

DOI: <https://doi.org/10.36910/6775-2524-0560-2025-59-18>

УДК 004.9

Коломось Г. Геннадій Павлович, к.ф.-м.н., доцент

<https://orcid.org/0000-0003-3667-0316>

Запорізький національний університет, м. Запоріжжя, Україна

РОЗРОБКА ОБ'ЄКТНО-ОРІЄНТОВАНОЇ БІБЛІОТЕКИ JAVA ДЛЯ ВЗАЄМОДІЇ З ПЛАТФОРМОЮ ЕЛЕКТРОННОЇ КОМЕРЦІЇ WOOCOMMERCE

Коломось Г. П. Розробка об'єктно-орієнтованої бібліотеки Java для взаємодії з платформою електронної комерції WooCommerce. Виконано проектування моделі класів та розробка на її основі Java-бібліотеки із засобами для роботи з сутностями популярної платформи електронної комерції WooCommerce, включаючи товари, покупців та замовлення. Перевагами розробленої бібліотеки є підвищена безпека використання типів на рівні компіляції, досягнута за рахунок використання перелічень (англ. enumerations) та об'єктних типів для атрибутів класів моделі, на противагу роботі з рядками при організації взаємодії з безпосереднім використанням запитів WooCommerce REST API. Спроектвана модель використовує засоби програмної бібліотеки Project Lombok для зменшення обсягу шаблонного коду, обмеження доступу до атрибутів класів у відповідності з документацією WooCommerce REST API та визначення методів-"будівельників" об'єктів із можливістю гнучкої ініціалізації їх властивостей. Засобами бібліотеки Gson реалізована користувацька серіалізація-десеріалізація деяких атрибутів об'єктів WooCommerce до більш зручних та безпечних при програмуванні типів даних. Для організації запитів із підтримкою авторизації використовувалася бібліотека OkHttp. Пропонована бібліотека має відкритий програмний код і представлена на GitHub. Її використання може мати прикладну користь при організації взаємодії програмних систем на Java з Інтернет-магазинами на платформі WooCommerce.

Ключові слова: платформа електронної комерції WooCommerce, програмна бібліотека, класи об'єктної моделі, WooCommerce REST API, безпека типів, серіалізація-десеріалізація даних.

Kolomoiets H. Development of the Object-Oriented Java Library for Interaction with the WooCommerce E-Commerce Platform. A class model was designed and a Java library was developed based on it with tools for working with entities of the popular e-commerce platform WooCommerce, including products, customers, and orders. The advantages of the developed library are increased type safety at the compilation level, achieved by using enumerations and object types for model class attributes, as opposed to working with strings when organizing interaction with direct use of WooCommerce REST API requests. The designed model uses the tools of the Project Lombok software library to reduce boilerplate code, restrict access to class attributes in accordance with the WooCommerce REST API documentation, and define object "builder" methods with the ability to flexibly initialize their properties. The Gson library tools implemented custom serialization-deserialization of some WooCommerce object attributes to more convenient and secure data types for programming. The OkHttp library was used to organize requests with authorization support. The proposed library is open source and available on GitHub. Its use can be useful in organizing the interaction of Java software systems with online stores on the WooCommerce platform.

Keywords: WooCommerce e-commerce platform, software library, object model classes, WooCommerce REST API, type safety, data serialization-deserialization.

Постановка проблеми. Електронна комерція є невід'ємною частиною людського життя і забезпечує зручність та ефективність маркетингу та торгівлі. Велика частка сайтів електронної комерції побудована на базі систем керування контентом, серед яких безумовним лідером є WordPress [1]. Додання до сайту на базі WordPress плагіну платформи електронної комерції WooCommerce перетворює його на Інтернет-магазин з широким функціоналом сучасних інструментів торгівлі та маркетингу [2]. Популярність цієї системи керування контентом та платформи електронної комерції пояснюється, зокрема, їх вільним розповсюдженням та відкритим вихідним кодом. Важливою перевагою WooCommerce є наявність прикладного програмного інтерфейсу WooCommerce REST API, який робить можливим обмін даними між Інтернет-магазином та іншими інформаційними системами та програмними застосунками [3]. Це дозволяє, наприклад, розробляти мобільні клієнти Інтернет-магазинів з підтримкою функціоналу, що надає WooCommerce: перегляд та замовлення товарів покупцями, додання та оновлення характеристик товарів менеджерами, управління замовленнями покупців тощо. Дане дослідження присвячене проблемі проектування та створення програмної бібліотеки, використання якої підвищує ефективність та забезпечує безпеку типів при організації взаємодії програмних систем, створених для платформи Java, з Інтернет-магазинами на базі WooCommerce.

