



**Клієнтський сайт WebAPI сервісів департаменту торгівлі цінними
паперами**

**Текстова частина до курсової роботи
за спеціальністю «Інженерія програмного забезпечення»- 121**

Керівник курсової роботи

Кандидат фізико-математичних наук,

Сініцина Р.В.

_____ (підпис)

“ ____ ” _____ 2020 р.

Виконала студентка ІІЗ-3

Бутенко І.С.

“ ____ ” _____ 2020 р.

Київ 2020

Календарний план виконання роботи:

№	Назва етапу	Термін виконання	Примітка
1 .	Отримання теми курсової	09.10.2019	
2 .	Пошук тематичної наукової літератури	20.10.2019	
3 .	Ознайомлення з науковою літературою	11.12.2019	
4 .	Визначення можливостей клієнта	25.12.2019	
5 .	Створення макету сайту	01.01.2020	
6 .	Визначення структури програми	05.02.2020	
7 .	Написання першої частини курсової роботи	1.04.2020	
8 .	Написання другої частини курсової роботи	15.04.2020	
9 .	Написання висновків курсової роботи	16.04.2020	
1 0 .	Перегляд змісту роботи з керівником	17.04.2020	
1 1 .	Внесення змін до роботи	18.04.2020	
1 2 .	Створення презентації	19.04.2020	
1 3 .	Завантаження курсової роботи	19.04.2020	

Тема: Клієнтський сайт WebAPI сервісів департаменту торгівлі цінними паперами

Зміст

У

Календарний план виконання роботи:.....	2
Зміст.....	3
Вступ.....	4
РОЗДІЛ 1. Структура веб-сервісу і його архітектура.....	5
1.1. Опис потрібного результату.....	5
1.2 Види цінних паперів.....	7
Атрибути сутностей.....	8
РОЗДІЛ 2. Структура веб-сервісу і його архітектура.....	10
2.1 Визначення Web-API.....	10
2.2 REST-підхід.....	11
2.3 Побудова запитів.....	12
2.4 Тіло запиту та формат JSON.....	13
2.5 Приклади використання HTTP-запитів.....	14
Фільтрація.....	17
РОЗДІЛ 3. Розробка клієнтської частини.....	18
3.1 Зовнішній вигляд сайту.....	18
3.2 Використані матеріали.....	20

Вступ

На сьогоднішній день обсяги торгів цінними паперами на біржовому ринку постійно зростають, що вказує на актуальність даного сектору у фінансовій сфері країни. Так, наприклад, обсяги торгів за січень 2020 року перевищили 33 мільярди гривень.

Особливістю торгівлі на біржі є неможливість взаємодії безпосередньо між покупцем та продавцем: потрібен посередник, що встановлює цей зв'язок. Зазвичай цим посередником виступає брокерська установа (компанія), що має державну ліцензію та може здійснювати громадсько-правові угоди не тільки від імені клієнта, а і за його рахунок.

Для здійснення купівлі чи продажу кожному новому клієнту брокер має створити окремий торговий рахунок. Окрім цього, брокер зобов'язаний слідкувати за ходом виконання кожної угоди та операцій на рахунках клієнтів.

Основна мета даної роботи є створення сайту для брокерської компанії, що допоможе її працівникам оптимізувати процес реєстрації клієнтів, відкриття нових рахунків та подальші операції з купівлі чи продажу цінних паперів на Українській біржі з відслідковуванням процесу виконання угод.

Перший розділ роботи присвячено розробці та аналізу структури системи, можливостей клієнта в залежності від його статусу, зв'язок клієнта та сервера і самого сайту, визначенню потрібного функціоналу веб-сервісу для подальшої реалізації. Окрім цього в цьому розділі обґрунтовано доцільність використання архітектурного підходу REST при побудові запитів, вибір інструментарію для розробки самого сайту.

У другому розділі наводяться приклади побудови запитів для проведення торгових операцій відносно рахунку клієнта та фондової біржі в залежності від типу цінного паперу та отримання відповідної відповіді ці запити та їх обробка.

Третій розділ присвячено інтерфейсу продукту з відповідно наведеними ілюстраціями та демонстрацією вигляду сайту в залежності від типу користувача.

Реалізація клієнтської частини веб-сервісу для проведення купівлі-продажу цінних паперів на біржовому ринку.

Вимоги:

1. Розподіл можливостей залежно від типу користувача
2. Визначення принципів взаємодії клієнта з сервером
3. Визначення архітектури серверу
4. Побудова запитів та обробка відповідей на них відповідно до типу цінного паперу
5. Реалізація Web-API за допомогою HTTP-запитів .
6. Визначення інструментарію для створення клієнтського web-застосунку
7. Реалізація сайту компанії

РОЗДІЛ 1. Структура веб-сервісу і його архітектура

1.1. [Опис потрібного результату](#)

Результатом цієї роботи є веб-сайт , на меті якого оптимізація роботи кожного з брокерів та її автоматизованість. Для створення інтерфейсу сайту використовується вільний набір інструментів Bootstrap (CSS -фреймворк) , що включає в себе набір шаблонів , побудованих за допомогою HTML,CSS,SASS та мови сценаріїв Javascript. Bootstrap ,окрім цього, забезпечує сайту адаптивність при зміні розміру браузерного вікна чи перегляді на мобільних пристроях.

