Smogon API Final Presentation

By Jason Wang, Marques J Chacon, Ramiro Steinmann Petrasso, and Yangyang Yao

Background

- https://Smogon.com is a website that provides data on the competitive viability of different Pokemon in the popular video game series
  - It divides Pokemon into “Tiers” based on how good they are relative to other Pokemon in the same game. The most popular tier is “OverUsed”, or OU, which allows every Pokemon except those that the community has elected to ban
- Although Smogon data is publicly available, it is difficult to access in the context of software: it is only available in the website’s raw HTML, which is difficult to work with when coding
- Our project is focused on creating an API that allows its users to programmatically access data from Smogon
- Using simple GET requests, an app that uses our API should be able to receive the public data on Smogon, but in JSON format

Data Used

- https://smogon.com
  - The main focus of the project
  - Contains competitive data on the Pokemon from each game
  - Lacks key info needed to get the Pokemon in the game in the first place, like the evolutionary requirements.
- https://www.kaggle.com/datasets/mrdew25/pokemon-database
  - The static dataset is used to provide static data for supplemental queries like the evolutionary requirements of a Pokemon, which is not available directly on Smogon but is useful regardless.
  - Mostly just here to grab some key details Smogon does not list

Use Cases

- User Alpha wants to practice his web development skills. They decide to make a reproduction of Smogon.com, because they are passionate about competitive Pokemon and want to make something they are interested in. In order to do so, they will need a way to access the Smogon data.
- User Brown is a data science student. They are taking a data visualization course and have been assigned a project where they must display a dataset of their choice. They decide to make a bar graph showing how the average attack stat for a Pokemon has changed across Pokemon generations. To do so, they need to find data on Pokemon by gen so they can aggregate Attack Stat values.
- User Charlie is a competitive pokemon player and amateur coder. They would like a faster way to build teams than to scroll through each Pokemon's Smogon page and check which one has sample sets. They decide to write a program to find the sample sets or Pokemon with high values in a certain stat

Design

- The main components of our API are:
  - User facing API interface
    - The user makes a data request either through the url bar/http request or through the GUI at https://smogonapi.herokuapp.com/docs
  - Component 2 - Service request handler
    - Each endpoint finds its data either using the request package or through a function in the package that parses the static data
  - Component 3 - Smogon webpage URL generator
  - Component 4 - Smogon webpage crawler
    - For both components 3 and 4, if the endpoint interacts with Smogon, the function will generate a URL and then use beatifulsoup to scrape it
- By chaining these components together, the API repackages Smogon data for the user to process through their own code

Demo

https://www.youtube.com/watch?v=w6nxmmZwPxw

Lessons learned and future work

Lessons learned-

1. Be mindful for cross-platform support, e.g. Windows vs. OSX
2. Beware of implicit package structure assumptions, e.g. Heroku vs. Github Actions
3. Always add test command & results in code reviews
4. Ask for the right approvers for the right code

Future work

1. More structured exceptions and error handling
2. Client package with more user friendly functions & multi-language support
3. Move from online scraping to pre-scraped lookup with external data storage

Thank you!