Аналіз останніх публікацій. Необхідно зазначити, що WooCommerce REST API – це фактично надбудова, яка використовує можливості WordPress REST API, додаючи власні кінцеві точки REST для товарів, замовлень, клієнтів, купонів тощо та реєструючи їх функціями WordPress [4]. Програмні бібліотеки для взаємодії з Інтернет-магазинами засобами WooCommerce REST API є корисними інструментами, які надають розробникам більш зручні та ефективні програмні засоби, аніж безпосереднє формування запитів. На сайті підтримки WooCommerce REST API [3] наведені

посилання на програмні бібліотеки для мов програмування JavaScript [5], Python [6], PHP [7] та Ruby [8]. Дослідження цих бібліотек показало, що вони не містять класів об'єктної моделі WooCommerce, а є "огортками" для формування REST-запитів. При використанні бібліотек такого типу дані об'єктів WooCommerce передаються у вигляді асоціативних масивів рядків "ключ-значення", а не як об'єкти моделі WooCommerce [9]. Такий підхід не є типобезпечним та погіршує можливості розширення функціоналу бібліотек для нового функціоналу WooCommerce. Окрім того, вказані бібліотеки працюють з WooCommerce REST API попередніх версій, на сьогодні актуальною є 3 версія WooCommerce REST API, яка вимагає використання WooCommerce версії, більше 3.5.x та WordPress версії більше 4.4 (на момент написання статті актуальними є версії 9.8.1 та 6.7.2, відповідно) [10]. Зауважимо, що посилання на бібліотеку для платформи Java на сайті підтримки WooCommerce REST API [3] відсутнє.

Постановка завдання. Метою цієї роботи було створення Java-бібліотеки, що спрощує інтеграцію з платформою електронної комерції WooCommerce через REST API шляхом реалізації об'єктної моделі основних сутностей платформи та підтримки типобезпечної роботи з даними.

Виклад основного матеріалу. Для реалізації мети дослідження на локальному комп'ютері було встановлене середовище для Веб-розробки WampServer, у ньому розгорнута система керування контентом WordPress, у якій встановлений плагін WooCommerce та створений Інтернет-магазин з тестовими товарами. Програмний код розроблявся у середовищі програмування IntelliJ IDEA із підтримкою технології побудови проєктів Apache Maven. Увесь напрацьований код є відкритим [11] та може бути вільно використаний на умовах ліцензії MIT.

В рамках цього дослідження були поставлені завдання розробити програмний код для отримання програмою-клієнтом на Java даних опублікованого в Інтернет-магазині товару, розміщення замовлення на цей товар та реєстрацію покупки Інтернет-магазину.

Для можливості комунікації зовнішніх програмних засобів з Веб-сервісами WooCommerce необхідно створити криптографічний ключ та пароль, які будуть ідентифікувати користувача програмного клієнта. Це можна зробити у налаштуваннях WooCommerce на вкладці *Advanced – REST API*. Зазначимо, що існує можливість створення ключа, який буде дозволяти тільки читання даних, та ключа, який буде дозволяти і читання, і створення даних у Інтернет-магазині, було обрано другий варіант. Було встановлено, що локальний сайт (localhost) не використовує криптографічно захищений протокол передачі даних між клієнтом та сервером (HTTPS), а Веб-сервіси WooCommerce передбачають саме його використання. Для дослідницького середовища вдалося обійти цю проблему за допомогою інструкції `SetEnv HTTPS on` у файлі налаштування Веб-сервера Apache `.htaccess`.