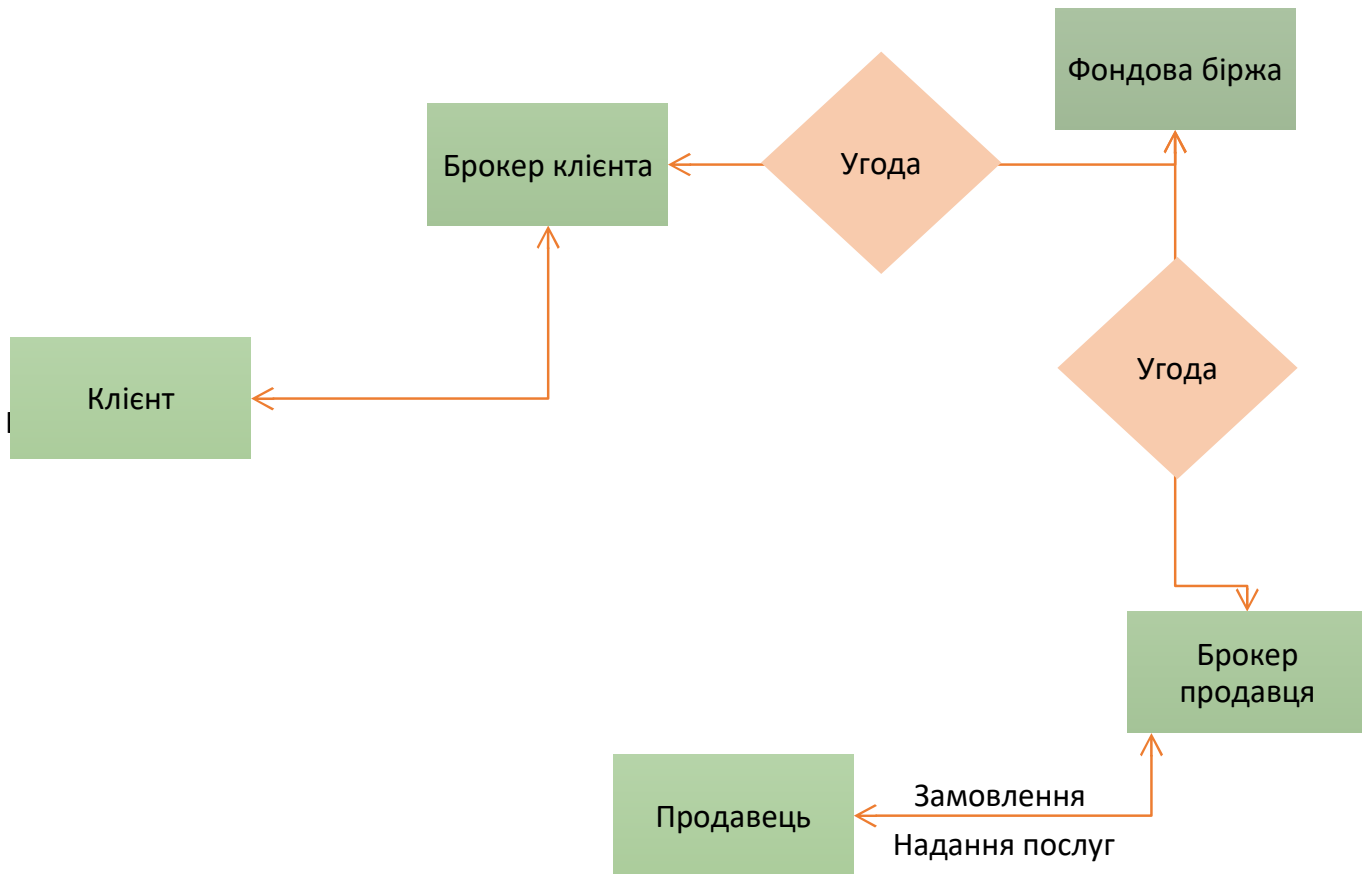


Рисунок 1.1

Принципи функціонування даної брокерської компанії схематично зображено на *рисунку 1.1*. Клієнт, що вже відкрив рахунок в компанії, може звернутися брокера для того, щоб придбати певний вид цінного паперу, від певної юридичної чи фізичної особи, а останній в свою чергу ,маючи право діяти від імені клієнта ,звертається до фондової біржі ,де брокер продавця виставляє потрібний документ для продажі за замовленням продавця.

Користувач повинен мати можливість до автентифікації за допомогою логіна та пароля. Основні можливості користувача : реєстрація нового користувача , тобто відкриття на нього рахунку , поповнення цього рахунку , операції купівлі або продажу цінних паперів від свого імені , але за рахунок клієнта.

