



ПРОЄКТУВАННЯ ТА РОЗРОБКА API В СЕРВІС-ОРІЄНТОВАНИХ СИСТЕМАХ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	122 Комп'ютерні науки
Освітня програма	Цифрові технології в енергетиці
Статус дисципліни	Вибіркова
Форма навчання	Очна (денна)
Рік підготовки, семестр	3 курс, весняний семестр
Обсяг дисципліни	5 кред./150 год.: 36 год. лекцій, 18 год. лабораторних занять, 96 год. СРС
Семестровий контроль/ контрольні заходи	МКР, залік
Розклад занять	https://schedule.kpi.ua/
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: к.т.н., доцент каф. ЦТЕ, Тихоход Володимир Олександрович. Лабораторні роботи: к.т.н., доцент каф. ЦТЕ, Тихоход Володимир Олександрович
Розміщення курсу	https://campus.kpi.ua

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Дисципліна присвячена принципам проєктування, реалізації, документування, тестування та супроводу програмних інтерфейсів як засобів взаємодії сервісів у сучасних розподілених програмних системах. Предметом вивчення є моделі сервісної взаємодії, архітектурні стилі програмних інтерфейсів, протокол HTTP, ресурсне моделювання, контрактний опис інтерфейсів, обробка помилок, безпека, версіювання, тестування, забезпечення продуктивності та надійності, а також взаємодія сервісів у мікросервісних і подієво-орієнтованих системах.

Метою дисципліни є формування здатності проєктувати та реалізовувати програмні інтерфейси сервісів, обґрунтовано обирати моделі й засоби сервісної взаємодії, формувати стабільні контракти, забезпечувати безпеку, надійність, сумісність та контрольовану еволюцію програмних інтерфейсів.

У результаті вивчення дисципліни у студентів мають сформуватися такі **компетентності**:

загальні:

- здатність до абстрактного мислення, аналізу та синтезу;
- здатність застосовувати знання у практичних ситуаціях;
- здатність вчитися та оволодівати сучасними знаннями;

- здатність до пошуку, оброблення та аналізу інформації з різних джерел;
- здатність працювати в команді та приймати обґрунтовані рішення.

фахові:

- здатність проектувати програмні системи з чітким розподілом відповідальності між сервісами та їхніми інтерфейсами;
- здатність моделювати ресурси, операції та контракти взаємодії відповідно до семантики протоколів і архітектурних принципів;
- здатність реалізовувати механізми автентифікації, авторизації, обробки помилок, версіювання та документування програмних інтерфейсів;
- здатність забезпечувати надійність сервісної взаємодії з урахуванням ідемпотентності, повторних запитів, обмежень часу очікування та часткових відмов;
- здатність тестувати програмні інтерфейси та оцінювати їхню якість, продуктивність, сумісність і придатність до розвитку.

Очікувані результати навчання:

- визначати межі відповідальності сервісів і проектувати їхні програмні інтерфейси;
- обґрунтовано обирати архітектурний підхід до сервісної взаємодії з урахуванням вимог системи;
- проектувати ресурсні моделі, адресацію, операції та формати представлення даних;
- застосовувати семантику HTTP-методів, коди стану, властивості безпечності та ідемпотентності операцій;
- формувати узгоджені моделі помилок і стратегії повторного виконання запитів;
- створювати формалізовані специфікації та документацію програмних інтерфейсів;
- реалізовувати механізми автентифікації та авторизації, застосовувати маркери доступу;
- забезпечувати версіювання, зворотну сумісність, тестування та контрольовану еволюцію програмних інтерфейсів.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Пререквізити дисципліни. Вивчення дисципліни спирається на знання основ програмування, об'єктно-орієнтованого аналізу та проектування, баз даних, комп'ютерних мереж, веб-технологій і клієнт-серверної архітектури, отримані під час опанування дисциплін попередніх семестрів:

- «Об'єктно-орієнтоване програмування» (3 семестр),
- «Веб-технології та веб-дизайн» (3 семестр),
- «Технології розробки програмного забезпечення» (5 семестр).