У якості засобу організації програмної взаємодії Java-клієнта з Веб-сервісами WooCommerce була обрана програмна бібліотека OkHttp [12], розроблена Square Open Source. Її вибір зумовлений підтримкою сучасних комунікаційних протоколів, широкою функціональністю та простотою використання [13], зокрема для передачі у HTTP-заголовку даних автентифікації запитів. Як рішення для серіалізації-десеріалізації об'єктів WooCommerce до файлу формату JSON (стандартний для WooCommerce формат серіалізації) було вирішено обрати бібліотеку Gson [14], розроблену Google, яка надає можливості налаштування користувацької серіалізації-десеріалізації об'єктів та демонструє найбільшу швидкість роботи серед конкурентів [15].

Архітектура програмної бібліотеки передбачає каталог `model` із класами, які відповідають об'єктам WooCommerce (Рис. 1). Цей каталог має підкаталоги `customer`, `order` та `product` із класами та перерахунками (англ. enumerations), які містять інформацію, що відноситься до відповідних об'єктів програмної моделі WooCommerce. Підкаталог `customer` містить клас-сутність `Customer` з атрибутами покупки Інтернет-магазину, а також перерахунки `CustomerRole` з можливими ролями (окрім покупки, WooCommerce передбачає також роль менеджера магазину) та клас `CustomerMetaData` з додатковими властивостями покупки. У підкаталозі `order` розміщені класи-сутності, пов'язані з замовленнями товарів WooCommerce – самим замовленням (`Order`) та інформацією про товар у замовленні (`ProductItem`). Окрім того, цей підкаталог містить перерахунки з підтримуваними WooCommerce валютами (`Currency`) та статусами обробки замовлення (`OrderStatus`). Застосування перерахунків є сталою практикою, що зменшує вірогідність помилок при використанні обмеженої кількості постійних значень та підтримує безпеку використання типів. У підкаталозі `product` розміщені класи-сутності, пов'язані

з товарами WooCommerce – сам товар (Product), категорія товару (ProductCategory), атрибут товару (ProductAttribute), та фізичні розміри товару (Dimensions). Окрім того, цей підкаталог містить перерахування з підтримуваними типами товарів (ProductType) та статусами публікації товару (ProductStatus). Каталог model, окрім підкаталогів, містить класи-сутності, що використовуються різними об'єктами моделі – інформацію про платіжні реквізити (Billing) та про адресу доставки (Shipping) товару, а також клас-сутність Image, який використовується для зображень в Інтернет-магазині.

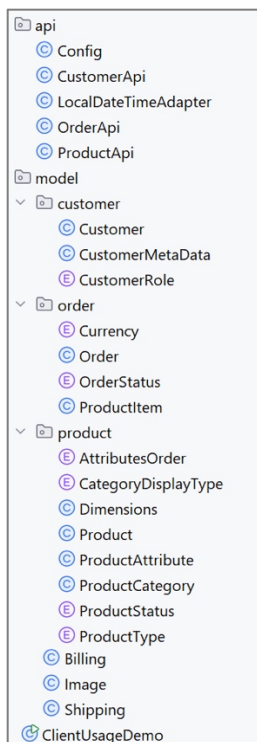


Рис. 1. Архітектура бібліотеки

При побудові моделі бібліотеки з метою повторного використання коду та можливості розширення функціоналу використовувалися технології об'єктно-орієнтованого проєктування: композиція об'єктів моделі, наприклад, використання об'єктів класу Image для товарів та категорій, та успадкування об'єктів – Billing є підкласом Shipping (Рис. 2).

