. //define arguments
2. var args = {
3.   format: 'geojson'
4. };
5.
6. //jQuery POST
7. $.post('http://services.fspmaps.com/services', args).done(function() {
8.   //Callback on success
9.   console.log(data);
10. })
```

GET

```
1. http://services.fspmaps.com/services?format=geojson
```



[SERVICES](#) [TABLE LISTING](#) [GEOPROCESSING](#) [UTILITIES](#)

[Home](#) ▶ [Table Listing](#)

TABLE LISTING

A list of your PostGres Tables

[bangladesh_cicos](#)
[bangladesh_coverage](#)
[bangladesh_district_landuse](#)
[bangladesh_districts](#)
[bangladesh_population_raster](#)
[bangladesh_urbanareas](#)

[cicos_2014](#)
[cicos_2014_retrofit](#)

[india_cicos](#)
[india_urbanareas](#)

[kenya_cicos](#)
[kenya_counties](#)
[kenya_coverage](#)
[kenya_district_landuse](#)
[kenya_population_raster](#)
[kenya_urbanareas](#)

[nigeria_cicos](#)
[nigeria_coverage](#)
[nigeria_district_landuse](#)
[nigeria_hexbins](#)
[nigeria_population_raster](#)
[nigeria_statecapitals](#)
[nigeria_states](#)
[nigeria_surveycoverage](#)
[nigeria_urbanareas](#)

[philippines_cicos](#)
[philippines_coverage](#)
[philippines_districts](#)
[philippines_urbanareas](#)

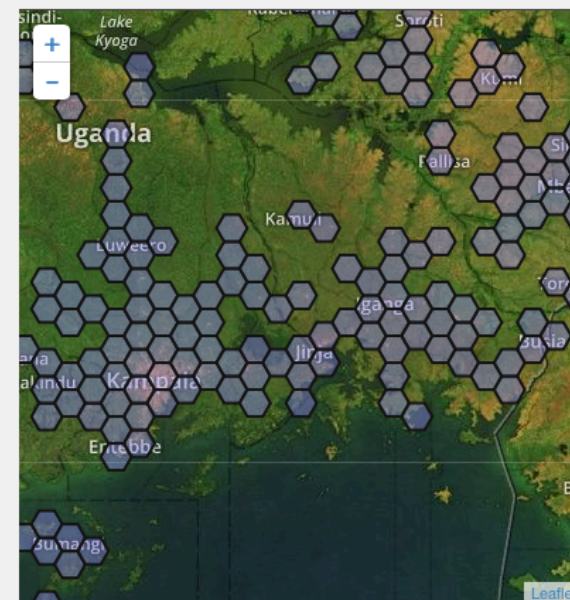
The User Interface



[SERVICES](#) [TABLE LISTING](#)

[Home](#) ▶ [Table Listing](#) ▶ [uganda_hexbins](#) ▶ [Dynamic Vector Tiles](#)

Vector Tile Service



Code Sample

Here's the vector tile endpoint:

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

<http://spatialdev.github.io>

We're Spatial Dev - an award-winning data visualization team.

We use cool technologies to represent complex information in ways that make sense — make important things more visible - help reinforce values that can make a difference — and move things forward.

Open source geospatial GitHub repos

STATIC-API-DOCS

HTML

Create API documentation in Github flavored Markdown or static HTML

PGRESTAPI

JavaScript

Node.js REST API for PostGres Spatial Entities. AKA: SpatialServer

EXPRESS-USER-MANAGEMENT

JavaScript