Постреквізити дисципліни. Матеріал дисципліни може бути використаний під час вивчення дисциплін, пов'язаних із проектуванням інформаційних систем, розподіленими та хмарними обчисленнями, програмною інженерією, а також під час виконання курсових і кваліфікаційних робіт.

3. Зміст навчальної дисципліни

Тема 1. Основи сервісної взаємодії та роль програмних інтерфейсів.

Тема 2. Протокол HTTP та ресурсна модель сервісів.

Тема 3. Проектування програмних інтерфейсів і забезпечення надійності взаємодії.

Тема 4. Формалізація контрактів та документування програмних інтерфейсів.

Тема 5. Безпека сервісної взаємодії.

Тема 6. Еволюція та супровід програмних інтерфейсів.

Тема 7. Забезпечення якості та тестування програмних інтерфейсів.

4. Навчальні матеріали та ресурси

Основна література

1. Jin B., Sahni S., Shevat A. *Designing Web APIs: Building APIs That Developers Love*. Sebastopol : O'Reilly Media, 2018. 230 p.
2. Newman S. *Building Microservices: Designing Fine-Grained Systems*. 2nd ed. Sebastopol : O'Reilly Media, 2021. 612 p.
3. Fielding R., Nottingham M., Reschke J. *HTTP Semantics*. RFC 9110. Internet Engineering Task Force, 2022. Режим доступу: <https://www.rfc-editor.org/rfc/rfc9110.html> (дата звернення: 01.06.26).
4. OpenAPI Initiative. *OpenAPI Specification* [Електронний ресурс]. Режим доступу: <https://spec.openapis.org/oas> (дата звернення: 01.06.2026).
5. OWASP Foundation. *OWASP API Security Top 10 – 2023* [Електронний ресурс]. Режим доступу: <https://owasp.org/API-Security/editions/2023/en/0x11-t10/> (дата звернення: 01.06.2026).

Додаткова література

1. Richardson L., Amundsen M., Ruby S. *RESTful Web APIs*. Sebastopol : O'Reilly Media, 2013. 406 p.
2. Masse M. *REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces*. Sebastopol : O'Reilly Media, 2011. 116 p.
3. Nottingham M., Wilde E., Dalal S. *Problem Details for HTTP APIs*. RFC 9457. Internet Engineering Task Force, 2023. Режим доступу: <https://www.rfc-editor.org/rfc/rfc9457.html> (дата звернення: 01.06.2026).
4. Jones M., Bradley J., Sakimura N. *JSON Web Token (JWT)*. RFC 7519. Internet Engineering Task Force, 2015. Режим доступу: <https://www.rfc-editor.org/rfc/rfc7519.html> (дата звернення: 01.06.2026).
5. Hardt D. *The OAuth 2.0 Authorization Framework*. RFC 6749. Internet Engineering Task Force, 2012. Режим доступу: <https://www.rfc-editor.org/rfc/rfc6749.html> (дата звернення: 01.06.2026).
6. Fette I., Melnikov A. *The WebSocket Protocol*. RFC 6455. Internet Engineering Task Force, 2011. Режим доступу: <https://www.rfc-editor.org/rfc/rfc6455.html> (дата звернення: 01.06.2026).
7. Nottingham M., Fielding R. *Additional HTTP Status Codes*. RFC 6585. Internet Engineering Task Force, 2012. Режим доступу: <https://www.rfc-editor.org/rfc/rfc6585.html> (дата звернення: 01.06.2026).
8. gRPC Authors. *gRPC Documentation* [Електронний ресурс]. Режим доступу: <https://grpc.io/docs/> (дата звернення: 01.06.2026).

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

5.1 Лекційні заняття

Тема 1. Односторінкові вебзастосунки, Angular та TypeScript

Лекція 1. Роль програмних інтерфейсів у сервіс-орієнтованих системах.