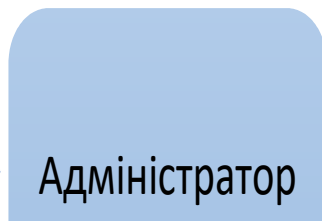
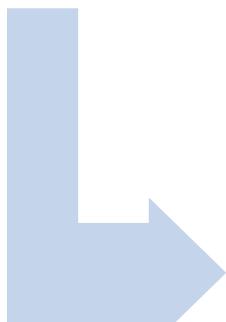
Сайт повинен мати два режими прав доступу : адміністратор та звичайний користувач (брокер). Адміністратор – це брокер , який має привілейовані права : він має можливість додатково редагувати основні дані клієнта , видаляти дані у разі того, якщо клієнт вже відмовився від послуг даної компанії. Окрім цього, адміністратор має права продивлятися статистику кожного брокера.

Брокери можуть спілкуватися з адміністратором через внутрішній чат і навпаки.

Рівні доступів та розподіл прав для продемонстровано на *рисунку 1.2*.



реєстрація нових клієнтів (створення рахунків)
операції купівлі/продажу цінних паперів
відслідковування за процесом укладання угоди
отримання статистики угод від свого імені



редагування/видалення інформації та даних клієнта
отримання статистики кожного брокера
можливість корекції даних про персонал)

Рисунок 1.2



відміна угоди, що ще не відбулася

1.2 Види цінних паперів

Цінні папери є документами, що беруть участь у операціях купівлі/продажу і є результатом угоди сторін. В даній компанії існують два види цінних паперів: акція і облігація.

Акція – це цінний папір, що встановлює право на володіння частиною майна компанії-емітента. Акції діляться на дві групи: привілейовані та звичайні. Вартість привілейованих визначається вартістю звичайної акції на коригуючий коефіцієнт.

Облігація – цінний папір, що підтверджує, що її власник зробив фінансовий внесок в капітал емітента. Компанія зобов'язується відшкодувати ці кошти в обумовлений термін.

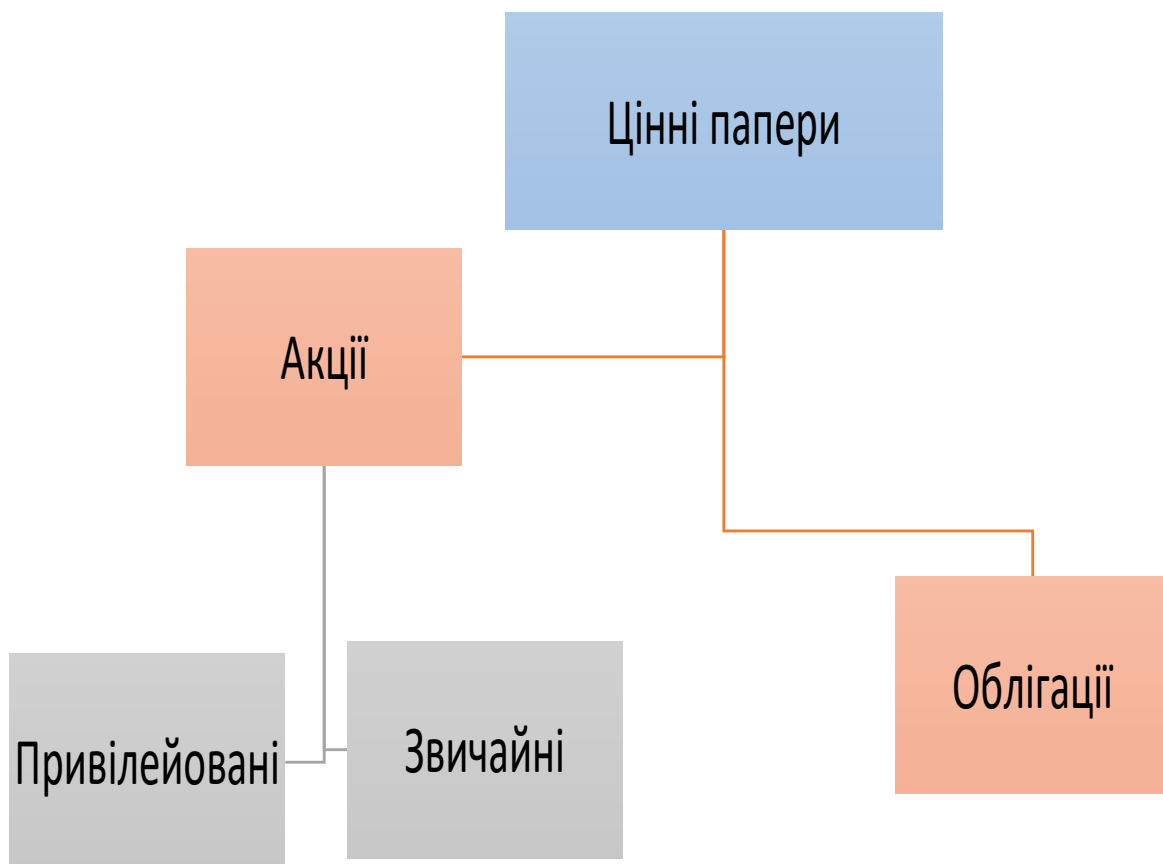


Рисунок 1.3

Атрибути сутностей

Договір між брокером і клієнтом створюється при реєстрації клієнта та створенні його торгового рахунку.