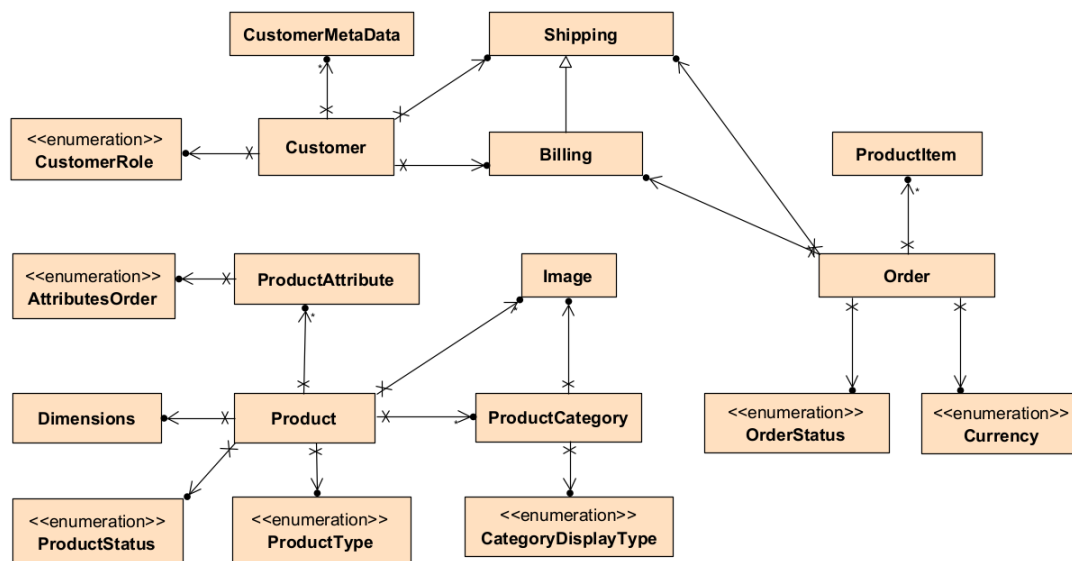


Рис. 2. UML діаграма класів моделі бібліотеки

Засобами програмної бібліотеки Project Lombok [16], окрім усунення з вихідного коду класів шаблонного коду, були організовані передбачені документацією Веб-сервісів WooCommerce доступи до атрибутів – тільки для читання або для читання та запису [9]. Також, враховуючи, що при створенні об'єктів встановлення усіх властивостей не є обов'язковим, у класах моделі були розроблені користувацькі методи-будівельники об'єктів (англ. builder) [17], наприклад:

```
import lombok.*;
import ua.edu.znu.wcrestapp.model.Billing;
import ua.edu.znu.wcrestapp.model.Shipping;
import java.time.LocalDateTime;

@Data
public class Customer {
    @Builder(builderMethodName = "customerBuilder")
    public static Customer of(String first_name, String last_name,
                             String email, String username, String password,
                             Billing billing, Shipping shipping) {
        Customer customer = new Customer();
        customer.setFirst_name(first_name);
        ...
        return customer;
    }

    @Setter(AccessLevel.NONE)
    private Long id;
    private String first_name;
    ...
    @Getter(AccessLevel.NONE)
    private String password;
    @Setter(AccessLevel.NONE)
    private CustomerRole role;
    @Setter(AccessLevel.NONE)
    private LocalDateTime date_created;
    ...
}
```

Використання методу-будівельника із частковим встановленням атрибутів та відсутністю можливості встановлення атрибутів тільки для читання може виглядати так:

```
Customer.customerBuilder()
    .last_name(lastName)
    .email(email)
    .username(username)
    .password(password)
    .build();
}
```

Для доступу до Веб-сервісів WooCommerce використовувалася базова адреса:

http://localhost/назва_сайту/wp-json/wc/v3,

що забезпечує роботу з актуальною 3 версією REST API, підтримуваною поточними версіями WordPress та WooCommerce.

Типи даних атрибутів сутностей більшою мірою співпадали з зазначеними у документації [10], але були й виключення, пов'язані з більш доцільними для обробки клієнтом типами, – наприклад, використання значень Double та Boolean замість рядка для значень цін та атрибутів з булевими значеннями, а також об'єктів java.time.LocalDateTime для дат та часу. Бібліотека Gson передбачає можливість створення для об'єктних типів користувацьких серіалізаторів та десеріалізаторів [18], що і було зроблено.