Поняття сервісу та меж його відповідальності. Програмний інтерфейс як формалізований контракт сервісу. Клієнти й постачальники сервісів. Еволюція підходів до програмної інтеграції. Підхід попереднього проектування контракту. Внутрішні, партнерські та загальнодоступні інтерфейси.

Лекція 2. Архітектурні підходи до організації сервісної взаємодії.

Поняття архітектурного стилю взаємодії. Ресурсно-орієнтований підхід, виклик віддалених процедур, декларативні запити та високопродуктивна міжсервісна взаємодія. Порівняння REST, GraphQL і gRPC. Критерії вибору підходу.

Лекція 3. Протокол HTTP як основа сервісної взаємодії.

Клієнт-серверна модель. Структура запиту та відповіді. Методи HTTP та їхня семантика. Безпечні й ідемпотентні операції. Коди стану. Заголовки та представлення даних.

Тема 2. Компонентна модель та механізми взаємодії компонентів

Лекція 4. Ресурсна модель сервісу.

Ресурс як зовнішнє представлення стану сервісу. Ідентифікація ресурсів. Колекції, вкладені ресурси та зв'язки. Правила побудови адрес. Типові помилки та гіпермедійні обмеження.

Лекція 5. Принципи проектування ресурсно-орієнтованих програмних інтерфейсів.

Архітектурні обмеження REST. Взаємодія без збереження клієнтського стану. Ресурси та дії. Фільтрація, сортування і посторінкове подання результатів. Типові помилки.

Лекція 6. Обробка помилок та забезпечення надійності сервісної взаємодії.

Помилки як частина контракту. Класифікація помилок. Часткові відмови. Обмеження часу очікування. Стратегії повторного виконання запитів. Ідемпотентність та ключі ідемпотентності.

Лекція 7. Формалізація контрактів програмних інтерфейсів.

Контрактно-орієнтований підхід. Призначення та структура специфікації OpenAPI. Опис операцій, параметрів, структур даних, відповідей і помилок. Формати YAML та JSON.

Тема 3. Сервіси, впровадження залежностей та архітектурна організація

Лекція 8. Документування програмних інтерфейсів.

Документація як інженерний артефакт. Структура документації. Автоматизоване подання специфікацій. Приклади запитів і відповідей. Супровід документації.

Лекція 9. Забезпечення безпеки програмних інтерфейсів.

Програмний інтерфейс як контрольована межа доступу до сервісу. Автентифікація та авторизація. Принцип мінімальних привілеїв. Основні класи загроз. Перевірка вхідних даних і журналювання.

Тема 4. Маршрутизація та організація функціональних частин застосунку

Лекція 10. Маркерні механізми автентифікації та делегування доступу.

Модель доступу на основі маркерів. Структура JWT. Підпис і перевірка маркерів. Призначення OAuth 2.0. Основні сценарії надання доступу. Маркери доступу й оновлення.

Лекція 11. Версіювання та сумісність програмних інтерфейсів.

Причини зміни контрактів. Сумісні та несумісні зміни. Зворотна сумісність. Стратегії версіювання. Паралельна підтримка версій.

Тема 5. Взаємодія з серверними API та реактивне програмування

Лекція 12. Життєвий цикл програмного інтерфейсу.

Проектування, реалізація, публікація, експлуатація та супровід. Керування змінами. Оголошення застарівання та виведення версій з експлуатації. Комунікація зі споживачами.

Лекція 13. Методи тестування програмних інтерфейсів.

Рівні тестування. Модульні, інтеграційні та контрактні перевірки. Позитивні й негативні сценарії. Перевірка схем даних, кодів стану та правил контракту.

Лекція 14. Автоматизація тестування та контроль якості програмних інтерфейсів.

Організація наборів автоматизованих перевірок. Параметризація сценаріїв. Регресійне тестування. Інтеграція перевірок у процес розробки. Критерії готовності до випуску.

Тема 6. Форми та валідація даних

Лекція 15. Продуктивність та стійкість сервісної взаємодії.

