

CE 2017

Agenda

1. Introduction
2. A bit on developers
3. Introduction ArcGIS bindings for R
4. Introduction to ArcGIS Python API

Introduction

During the second day of EC2017 I attended the developers session and the last technical talk, an introduction to ArcGIS Python API.

The developers session was a light introduction to the geodevelopers.org community resources, as well as some of the key components of ArcGIS platform.

WebGL, Three.js and ArcGIS JavaScript API were briefly introduced to the attendees.

ArcGIS R-bindings, which gives access to ArcGIS users to the vast array of data and statistical analysis of R environment, was succinctly introduced to audience.

Eventually, the last technical talk gave a general overview of the ArcGIS Python API modules and functionalities.

A bit on developers

- ESRI is funding geodevelopers.org, a community for geodevelopers working mainly on ArcGIS technologies, APIs and apps.
- Community members can interact and access the resources via:

Meetup: [meetup.com/Geo-Developers](https://www.meetup.com/Geo-Developers)

Twitter: twitter.com/Geo_Developers

Github: github.com/Geo-Developers

YouTube: [youtube.com/GeoDevelopers](https://www.youtube.com/GeoDevelopers)

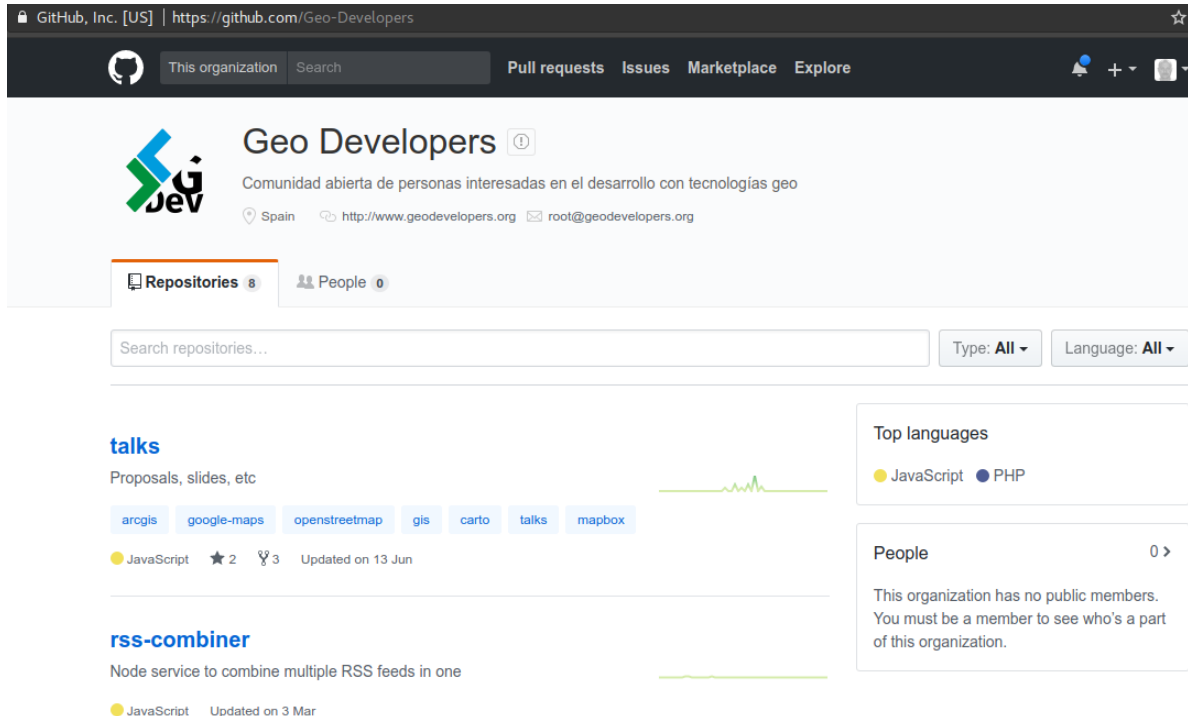
Slack: geodevelopers.org/community

Geo-Developers on GitHub

The screenshot shows the GitHub profile page for the 'Geo-Developers' organization. The header includes the GitHub logo, the organization name 'Geo Developers', and a description in Spanish: 'Comunidad abierta de personas interesadas en el desarrollo con tecnologías geo'. It also lists the location as 'Spain' and provides a website and email address. Below the header, there are tabs for 'Repositories' (8) and 'People' (0). A search bar for repositories is present, along with filters for 'Type' and 'Language'. The main content area displays two repositories: 'talks' and 'rss-combiner'. The 'talks' repository is described as 'Proposals, slides, etc' and includes tags for 'arcgis', 'google-maps', 'openstreetmap', 'gis', 