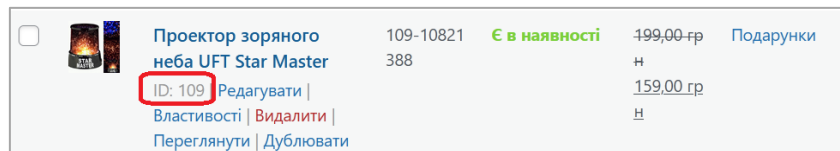
Архітектура розробленої бібліотеки передбачає каталог api (Рис. 1) з класами, що містять методи роботи з об'єктами моделі: CustomerApi, OrderApi, ProductApi, а також клас Config з методами конфігурування бібліотеки та користувацький адаптер серіалізації-десеріалізації.

Клас `CustomerApi` містить методи створення покупця, реєстрації його у магазині та метод отримання покупця за ідентифікатором. Клас `OrderApi` містить методи створення замовлення товару або декількох товарів та метод реєстрації замовлення у Інтернет-магазині. Клас `ProductApi` містить метод отримання товару за ідентифікатором. Всі методи, які формують REST-запити, додають до заголовків запитів криптографічний ключ та пароль, що містяться у текстовому файлі `credentials.properties` у підкаталозі ресурсів бібліотеки. Доступ до цього файлу організований засобами класу `java.util.Properties` стандартної бібліотеки Java.

Клас `Config` містить метод отримання базової адреси сайту Інтернет-магазину, яка зазначається у текстовому файлі `config.properties` у каталозі ресурсів бібліотеки, метод отримання криптографічного ключа та пароля з файлу-ресурсу `credentials.properties` та отримання об'єкта `Gson` з зареєстрованим класом-адаптером, що забезпечує користувацьку серіалізацію-десеріалізацію дати та часу.

У корні проєкту також міститься клас `ClientUsageDemo` з демонстрацією використання функціоналу бібліотеки. Код цього класу може бути взятий як прототип розробниками, що використовуватимуть бібліотеку.

Розроблені засоби бібліотеки були апробовані для отримання даних товару, опублікованого в Інтернет-магазині. Для визначення товару програмі передавався ідентифікатор товару у Інтернет-магазині. На Рис. 3 представлені властивості товару та отримані засобами бібліотеки дані цього товару у форматі JSON (виділені значення ідентифікатора товару, данні якого отримуються).



a)

```
C:\Program Files\Java\jdk-21\bin\java.exe" ...
Response Code: 200
{"id":109,"name":"\u041f\u0440\u043e\u0435\u043a\u0442\u043e\u0440 \u0437\u043e\u0440\u044f\u043d\u043e\u0433\u043e \u043d\u0435\u0431\u0430 UFT Star Master, slug=proektor-zoryanogo-neba-uft-star-master, permalink=https://localhost/edusite/f
<video class="wp-video-shortcode" id="video-0-1" width="870" height="489" preload="metadata" controls="controls"><source type="video/mp4" src="ht
<p>Магія зірок в приміщенні! Зображення різнобарвного зоряного неба прикрасить вашу кімнату і зробить її креативною. Завдяки цьому проєктору, зор
Працює від трьох батарейок типу AA.</p>
<p><strong>У нічника три режими роботи:</strong></p>
<ul>
<li>Проектується тільки білі візерунки.</li>
<li>Візерунки зірок плавно переливаються всіма кольорами.</li>
<li>Проектується білі візерунки + йде переливання різними кольорами.</li>
</ul>
```

6)

Рис. 3. Властивості товару Інтернет-магазину (а) та отримані дані у форматі JSON (б)

Також засобами бібліотеки було підготовлене замовлення товару та передане до Інтернет-магазину. На Рис. 4 представлені дані замовлення у програмі та зареєстроване замовлення у Інтернет-магазині.

```
public Order prepareOrder(ProductItem[] products) {
    Shipping shipping = Shipping.builder()
        .first_name("John")
        .last_name("Doe")
        ...
        .build();

    ...
    Order order = Order.orderBuilder()
        .line_items(products)
        .billing(billing)
        ...
        .build();
    return order;
}
```

a)

