Collectible MapCards Generator

Scope of work

General description

Script that will generate a picture of a map based of inputs.

This script should expose an HTTP server with the following capabilities:

- POST: generate picture map
  - receive a body param with the needed inputs
  - generate the desired picture
  - upload to s3
  - return response with s3 url
  - store the picture record in a mongoDB with a uuid
- GET: single picture record
  - get picture/card record by uuid
- GET: multiple pictures record
  - get multiple pictures/cards by query (queries will be based of project identifier)

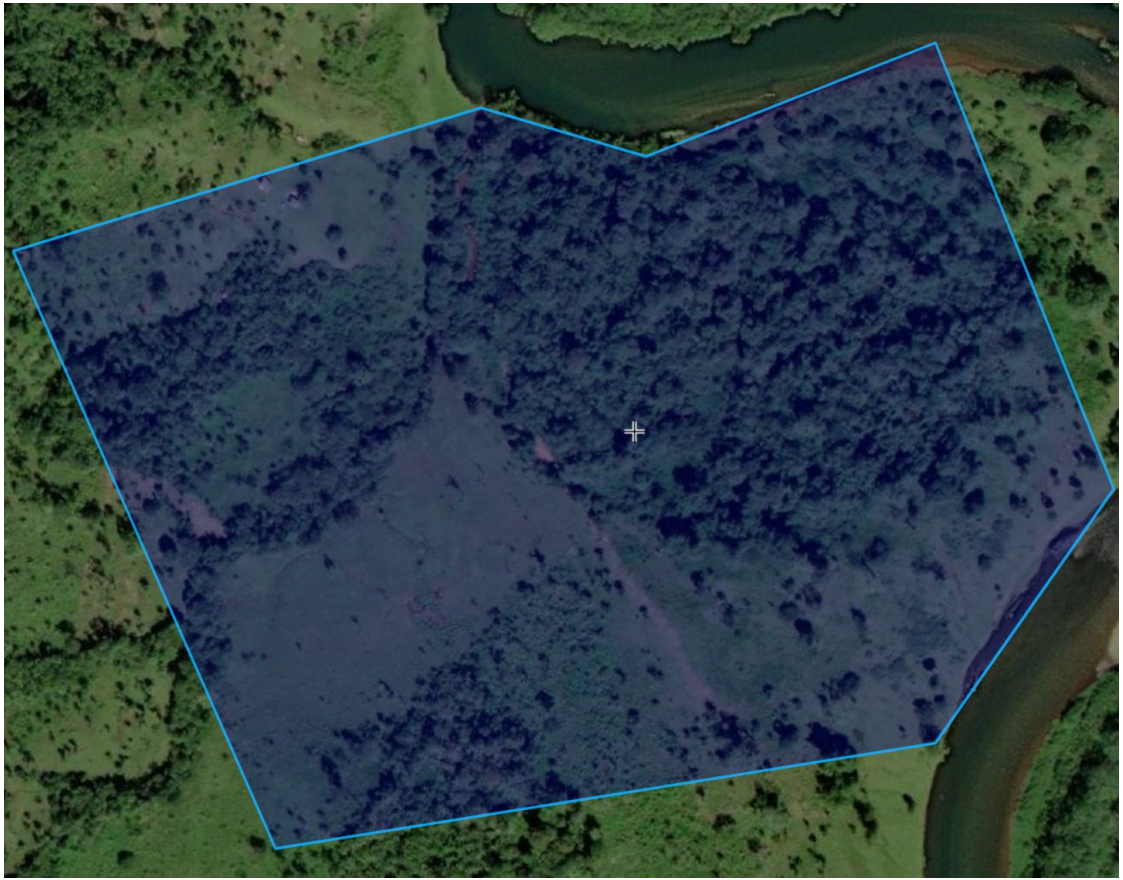
General description

The script should create multiple pictures of maps. Each of these pictures represent a piece of land within the parenting project/territory.

This card will contain multiple components that will describe the attributes/characteristics of piece of land it represents as well as it’s parenting project.

The script should be able to divide the project’s piece of land into small pieces of the same size and create generate the pictures out of each one of this small pieces.

Project land division example:

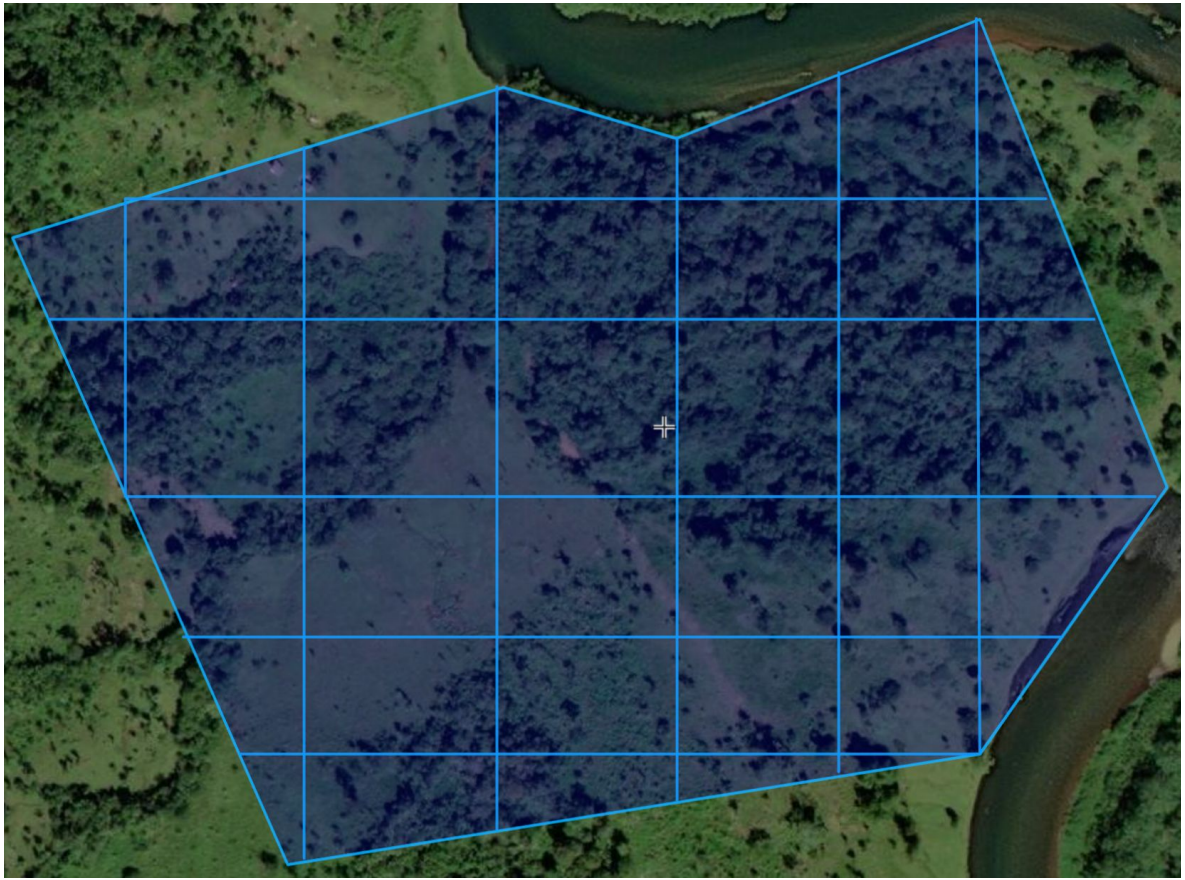


General project land, which has its own polygon of coordinates.

This will be the input for a new set of small pieces of land generated.

This piece of land will contain characteristics that describe stuff like: biodiversity grade, forestation grade, water grade, etc…

Project land division example:

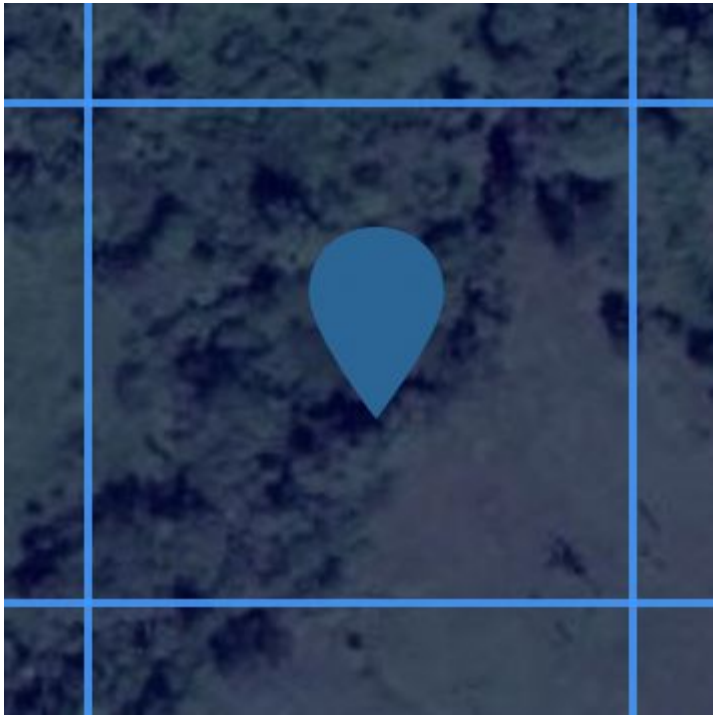


This is how the project will get divided into small pieces of land each representing an individual collectible “card/image”

Each resulting sub-piece of land will have its own polygon of coordinates.

