API Tennis Project: A Comprehensive Application for Tennis Data and Tournament Information

Victor Racelescu

January 12, 2025

1 Introduction

The Tennis Project is a Python-based application designed to offer a seamless experience for users interested in tennis-related data. The project employs the Tkinter library to create a graphical user interface (GUI) that allows users to interact with the application and retrieve useful information. Key features include viewing rankings, searching for player details, exploring ATP tournament data, and checking weather forecasts for specific locations.

The application connects to various online data sources via APIs, enabling real-time retrieval of rankings, player statistics, tournament information, and weather data. These features are organized into distinct windows accessible through the main menu, providing a comprehensive toolkit for tennis-related inquiries.

2 Project Structure

The project is divided into several Python files, each responsible for a different aspect of the application. These files are as follows:

- `main.py` - This file contains the main program, which initializes the Tkinter window and sets up the main menu, including buttons to access different features of the application.

- `search.py` - Contains the functionality for searching player details by name. It allows users to enter a player's first and last name and retrieves detailed statistics such as rank, points, and country of origin.

- `rankings.py` - Provides the functionality for displaying player rankings by gender. It fetches the top players' rankings for men and women and displays them in the user interface.
- `tournaments.py` - Handles the functionality for displaying ATP tournament data. Users can search for tournaments by year or city, and the application fetches relevant tournament details from an external API.
- `weather.py` - Contains a function for fetching weather data for a given city, providing users with up-to-date weather forecasts related to specific tournament locations.
- `config.py` - Stores configuration data such as API keys, URLs for external APIs, and headers required for making requests to external services.

3 Main Functionality

The application opens with a main window that provides users with several buttons, each corresponding to a different feature. These buttons allow users to access detailed views for player rankings, player search, ATP tournaments, and more. Each feature is implemented in its own separate window, created using Tkinter's `Toplevel` method.

3.1 View Rankings

In the `show_rankings_window` function, users can input a gender ('men' or 'women') and specify how many top players they wish to view. The application then fetches the rankings from an external API, processing the data to display player names, ranks, and points. The top players are displayed in a Tkinter `Text` widget, which provides a scrollable area for the results.

3.2 Search Player

The `show_player_search_window` function allows users to search for players by first name and surname. When the user enters this information and clicks the search button, the application queries the external API and displays the player's rank, points, and other relevant statistics in a `Text` widget. If the player is not found, an error message is shown.

3.3 ATP Tournaments

Users can view ATP tournaments by year or city. In the `show_atp_tournaments_window` and `show_tournaments_by_city_window` functions, users can input a year or city to search for tournaments. The application then makes a request to the external API, fetching tournament names, locations, and dates. This data is displayed in a scrollable text box. The application also provides the option to view weather information for the tournament location.

3.4 Weather Information

For users interested in weather forecasts for specific tournament locations, the `fetch_weather_for_city` function allows users to view the current weather for a selected city. This feature helps users keep track of conditions at specific tournament locations, particularly when planning to attend events.

4 API Integration

The application integrates with multiple external APIs to retrieve real-time data:

- **Tennis Rankings API:** The rankings data is fetched using the Sportradar Tennis API. The API provides up-to-date information on the world's top-ranked male and female players, their points, ranks, and competition details.

- **ATP Tournament API:** Tournament data is retrieved from the Ultimate Tennis API. This API provides detailed information on ATP tournaments, including tournament names, locations, dates, and other relevant details.

- **Weather API:** The application uses the OpenWeather geocoding API to fetch weather forecasts based on the selected city. This allows users to get current weather conditions for the locations of tournaments.

5 User Interface

The user interface of the application is built using Tkinter, a standard Python library for GUI development. The main window includes several buttons, each of which opens a new window for a specific feature. For example, the ”View Rankings” button opens a window where users can specify the gender and number of players they want to see. Similarly, the "Search Player" button opens a window where users can search for player information.

The application uses `Label`, `Entry`, and `Button` widgets to allow users to input data and interact with the program. Additionally, `Text` widgets are used to display large amounts of data in a scrollable area. Error handling is implemented throughout the application using Tkinter's `messagebox` to alert the user when incorrect input is provided or when data fetching fails.

6 Conclusion

The Tennis Project is a comprehensive Python application that integrates several external APIs to provide a wide range of tennis-related data. By using Tkinter for the user interface, the project offers a responsive and interactive experience. Users can easily access player rankings, search for player details, view ATP tournaments, and check weather forecasts for tournament locations. This project demonstrates the ability to work with external APIs, create interactive graphical user interfaces, and handle various types of data in Python.

This project can be further extended to include more detailed features, such as adding user authentication, creating a database of historical tournament results, or integrating additional APIs for more diverse information. The flexibility of the Tkinter framework and the power of Python make this application a solid foundation for building more advanced tennis-related tools.