☐ Замовлення	Дата	Статус	Загалом
☐ #20 John Doe	53 хвилини тому	В обробці	159,00 грн

б)

Рис. 4. Дані замовлення товару (а) та зареєстроване у Інтернет-магазині замовлення (б)

Також засобами бібліотеки була виконана реєстрація покупця Інтернет-магазину. На Рис. 5 представлені дані покупця у програмі та його реєстрація у Інтернет-магазині.

```
CustomerApi customerApi = new CustomerApi();
String firstName = "Марія";
String lastName = "Шевченко";
String email = "maria@gmail.com";
String username = "maria";
...
return Customer.customerBuilder()
    .first_name(firstName)
    .last_name(lastName)
    .email(email)
    .username(username)
    .password(password)
    .billing(billing)
    .shipping(shipping)
    .build();
```

а)

Ім'я	Логін	Остання активність	Дата реєстрації	Email	Замовлення	Загальна сума витрат	Середнє
Марія Шевченко	maria	30.03.2025	30.03.2025	maria@gmail.com			5

б)

Рис. 5. Дані покупця Інтернет-магазину (а) та зареєстрований у Інтернет-магазині покупець (б)

Апробація продемонструвала ефективність використання бібліотеки, яка полягала у зручності та безпеці типів на рівні компіляції – використанні клієнтською кодовою базою методів, що приймають як аргументи і працюють з об'єктами моделі WooCommerce, наприклад, `public Customer createCustomer(Customer customer, OkHttpClient client, Gson gson, Config config)`, `public Order prepareOrder(ProductItem[] products)` тощо. У разі використання програмних бібліотек, що представляють собою огортки [4-7], вхідні дані передаються як рядки [8], що ускладнює їх використання та з більшою вірогідністю може привести до помилок.

Висновки та перспективи подальших досліджень. В результаті проведення цього дослідження спроектована модель класів, які відповідають сутностям платформи електронної комерції WooCommerce. На основі цієї моделі розроблена програмна бібліотека, що дозволяє працювати з об'єктами WooCommerce клієнтському коду на Java. Бібліотека містить засоби роботи з товарами, замовленнями та покупцями. У подальшому планується розширення бібліотеки для підтримки роботи з варіативними товарами та купонами.

Список бібліографічного опису

1. Historical trends in the usage statistics of content management systems. URL: https://w3techs.com/technologies/history_overview/content_management/all (access date: 15.05.2025).
2. Lisa Sims Building Your Online Store with WordPress and WooCommerce : монографія / L. Sims. – Conyers : APress, 2018. – 190 p.
3. WooCommerce REST API. URL: <https://woocommerce.com/document/woocommerce-rest-api> (access date: 15.05.2025).