'carto', 'talks', and 'mapbox'. It shows 2 stars and 3 forks, updated on 13 Jun. The 'rss-combiner' repository is described as 'Node service to combine multiple RSS feeds in one' and is updated on 3 Mar. On the right side, there are two sidebars: 'Top languages' showing JavaScript and PHP, and 'People' which states that the organization has no public members.

GitHub, Inc. [US] | <https://github.com/Geo-Developers>

This organization Search Pull requests Issues Marketplace Explore

 **Geo Developers** ⓘ

Comunidad abierta de personas interesadas en el desarrollo con tecnologías geo

Spain <http://www.geodevelopers.org> root@geodevelopers.org

Repositories 8 People 0

Search repositories... Type: All Language: All

talks

Proposals, slides, etc

arcgis google-maps openstreetmap gis carto talks mapbox

JavaScript ★ 2 🍴 3 Updated on 13 Jun

rss-combiner

Node service to combine multiple RSS feeds in one

JavaScript Updated on 3 Mar

Top languages

JavaScript PHP

People 0 >

This organization has no public members. You must be a member to see who's a part of this organization.

A bit of R bindings for ArcGIS

ESRI released a bridge library to connect ArcGIS and R.

This library was developed with the purpose of facilitating management and processing of ArcGIS data for R users.

At the same time it makes easier for ArcGIS users to incorporate all the power of R analysis tools into their workflows.

Scripts structure is very simple, we load the required libraries, we may code additional functions or classes and finally we include a `tool_exec` function.

The `tool_exec` functions requires two parameters, input and output arguments, which will allow us to interact with ArcGIS through a toolbox interface.

Besides writing results to a geodatabase or shapefile, this function may return elements like plots, which will be interactively displayed by ArcGIS.

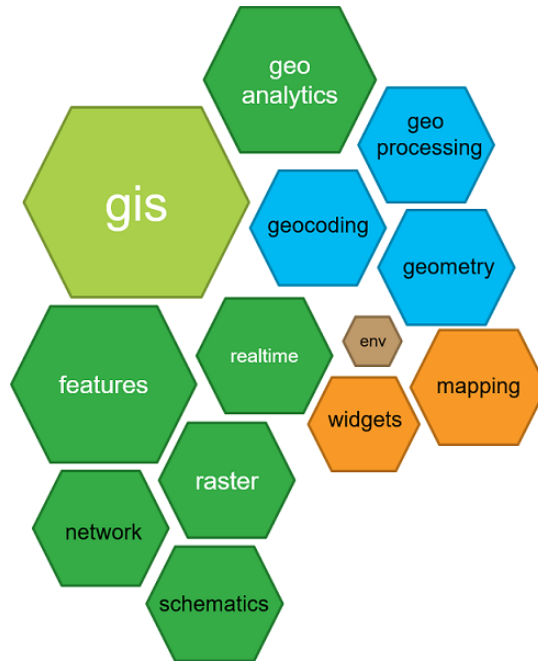
Rbridge contains functions to convert spatial entities to R spatial structures and from R spatial structures to ArcGIS spatial entities