Атрибути договору:

- Ідентифікаційний номер (первинний ключ)
- Прізвище
- Ім'я

- По-батькові
- Серія і номер паспорта або номер документа
- Місце народження
- Дата видачі паспорта
- Адреса реєстрації
- Місце роботи
- Номер телефону
- Електронна адреса

Додавати до системи нового брокера має право тільки адміністратор.
Атрибути при додаванні нового брокера:

- Логін
- Пароль
- Прізвище
- Ім'я
- По-батькові
- Статус (адміністратор чи звичайний брокер)

Угода є результатом при продажі або купівлі цінних паперів.

Атрибути угоди

- Унікальний номер
- Тип операції (продаж чи купівля)
- Тип цінного паперу
- Дата угоди
- Статус
- Вартість

При купівлі на рахунку клієнта повинно бути достатньо коштів.

Фільтрація

Список клієнтів та їх угод може бути відфільтрований таким чином:

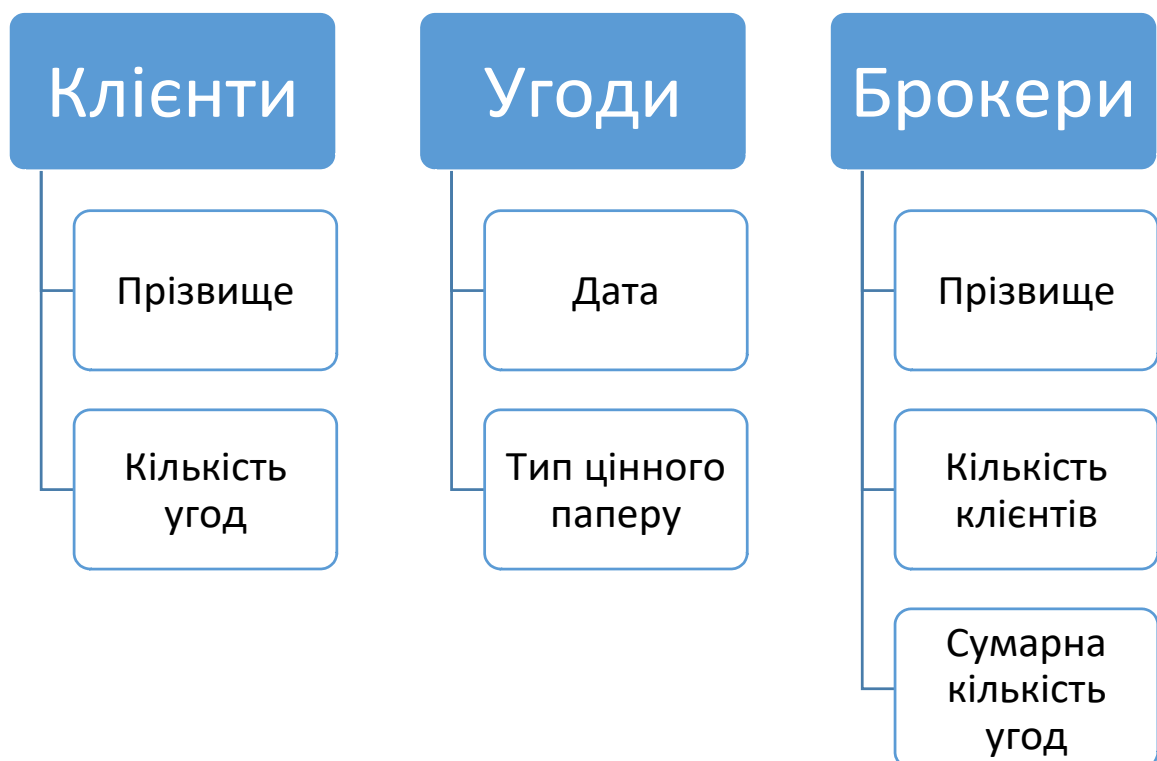
- За датою угод (за зростанням чи спаданням)
- За датою створення рахунків клієнтів (за зростанням чи спаданням)

- За типом цінного паперу угоди(за зростанням чи спаданням)
- За прізвищем клієнта (за зростанням чи спаданням)
- За активністю клієнта (тобто кількістю складених угод) (за зростанням чи спаданням)

Адміністратори можуть відфільтрувати список брокерів за такими критеріями:

- Прізвище брокера(спадання або зростання)
- Кількість клієнтів брокера (спадання або зростання)
- Кількість складених угод(спадання або зростання)
- Прізвище брокера(спадання або зростання)
- Кількість клієнтів брокера (спадання або зростання)
- Кількість складених угод(спадання або зростання)

