Data Collection

---

1. GET request

To utilize the API effectively, one can browse through the API documentation to identify the appropriate URL and parameters needed for constructing a GET request.

Below is a sample from the API Specifications for Real-time "Next Bus" arrival times and associated data from Citybus Limited:

Citybus API Specifications

By consulting the documentation, you can attempt the following tasks:

- Utilize the `Route API` to retrieve route information for bus route `A28`.
- Employ the `Route-Stop API` to obtain the stop information for bus route `A28`:
  - Set the direction to `inbound`.
  - Note that the output from the `Route-Stop API` will provide a list of `stop_id` entries, which uniquely identify each stop but do not include the stop names.
- Combine the outputs from the `Stop` and `Route-Stop API` to ascertain the names of the stops along bus route `A28`.
- Identify the `stop_id` for the stop `將軍澳站, 寶邑路` on route `A28`, ensuring the direction is set to `inbound`.
- Use the `ETA API` to determine the estimated time of arrival for the next bus at the stop `將軍澳站, 寶邑路` on route `A28`, with the direction set to `inbound`.

By following these instructions, you can harness the full potential of the Citybus API to access real-time transit data.

---

2. POST request

The Azure OpenAI API is an advanced tool that enables the creation of human-like text through the power of ChatGPT. As a member of the HKUST community, you have the privilege to access and utilize this service.

Every month, HKUST provides a complimentary credit of HKD 8.00, which is renewed automatically on the first day of each month at 8:00 AM. This credit is shared across the HKUST ChatGPT platform usage. Should you require more resources, additional credits can be purchased.

For detailed guidance on the OpenAI API Service provided by HKUST, please refer to the following resource:

UST OpenAI API Service Documentation

To start leveraging the OpenAI API for text generation, you should:

- **Sign up for the OpenAI API Service**
- **Secure Your API Key**: Once subscribed, you will receive an API key.
  - **Remember to treat your API key as confidential information; do not disclose it to others, as it is linked to your account and its activities.**
- **Initiate a POST Request**: Utilize Python's `requests` library to make a POST request to the API endpoint.
  - In your request, include the `api-key` header with your personal API key and set the `Content-Type` header to your preferred format, such as `application/json`.
  - Consult the documentation for a complete list of necessary parameters.
- **Experiment and Create**: Use the API to generate a variety of texts.
  - Unleash your creativity by crafting poems, stories, dialogues, or any text format that sparks your interest.

---

3. Web Scraping

Web scraping is a method used to extract data from websites. This can be achieved by utilizing the `requests` library to issue a GET request to the target website, followed by employing the `BeautifulSoup` library to parse the HTML content. This process facilitates the extraction of the desired information. Wikipedia serves as an excellent resource for practicing web scraping techniques.

For practicing purposes, consider the following tasks centered around the List of Formula One World Drivers' Champions Wikipedia page:

- Determine the ages of the youngest and oldest Formula One World Drivers' Champions as listed on the page, also the mean value of the ages, and create a bar chart to display the age distribution among the champions.
- Extract the percentage points data (`% points`) for each World Drivers' Champion from the page, and construct a line chart to depict the distribution of `% points` across the champions over the

years.

- Generate a scatter plot to correlate the `% points` data with the ages of the World Drivers' Champions, allowing for a visual analysis of any potential relationship between the two datasets.