**API Udacity**

Для свого проекту я вирішила працювати з Udacity API. [Udacity](https://www.udacity.com/) – освітня платформа, яка пропонує масові онлайн курси з ціллю демократизації освіти. Працювати саме з Udacity я вирішила, адже я цікавлюсь онлайн курсами і цікаво попрацювати з цими даними і, в результаті, навіть допомогти іншим з вибором найбільш підходящих курсів, в чому і полягатиме мій проект.

[API](https://www.udacity.com/catalog-api) Udacity є деталізованим і доволі вичерпним, і вміщує в собі інформацію про всі курси та навчальні програми на Udacity. [Тут](https://www.udacity.com/public-api/v0/courses) знаходяться самі представлені дані.

HTTP method: GET

Query Parameters: None

Response Format: API Udacity відповідає з контентом типу application/json. У випадку успішного запиту і відповіді, кінцевою точкою API є об’єкт JSON (JavaScript Object Notation; текстовий формат обміну даними між комп'ютерами. JSON базується на тексті, може бути прочитаним людиною. Формат дозволяє описувати об'єкти та інші структури даних. Цей формат головним чином використовується для передачі структурованої інформації через мережу).

Якщо очікуване значення є рядком, і немає ніякого контенту для конкретного значення, то значення автоматично буде порожнім рядком. Довгі рядки (наприклад, короткий опис курсу або програма і розклад курсу) є відформатовані у форматі Markdown (полегшена мова розмітки даних, яку створено з ухилом на прочитність та зручність у публікації з подальшим перетворенням її на structurally valid XHTML або HTML). У конкретному випадку цього API, Markdown рядки можуть містити HTML-теги. Всі рядки використовують Unicode кодування (стандартний спосіб кодування символів будь-яких мов).

Тепер трохи детальніше про те, як буде виглядати відповідь на запит даних Udacity.

Об'єкт відповіді має три ключові значення: “courses”, “tracks” i “degrees”.

Courses i Degrees містять в собі наступні дані:

* key (string) – унікальний код курсу чи сертифікату
* title (string) – назва курсу
* homepage (string) – посилання на домашню сторінку. Включає в собі два параметри запитів (utm\_medium=referral і utm\_campaign=api)
* subtitle (string) – підназва курсу
* level (string) – один з трьох рівнів (beginner, intermediate чи advanced – може ще бути нічого не вказано. Цей пункт буде одним з ключових під час моєї роботи над цим проектом, адже я буду використовувати ці дані для дослідження та подальшої розробки)
* starter (boolean) – чи підходить курс для когось з нетехнічним бекґраундом (також використовуватиму)
* image (string) – посилання на зображення курсу
* banner\_image (string) – посилання на баннер-зображення курсу
* teaser\_video – об’єкт з ключем посилання на Youtube
* summary (string) – в форматі Markdown (використовуватиму для пошуку ключових слів, і, відповідно, категорій, до яких належить курс)
* short\_summary (string) – в форматі Markdown
* required\_knowledge (string) – список необхідних навичок в форматі Markdown
* expected\_learning (string) – що користувач навчиться на курсі, в форматі Markdown
* featured (boolean) – чи курс знаходиться в топі (також буду використовувати це для своїх досліджень і подальшої роботи над проектом)
* syllabus (string) – список уроків в форматі Markdown
* faq (string) – часто задані питання, в форматі Markdown
* full\_course\_available (boolean) – чи є доступний сертифікат
* expected\_duration (int) – якщо 0, то нема чітко визначеного таймлайну / дедлайну; використовує також наступний пункт (expected\_durarion\_unit)
* expected\_durarion\_unit (string) – порожньо, days, weeks чи months
* new\_release (boolean) – чи буде знову випущений
* tracks (array of strings) – до яких категорій курс належить
* affiliates (array of objects) – назва організацій, компаній, університетів
* instructors (array of objects) – name (string), bio (string), image (string) інструкторів курсу

Tracks містить наступні дані:

* name (string) – назва
* courses (array of strings) – курси, наявні в цьому треку
* description (string) – детальний опис треку

Парсити код можна кількома способами:

Використовуючи бібліотеки json та urllib\_request

1. **import** json  
   **import** urllib.request  
   response = urllib.request.urlopen('https://udacity.com/public-api/v0/courses')  
   json\_response = json.loads(response.read().decode('utf8'))  
   **for** course **in** json\_response['courses']:  
    print(course['title'])  
    print(course['homepage'])

Безпосередньо використовуючи python-бібліотеку udacity (як вказано [тут](https://github.com/tylucaskelley/udacity-api-python#catalog))

1. **import** udacity  
   c = udacity.Catalog()  
   tracks = c.tracks()  
   track\_names = [t['name'] **for** t **in** tracks]  
   web\_dev\_teachers = c.instructors('cs253')  
   nd001\_description = c.degree('nd001')['expected\_learning']  
   print(track\_names)

В разі виникнення помилок, відповіддю буде об’єкт типу JSON з ключами error та status.

**API Google Trends**

Також для мого проекту мені потрібне буде використання додаткового API від [Google Trends](https://trends.google.com/trends/). На жаль, для цього інструменту є лише неофіційні API, наприклад, [отакий](https://github.com/GeneralMills/pytrends) і [такий](https://github.com/suryasev/unofficial-google-trends-api). Також дещо ускладнює задачу, що дані Google Trends потрібно буде викачувати окремо кожен і вручну, отримуючи csv-файл, над яким треба буде ще здійснювати додаткові дії, аби його зробити придатним для використання.