Фільтрація



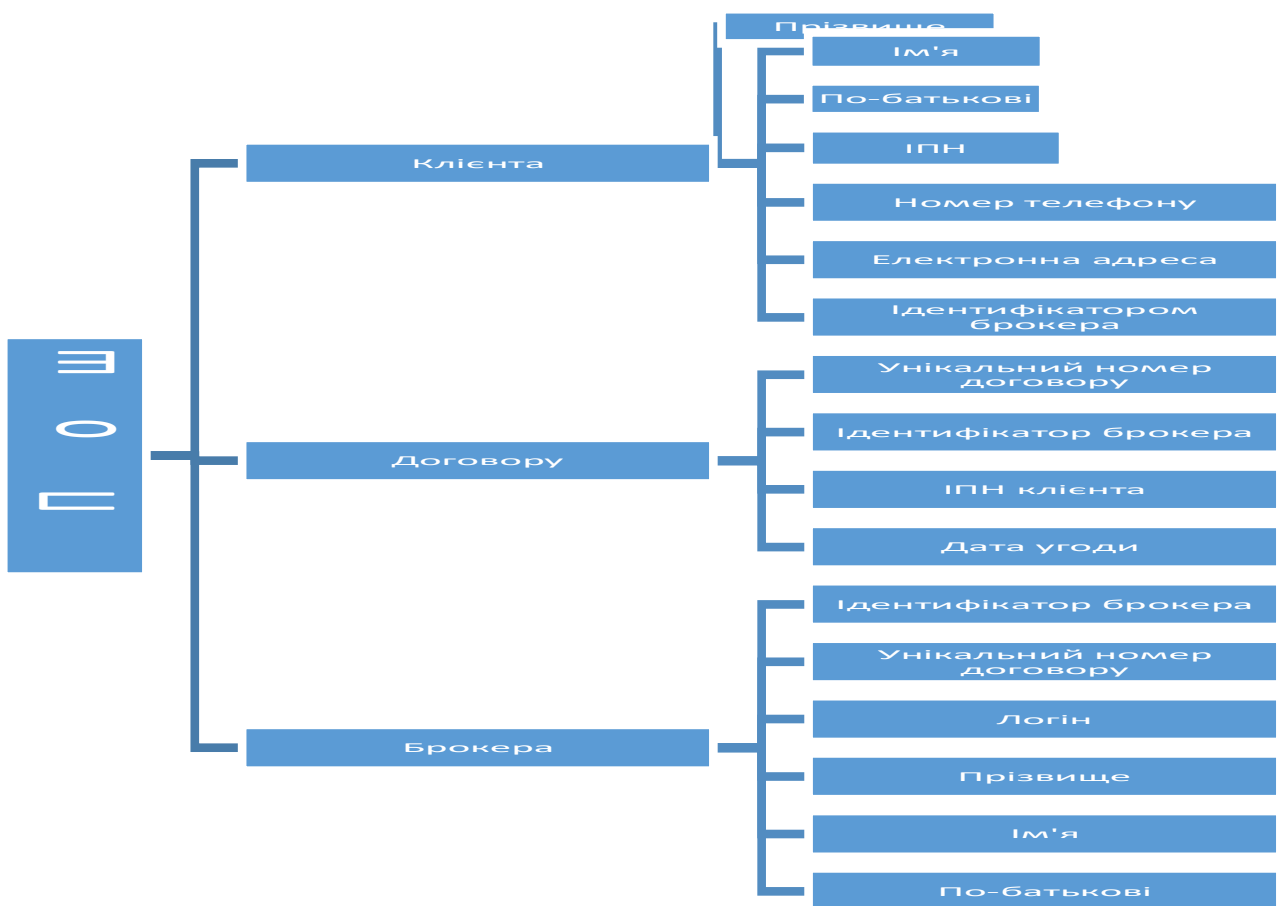


Рисунок 1.3

РОЗДІЛ 2. Структура веб-сервісу і його архітектура

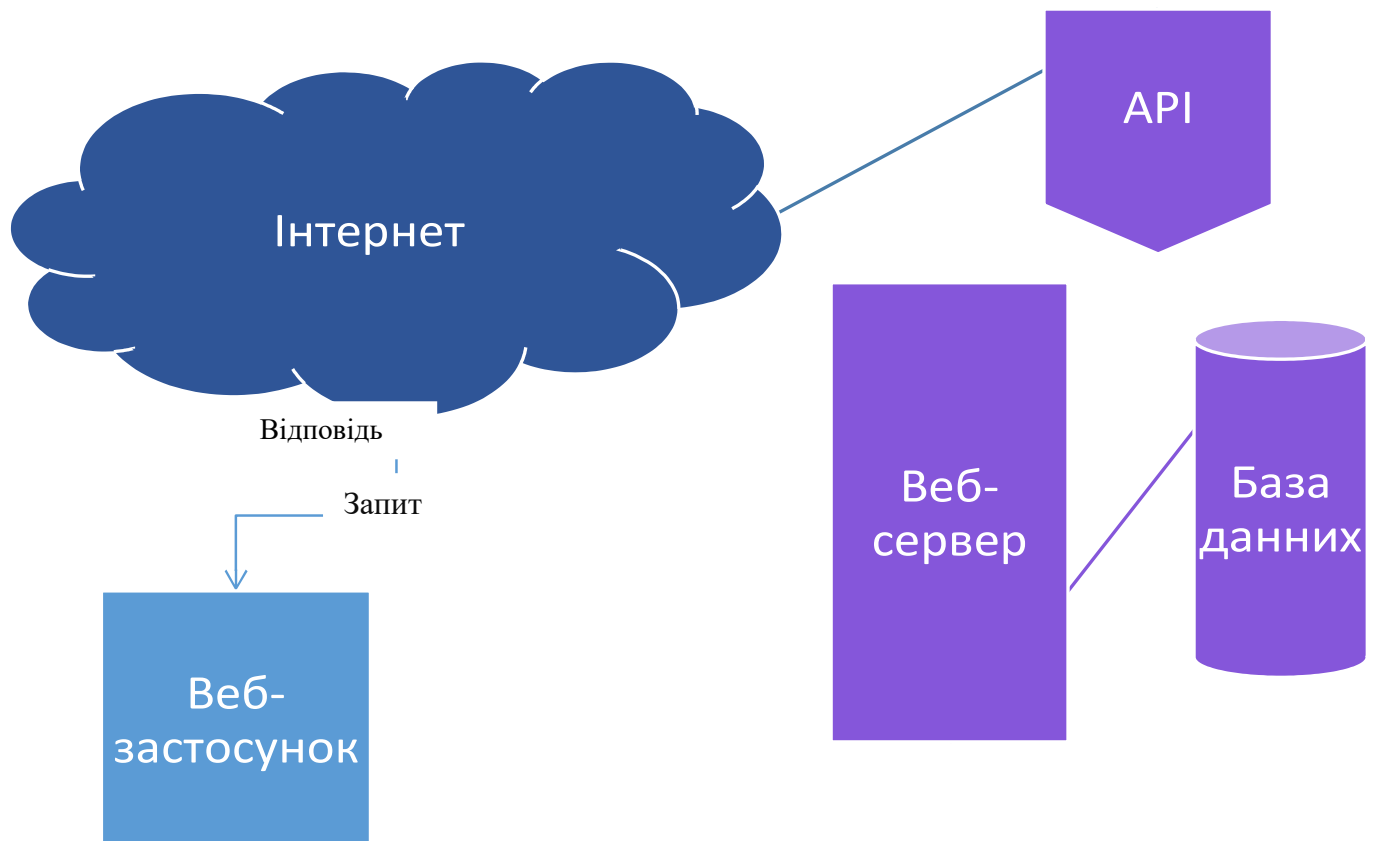
2.1 Визначення Web-API

Прикладний програмний інтерфейс (інтерфейс програмування застосунків, інтерфейс прикладного програмування) (*Application Programming Interface* або *API*) – набір певних правил, протоколів, що дозволяють програмам та підпрограмам взаємодіяти між собою.

API надає можливість розробнику використовувати вже готові об'єкти іншого програмного забезпечення через означені в ньому правила взаємодії, що полегшує розробку застосунку.

Web API створюється на сервері, куди потім клієнт зможе надсилати HTTP-запити та отримувати відповідно на них відповіді для безпосередньої взаємодії. Повідомлення можуть бути різних форматів, але формати XML та JSON використовуються частіше.