A bit of R bindings for ArcGIS

```
tool_exec <- function(in_params, out_params)
{
  ##### Load Library for Analysis #####
  if (!requireNamespace("SemiPar", quietly = TRUE))
    install.packages("SemiPar")
  require(SemiPar)
  ##### Get Input Parameters #####
  input_features <- in_params[[1]]
  input_predictions <- in_params[[2]]
  # ...
  output_features <- out_params[[1]]
  output_graph_pdf <- out_params[[2]]
  ##### Import Dataset to Dataframe #####
  fc <- arc.open(input_features)
  df <- arc.select(fc, c(dep_variable, nonlin_variables, lin_variables))
  df['x'] <- arc.shape(df)$x
  df['y'] <- arc.shape(df)$y
  ###
  # script functionality
  ###
  # Write data back to shapefile, geodatabase ...
  arc.write(output_features, pred_df)
  # Return objects: plots, ...
  return(out_params)
}
```

Introduction to ArcGIS Python API

- The ArcGIS API for Python provides a **pythonic representation of a GIS**.
- The API has Python *modules*, *classes*, *functions*, and *types* for managing and working with elements of the ArcGIS platform information model.



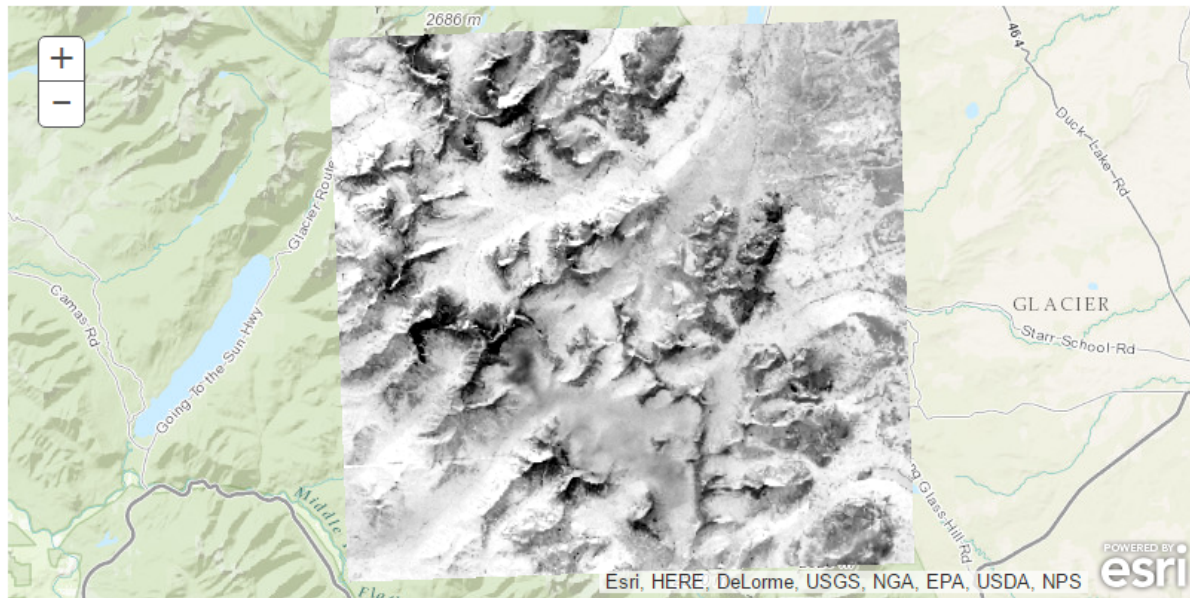
ArcGIS modules

ArcGIS API contains 13 modules which allow advanced users to perform and automatize administration, analysis and visualization tasks.

- **arcgis.gis** module
- **arcgis.env** module
- **arcgis.features** module
- **arcgis.raster** module
- **arcgis.network** module
- **arcgis.schematics** module
- **arcgis.geoanalytics** module
- **arcgis.geocoding** module
- **arcgis.geometry** module
- **arcgis.geoprocessing** module
- **arcgis.realtime** module
- **arcgis.mapping** module
- **arcgis.widgets** module

Python raster analysis

```
import arcgis
from arcgis.gis import GIS
portal = GIS("portal_url", "username", "password")
montana_ndvi = portal.content.search("GeneratedRasterProduct_JRA9W9")[0]
ndvi_map = portal.map(location, 12)
ndvi_map
```



Python Mapping

```
from arcgis.gis import *  
gis = GIS()  
map1 = gis.map('USA',3)  
map1
```