4. Sufyan bin Uzayr Learning WordPress REST API. Packt Publishing, 2016. 209 p.
5. WooCommerce REST API - JavaScript Library. URL: <https://github.com/woocommerce/woocommerce-rest-api-js-lib> (access date: 15.05.2025).
6. A Python wrapper for the WooCommerce REST API. URL: <https://pypi.org/project/WooCommerce> (access date: 15.05.2025).
7. A PHP wrapper for the WooCommerce REST API. URL: <https://packagist.org/packages/automattic/woocommerce> (access date: 15.05.2025).
8. WooCommerce API - Ruby Client. URL: <https://github.com/woocommerce/wc-api-ruby> (access date: 15.05.2025).
9. Libraries (Official) to Work with WooCommerce REST API and Examples. URL: <https://rudrastyh.com/woocommerce/libraries-to-work-with-rest-api.html> (access date: 15.05.2025).
10. Technical Documentation for the WooCommerce REST API. URL: <https://woocommerce.github.io/woocommerce-rest-api-docs> (access date: 15.05.2025).
11. WooCommerceRESTClientApp. Репозитарій проєкту Java-клієнта для REST Веб-сервісів платформи електронної комерції WooCommerce. URL: <https://github.com/kolgenpav/WooCommerceRESTClientApp> (дата звернення: 15.05.2025).
12. OkHttp. Офіційний сайт. URL: <https://square.github.io/okhttp> (access date: 15.05.2025).
13. Tom Akehurst Which Java HTTP client should I use in 2024? . URL: <https://www.wiremock.io/post/java-http-client-comparison> (access date: 15.05.2025).
14. Gson. Офіційний сайт. URL: <https://github.com/google/gson> (access date: 15.05.2025).
15. Nick Andrews The Ultimate JSON Library: JSON.simple vs. GSON vs. Jackson vs. JSONP. URL: <https://dzone.com/articles/the-ultimate-json-library-jsonsimple-vs-gson-vs-ja> (access date: 15.05.2025).
16. Project Lombok. Офіційний сайт. URL: <https://projectlombok.org> (access date: 15.05.2025).
17. Using Lombok's @Builder Annotation. URL: <https://www.baeldung.com/lombok-builder> (access date: 15.05.2025).
18. Gson User Guide. URL: <https://github.com/google/gson/blob/main/UserGuide.md#custom-serialization-and-deserialization> (access date: 15.05.2025).

References

1. Historical trends in the usage statistics of content management systems. URL: https://w3techs.com/technologies/history_overview/content_management/all (access date: 15.05.2025).
2. Lisa Sims Building Your Online Store with WordPress and WooCommerce : монографія / L. Sims. – Conyers : APress, 2018. – 190 p.
3. WooCommerce REST API. URL: <https://woocommerce.com/document/woocommerce-rest-api> (access date: 15.05.2025).
4. Sufyan bin Uzayr Learning WordPress REST API. Packt Publishing, 2016. 209 p.
5. WooCommerce REST API - JavaScript Library. URL: <https://github.com/woocommerce/woocommerce-rest-api-js-lib> (access date: 15.05.2025).
6. A Python wrapper for the WooCommerce REST API. URL: <https://pypi.org/project/WooCommerce> (access date: 15.05.2025).
7. A PHP wrapper for the WooCommerce REST API. URL: <https://packagist.org/packages/automattic/woocommerce> (access date: 15.05.2025).
8. WooCommerce API - Ruby Client URL: <https://github.com/woocommerce/wc-api-ruby> (access date: 15.05.2025).
9. Libraries (Official) to Work with WooCommerce REST API and Examples. URL: <https://rudrastyh.com/woocommerce/libraries-to-work-with-rest-api.html> (access date: 15.05.2025).
10. Technical Documentation for the WooCommerce REST API. URL: <https://woocommerce.github.io/woocommerce-rest-api-docs> (access date: 15.05.2025).
11. WooCommerceRESTClientApp. Java client project repository for REST Web services of the WooCommerce e-commerce platform.. URL: <https://github.com/kolgenpav/WooCommerceRESTClientApp> (access date: 15.05.2025).
12. OkHttp. Official site. URL: <https://square.github.io/okhttp> (access date: 15.05.2025).
13. Tom Akehurst Which Java HTTP client should I use in 2024? URL: <https://www.wiremock.io/post/java-http-client-comparison> (access date: 15.05.2025).
14. Gson. Official site. URL: <https://github.com/google/gson> (access date: 15.05.2025).
15. Nick Andrews The Ultimate JSON Library: JSON.simple vs. GSON vs. Jackson vs. JSONP URL: <https://dzone.com/articles/the-ultimate-json-library-jsonsimple-vs-gson-vs-ja> (access date: 15.05.2025).
16. Project Lombok. Official site. URL: <https://projectlombok.org> (access date: 15.05.2025).
17. Using Lombok's @Builder Annotation. Retrieved from: <https://www.baeldung.com/lombok-builder> (access date: 15.05.2025).
18. Gson User Guide. URL: <https://github.com/google/gson/blob/main/UserGuide.md#custom-serialization-and-deserialization> (access date: 15.05.2025).