Для демонстрації принципу роботи наданий *рисунок 1.4*.



Таблиця 2.1

2.2 REST-підхід

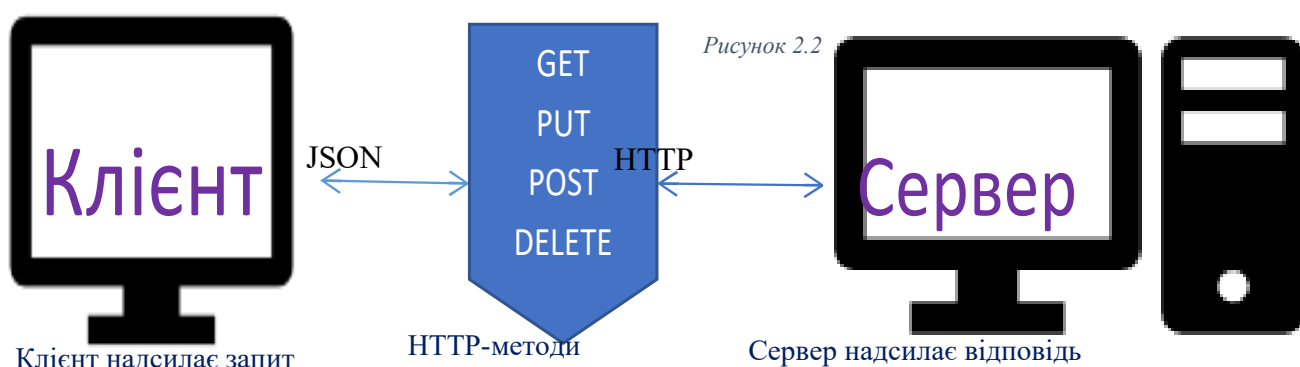
REST – це архітектурний підхід , який визначає ,яким буде API. Його правила були використані для створення цього застосунку.

Головні чотири принципи підходу REST : явне використання HTTP-методів, відсутність збереження станів, унікальний ідентифікатор URI для кожної з сутностей ,передача даних в XML, JavaScript Object Notation (JSON).

Перша складова до реалізації REST – ресурси . Даний веб-застосунок має адресу <http://localhost:88>) . Наприклад, наш сервер має записи про кожного клієнта та його рахунок . Щоб отримати доступ до перегляду даних клієнта з ідентифікатором 1, потрібно виконати команду <http://localhost:88/clients/1>.

