

Викладач: Ренгевич О.В.

Предмет: Інформатика

Урок 30

15.06.2022

Тема уроку: Прикладний програмний інтерфейс

Вивчення нового матеріалу

Перегляньте навчальну презентацію за посиланням:

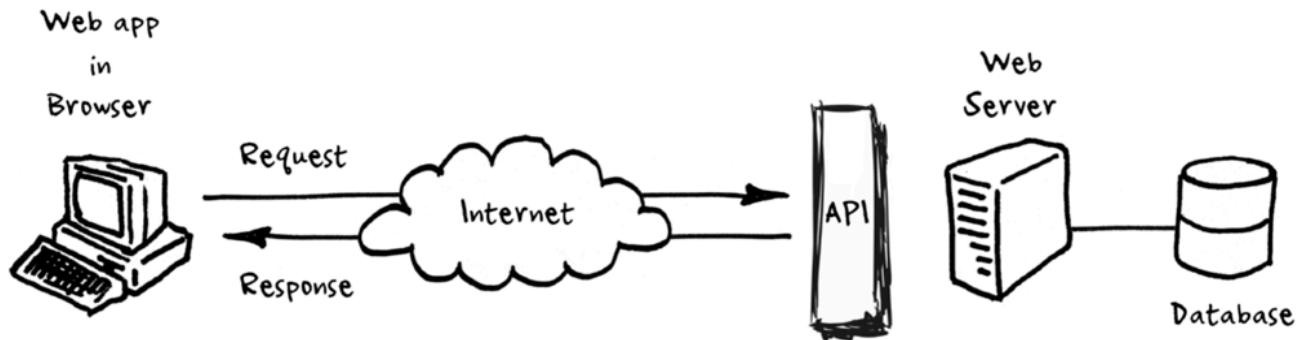
https://docs.google.com/presentation/d/1DoRuYgGgFLv2VjK0wWDAFaJJuqZ1wYVzi_UN9P-LnOQ/edit#slide=id.p12

Прикладний програмний інтерфейс (інтерфейс програмування застосунків, інтерфейс прикладного програмування) (*Application Programming Interface, API*) — набір означень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення. Спрощено – це набір чітко визначених методів (функцій) для взаємодії різних компонентів. API надає розробнику засоби для швидкої розробки програмного забезпечення оскільки можна скористатися готовими об'єктами (функціями) іншого програмного забезпечення через означені в останньому правила взаємодії. API може бути для веб-базованих систем, операційних систем, баз даних, апаратного забезпечення, програмних бібліотек.

При використанні прикладного програмного інтерфейсу в контексті веб-розробки, як правило, API визначається набором повідомлень-запитів HTTP та структурою повідомлень-відповідей. Повідомлення можуть мати різний формат, як правило це XML або JSON. Доступ відбувається до з однієї або декількох загально доступних кінцевих точок (endpoints).

Кінцеві точки є важливими аспектами взаємодії з веб-інтерфейсами на стороні сервера, оскільки вони вказують, де знаходяться ресурси, доступ до яких може отримати стороння програма. Зазвичай доступ здійснюється через URI, до якого надсилаються HTTP-запити, і звідки очікується відповідь. Кінцеві точки повинні бути статичними, інакше правильне функціонування програмного забезпечення, яке взаємодіє з нею, не може бути гарантоване. Якщо місце розташування ресурсу змінюється (і разом з ним кінцева точка), то раніше написане програмне забезпечення буде перервано, оскільки потрібний ресурс більше не може бути знайдено в одному місці. Оскільки постачальники API все ще хочуть оновлювати свої веб-API, багато хто з них запровадили систему версій в URI, яка вказує на кінцеву точку, наприклад, Clarifai API: кінцева точка для функцій позначення в Web API має такий URI: “https://api.google.com/v1/tag/ “. “/V1 /” частина URI визначає доступ до першої версії веб-API. Якщо Clarifai вирішить оновити до другої версії, вони можуть це зробити, зберігаючи при цьому

підтримку стороннього програмного забезпечення, яке використовує першу версію.



Веб-інтерфейси Web 2.0 часто використовують взаємодії на основі таких технологій як REST та SOAP. У той час як прикладний програмний інтерфейс у Web історично був практично синонімом для веб-служби, останнім часом тенденція змінилася (так званий Web 2.0) на відхід від Simple Object Access Protocol (SOAP) на основі веб-сервісів і сервіс-орієнтованої архітектури (SOA) до більш прямих передач репрезентативного стану (REST) стилів веб-ресурсів та ресурсно-орієнтованої архітектури (ROA). RESTful Веб-інтерфейси зазвичай базуються на основі методів HTTP для доступу до ресурсів за допомогою URL-кодуваних параметрів та використання JSON або XML для передачі даних. На відміну від цього, протоколи SOAP стандартизуються W3C і мандату на використання XML як формату корисного завантаження, як правило, над HTTP. Крім того, веб-API на базі SOAP використовує перевірку XML для забезпечення структурної цілісності повідомлень, використовуючи XML-схеми, забезпечені документами WSDL. Документ WSDL точно визначає повідомлення XML та транспортні прив'язки веб-служби.

При використанні деяких Web API, що мають певні обмеження для використання або потребують ідентифікації програмного забезпечення, що викликається, необхідно вказувати API key. **Application programming interface key (API key)** – це код, який передається комп'ютерними програмами, викликаючи прикладний програмний інтерфейс (API) на веб-сайті, для ідентифікації викликаючої програми, її розробника чи її користувача. API key використовуються для відстеження та керування використанням API, наприклад, для запобігання зловмисному використанню або зловживання API (як це визначено, можливо, умовами надання послуг). API key часто виступає і як унікальний ідентифікатор, так і секретний маркер (токен) для автентифікації, і, як правило, має набір прав доступу до пов'язаного з ним API. API key можуть базуватися на універсально унікальному ідентифікаторі (UUID) щоб забезпечити унікальність кожного користувача.

Самостійна робота

1. Опрацюйте навчальний матеріал "Прикладний програмний інтерфейс".
2. Зробіть стислий конспект.
3. Дайте відповіді на питання подані в навчальній презентації.
4. Виконайте на ПК завдання: **продовжуйте працювати над написанням власного сайту.**

Під час роботи за ПК дотримуйтесь правил безпеки.

1. Працюйте з чистими руками.
2. Не робіть різких ударів при роботі з клавіатурою.
3. Припиняйте роботу при появі незвичайного звуку, запаху або самовільного вимикання ПК і негайно повідомте про це дорослих.
4. Після виконання роботи не забудьте вимкнути комп'ютер.

ДОМАШНЄ ЗАВДАННЯ:

1. Виконайте самостійну роботу.

Зробіть скриншот виконаного домашнього завдання та надішліть на e-mail:

ksenija.rengevich@gmail.com або на Viber (096 699 17 66)