Final Presentation
Team 3: Download of public facing data for registered users

Jerry Bonnell   Gururaj Shriram   Lixiong Liang   Heyu Yao
Erica Chang

May 7, 2018

Agenda

Requirements

Project Purpose

- Users are interested in **downloading**:

Data types

1. **Cadastral** data - points and polygons on a map
2. **"Storytelling"** data - images, videos, audio files, etc.

- **Purpose:** Users can create their own maps from this data
    - e.g. a map pertaining to bus stop information
    - Expedites the process of receiving land titles.

Functional Requirements

- User can choose an **output format** for queried data and download **locally** to computer.
  - **Cadastral** data: Convert data to `Geojson`, `KML`, `shapefile`, `CSV`
  - **"Storytelling"** data: fetch directly for each request

Functional Requirements

- User can choose an **output format** for queried data and download **locally** to computer.
  - **Cadastral** data: Convert data to `Geojson`, `KML`, `shapefile`, `CSV`
    - Also `topojson`!
  - **"Storytelling"** data: fetch directly for each request

Non-functional Requirements

- Must handle up to 3 **simultaneous** requests.

Non-functional Requirements

- Must handle up to 3 **simultaneous** requests.
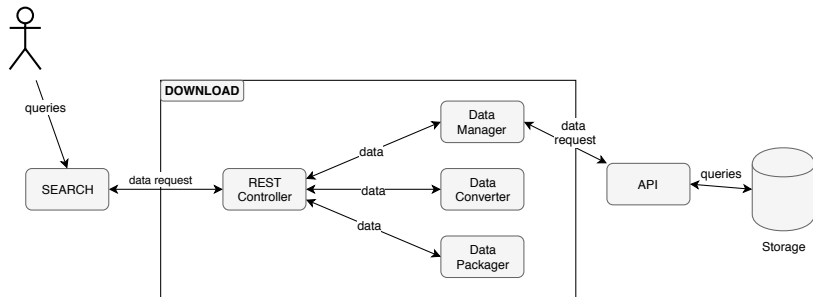- *Tried up to 100 simultaneous requests!*

Architecture

Architectural Style: Three-tier
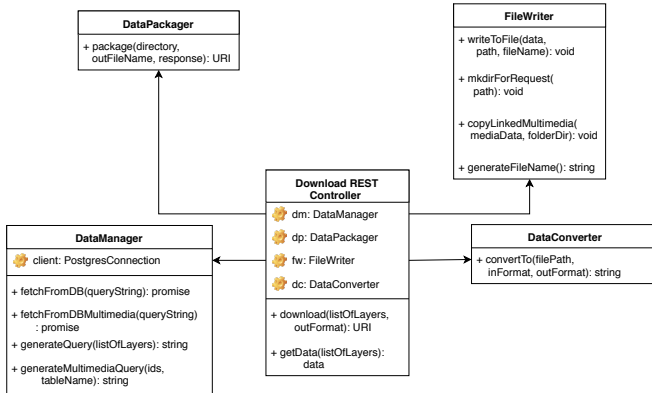
Client-server tier: Allows download server to handle requests

Business tier: converts and archives data

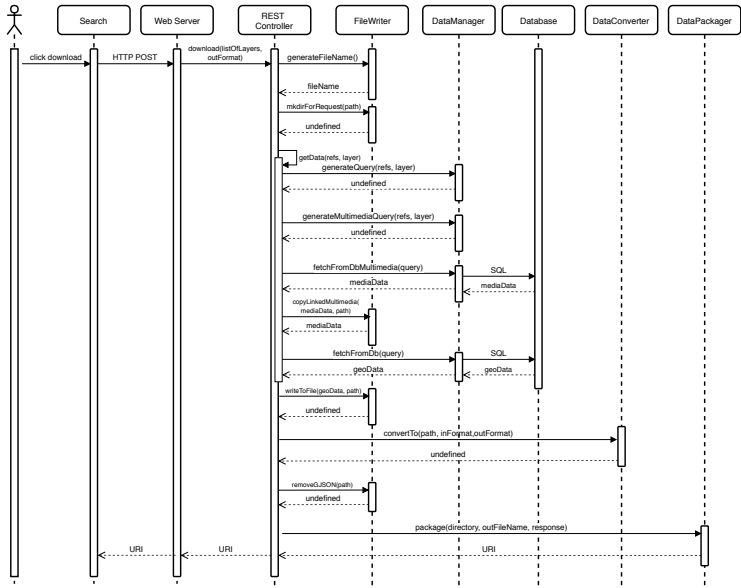Database-centric tier: requests geospatial data from database

System Diagram

Class Diagram

DataPackager

+ package(directory, outFileName, response): URI

FileWriter

+ writeToFile(data, path, fileName): void

+ mkdirForRequest(path): void

+ copyLinkedMultimedia(mediaData, folderDir): void

+ generateFileName(): string

Download REST Controller

🔹 dm: DataManager

🔹 dp: DataPackager

🔹 fw: FileWriter

🔹 dc: DataConverter

+ download(listOfLayers, outFormat): URI

+ getData(listOfLayers): data

DataManager

🔹 client: PostgresConnection

+ fetchFromDB(queryString): promise

+ fetchFromDBMultimedia(queryString) : promise

+ generateQuery(listOfLayers): string

+ generateMultimediaQuery(ids, tableName): string

DataConverter

+ convertTo(filePath, inFormat, outFormat): string

Sequence Diagram

Demo!

Questions?