Друга складова REST – методи запитів HTTP , а саме GET,POST,DELETE і PUT. За допомогою методу GET можна додати новий запис,POST -для отримання ресурсів, DELETE- для видалення запису і PUT- для зміни або оновлення записів.

Принцип взаємодіє між клієнтом та сервером у REST-підході показано на *рисунок 2.2*



2.3 Побудова запитів

Спершу треба зрозуміти саму структуру запиту (див. *рисунок 2.3*).

Структура запиту

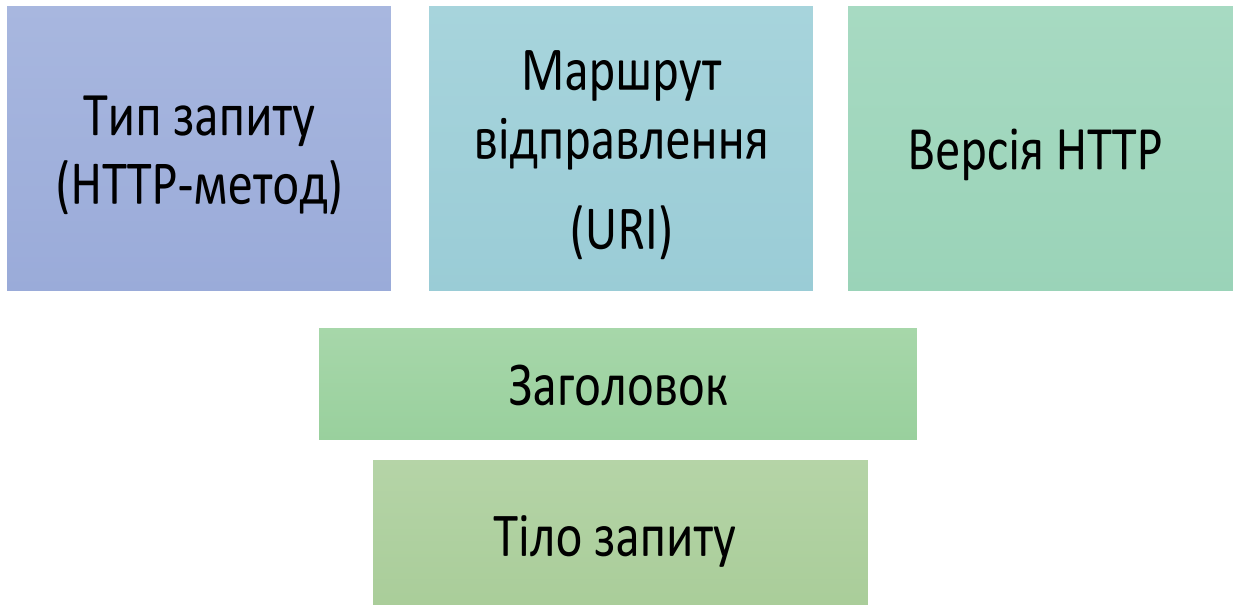


Рисунок 2.3

- Тип запиту визначає один з HTTP- методів (GET , POST ,PUT, DELETE);
- Маршрут відправлення (URI) – це адреса ,куди буде надіслано запит;
- HTTP-версія -визначає версію протоколу HTTP (зазвичай 1.1);
- Заголовок забезпечує метайнформацію про запит , наприклад, дата надсилання клієнтом цього запиту і розмір його тіла;
- Тіло повідомлення – це набір даних , які і передаються в запиті.

2.4 Тіло запиту та формат JSON

JSON (JavaScript Object Notation) – стандартний текстовий формат для представлення структурованих даних на основі JavaScript. Цей формат зазвичай використовується для створення веб-застосунків, а саме для передачі даних , наприклад, від сервера до клієнта.

JSON є рядком , синтаксис якого схожий на об'єкт JavaScript у буквенному форматі, окрім цього,JSON є набором пар *ключ-значення*, наприклад, якщо треба визначити посаду працівника як адміністратор при додавання в систему , у JSON-структурі вказується :

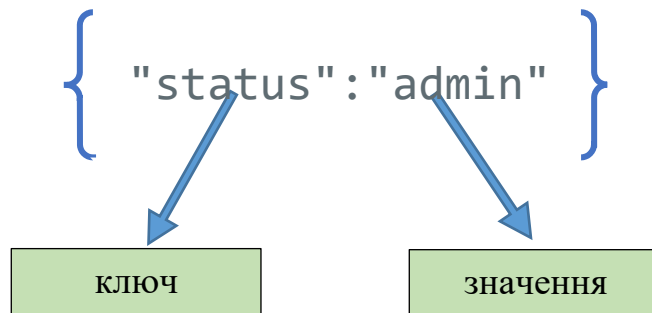


Рисунок 2.4

Для того, щоб сервер сприймав формат JSON при отриманні запиту, в заголовку повинно бути вказано, що формат вмісту є JSON:

```
Content-type: application/json
```

JSON був обраний через те, що він має вбудовану підтримку у найпопулярніших браузерах, починаючи з 2009 року, є простим і читабельним та найбільш розповсюдженим.

2.5 Приклади використання HTTP-запитів

У даному веб-застосунку метод POST використовується для реєстрації нового брокера та при додавання нового клієнта до системи разом із створенням його торгового рахунку.