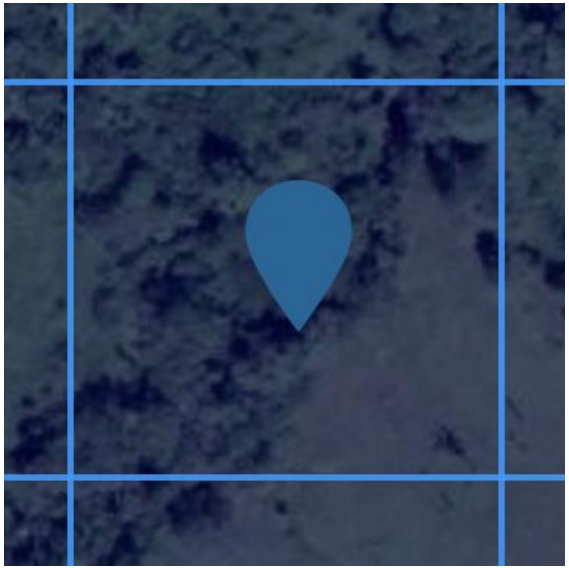
As you can see, this project will result in 37 collectible cards.

Single piece of land



This piece of land will share the projects attributes, and be represented a single collectible card within the project.

Single card attributes



Each resulting card/map will be saved with the following data:

{

id: (id),

project: (parent project id),

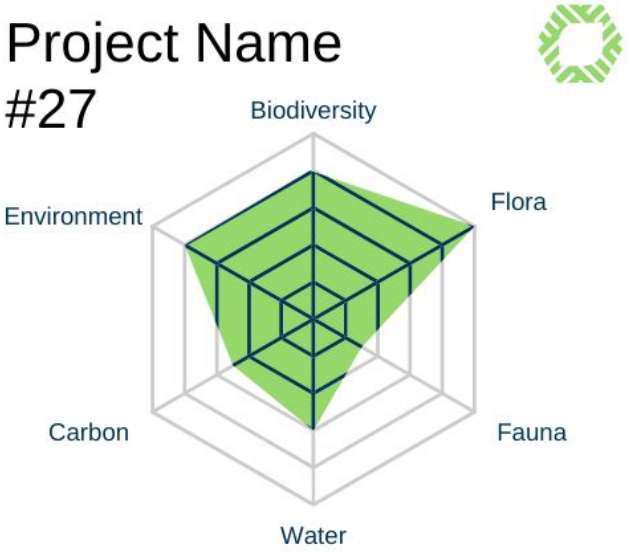
projectAttributes: {summary of project attributes},

polygon: {geoJson to define the piece of land}

}

Resulting card image

Each resulting card should have the following style



Resulting card image Components

Map polygon image with shape borders

Project name

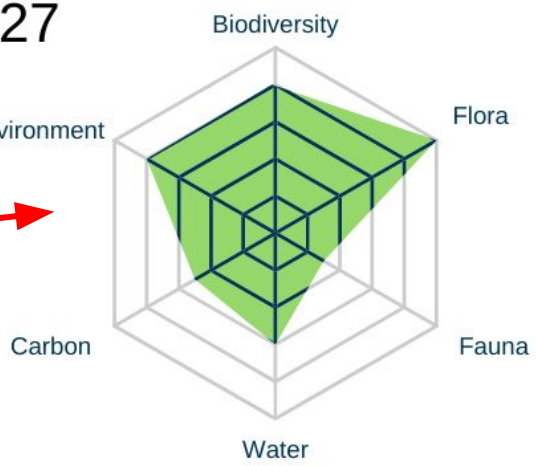
Project Name

Logo

Card/piece id

#27

Radar chart made up of the project's characteristics



Features

1) web HTTP server

This “script” or project should be available as a rest api endpoint in a web server (web microservice) and do its work based of on http calls / webhooks.

The required methods will be:

1. POST: generate cards
    a. Input/body params:
        i. projectId, projectName, **projectMeta** (metadata/descriptive information about the project)
        ii. project land (geojson of the land that defines the project’s extension)
        iii. grid division size: **x** hectares
2. GET: all cards from project
3. GET: single card from project
4. GET: projects
5. GET: single project

2) be written in any of the following programming languages

1. The project should be made using the following tools:
    a. Python
    b. javascript / typescript
    c. php
2. these are the tools our team knows better.

3) generate cards function

the generations of cards should receive the details of a project (see slide 11 for input details) and output the resulting images' names of this process. (can be saved locally for now (I'll make it work with s3 later on).

- The cards generation will split the geojson's land representation into x number of cards/images which are defined by:
  - number of cards: project area / card size
    - Example:
      - project area: 10ha
      - size per image: 1ha
      - this will result in 10 images/cards of 1ha each.
- after a single piece of land is defined, the process should create an image/card with the details described in slide 9.
  - this image will contain a chart that describes the project's characteristics which will be provided in the format that best suits your needs.
- After the image is generated, it should be saved into a folder to be used later on. following a naming convention to identify the card:
  - project-name-id.jpg (example: myFirstProject-12) this will be the 12th card for project: myFirstPorject.