Метод GET призначений для отримання інформації про самого працівника, його клієнтів, кількість купівлі/продажу, статистику, а також і про статистику цієї компанії в цілому.

Метод PUT використовується для оновлення інформації щодо брокера (пароль, логін), реквізитів клієнта та інших його даних.

Метод DELETE визначає видалення клієнта з системи або брокера при його звільненні. Цей метод використовується також при відміні ще не затвердженої угоди купівлі/продажу.

Треба зазначити, що при видаленні інформації про клієнта, видаляються всі його угоди та будь-які інші його згадування, адже компанія зацікавлена в купівлі/продаж тільки теперішніх клієнтів. При видаленні брокера інформація про його клієнтів зберігається.

На *рисунку 2.5* наведено використання усіх прикладів використання HTTP-методів.

Назва HTTP-методу	Ресурс	Опис прикладу
POST	/workers	Зареєструвати нового працівника
GET	/workers /workers/<id>	Отримати список всіх працівників Отримати інформацію про брокера з ідентифікатором id
PUT	/workers/<id>	Обновити інформацію про брокера з ідентифікатором id
DELETE	/workers/<id>	Видалити всю інформацію про брокера з ідентифікатором id

Рисунок 2.5

Назва HTTP-методу	Ресурс	Опис прикладу
POST	/clients	Зареєструвати нового клієнта
GET	/clients /clients/<id>	Отримати список всіх клієнтів Отримати інформацію про клієнта з ідентифікатором id
PUT	/clients/<id>	Обновити інформацію про клієнта з ідентифікатором id
DELETE	/clients/<id>	Видалити всю інформацію про клієнта

з ідентифікатором id

Лише адміністратор має право на вивід інформації про всіх клієнтів та їх угоди, про їх брокерів та на будь-які операції редагування чи видалення.

Назва HTTP-методу	Ресурс	Опис прикладу
POST	/workers/<id_staff>/clients	Зареєструвати нового клієнта працівником з ідентифікатором id_staff
GET	/workers/<id_staff>/clients /workers/<id_staff>/clients/<id_client>	Отримати список всіх клієнтів працівника з ідентифікатором id_staff Отримати інформацію про клієнта з ідентифікатором id_client працівника з ідентифікатором id_staff
PUT	/workers/<id_staff>/clients/<id_client>	Оновити інформацію про клієнта з ідентифікатором id_client працівника з ідентифікатором id_staff
DELETE	/workers/<id_staff>/clients/<id_client>	Видалити всю інформацію про клієнта з ідентифікатором id_client працівника з ідентифікатором id_staff

РОЗДІЛ 3. Розробка клієнтської частини

3.1 Зовнішній вигляд сайту

Для створення сайту були використані HTML, CSS, JavaScript та Bootstrap .
Bootstrap -це набір інструментів (HTML, CSS, JavaScript) для полегшення створення сайту.

Клієнти						
>						
Show 10 entries						Search: <input type="text"/>
Прізвище	Ім'я	По-батькові	Відповідальний	Кількість угод	Коштів на рахунку	
Валевська	Оксана	Геннадіївна	rozner	10	490000	
Діденко	Валерій	Олександрович	sergienko	8	100000	
Козаченко	Сергій	Борисович	nevirenko	5	200000	
Щербачов	Валентин	Сергійович	nevirenko	13	600000	
Прізвище	Ім'я	По-батькові	Відповідальний	Кількість угод	Коштів на рахунку	

Ось наведена таблиця з клієнтами компанії як приклад таблиці створеної за допомогою Bootstrap.

Login

Логін

jnjnjn

Пароль

.....

Запам'ятати пароль

>

Show 10 entries

Search: Оксана

Прізвище ↑↓	Ім'я ↑↓	По-батькові ↑↓	Відповідальний ↑↓	Кількість угод ↑↓	Коштів на рахунку ↑↓
Валевська	Оксана	Геннадіївна	pozner	10	490000
Прізвище	Ім'я	По-батькові	Відповідальний	Кількість угод	Коштів на рахунку

3.2 Використані матеріали

- <http://finpost.com.ua/news/16418>
- <https://pfts.ua/>

- https://bankchart.com.ua/finansii/investment/statti/vtorinniy_rinok_tsinnih_paperiv_zagalniy_opis_struktura_uchasniki_mehanizm_roboti
- <https://searchapparchitecture.techtarget.com/definition/RESTful-API>
- <https://bizzapps.ru/b/rest-api/>
- <https://www.tutorialsteacher.com/webapi/what-is-web-a>
- <https://www.slideshare.net/KishrorKumar/getting-started-with-wp-rest-api>
- <https://flylib.com/books/en/2.320.1.30/1>
- <https://www.smashingmagazine.com/2018/01/understanding-using-rest-api/>
- <https://zapier.com/learn/apis/chapter-2-protocols/>
- <https://developer.mozilla.org/ru/docs/Learn/JavaScript/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D1%8B/JSON>
- <https://ffin.ua/ru/service/prodadg-derdgavnykh-obligatsiy>
- <https://biznesat.com/finansy/97-aktsiji-shcho-tse-take.html>