Показники продуктивності. Кешування. Обмеження інтенсивності запитів. Повторні виклики, ідемпотентність, обмеження часу очікування та запобігання перевантаженню.

Лекція 16. Взаємодія сервісів у розподілених і мікросервісних системах.

Автономність сервісів і контрактні межі. Синхронна та асинхронна взаємодія. Шлюзи програмних інтерфейсів. Подієва взаємодія. Постійні з'єднання, WebSocket і двосторонній обмін повідомленнями.

Тема 7. Тестування та оптимізація Angular-застосунків

Лекція 17. Програмний інтерфейс як відображення предметної області.

Предметно-орієнтований підхід. Межі контекстів і сервісів. Програмний інтерфейс як зовнішня проєкція доменної моделі. Захисний шар взаємодії між моделями.

Лекція 18. Сучасні напрями розвитку сервісної взаємодії.

Узагальнення моделей сервісної взаємодії. Еволюція від ресурсно-орієнтованих до декларативних, потокових і подієвих моделей. Передумови вибору GraphQL, gRPC, потокової та подієвої взаємодії.

5.2 Лабораторні роботи

Лабораторна робота 1. Проєктування сервісу та його ресурсної моделі.

Визначення меж відповідальності сервісу, формування ресурсів і операцій, побудова адрес ресурсів, вибір методів HTTP та кодів стану, обґрунтування безпечності й ідемпотентності операцій.

Лабораторна робота 2. Реалізація ресурсно-орієнтованого програмного інтерфейсу.

Реалізація операцій створення, читання, оновлення та видалення ресурсів, фільтрації, сортування і посторінкового подання результатів; формування узгоджених відповідей та повідомлень про помилки.

Лабораторна робота 3. Формалізація контракту та документування програмного інтерфейсу.

Створення специфікації OpenAPI, опис параметрів, моделей даних, відповідей і помилок; перевірка відповідності реалізації контракту та формування документації.

Лабораторна робота 4. Реалізація механізмів безпеки програмного інтерфейсу.

Реалізація автентифікації та авторизації, застосування маркерів доступу, розмежування прав доступу до операцій сервісу та перевірка помилкових сценаріїв.

Лабораторна робота 5. Версіювання, сумісність та надійність сервісної взаємодії.

Реалізація версіювання програмного інтерфейсу, внесення сумісних і несумісних змін, опрацювання повторних запитів, обмежень часу очікування та ключів ідемпотентності.

Лабораторна робота 6. Тестування та контроль якості програмного інтерфейсу.

Розроблення набору інтеграційних і контрактних перевірок, позитивних та негативних сценаріїв; перевірка кодів стану, структур даних, правил доступу та сумісності контракту; автоматизація перевірок і оцінювання якості програмного інтерфейсу.

Лабораторна робота 7. Інтеграція сервісів, продуктивність та стійкість сервісної взаємодії.

Організація взаємодії двох сервісів через програмні інтерфейси; застосування кешування, обмеження інтенсивності запитів, обмежень часу очікування та механізмів обробки відмов; оцінювання продуктивності й стійкості сервісної взаємодії.

Календарна синхронізація лекцій і лабораторних робіт

Тиждень	Лекція	Лабораторна робота
2	Лекція 2	ЛР 1 — початок
4	Лекція 4	ЛР 1 — завершення та захист
6	Лекція 6	ЛР 2
8	Лекція 8	ЛР 3

10	Лекція 10	ЛР 4
12	Лекція 12	ЛР 5
14	Лекція 14	ЛР 6
17	Лекція 17	ЛР 7 — виконання та захист
18	Лекція 18; МКР	Завершення заборгованостей

6. Самостійна робота студента

Тема 1. Порівняння моделей сервісної взаємодії та критеріїв вибору архітектурного підходу.

Тема 2. Дослідження семантики HTTP, безпечності та ідемпотентності операцій.

Тема 3. Опрацювання підходів до моделювання ресурсів, помилок і повторного виконання запитів.

Тема 4. Розроблення та аналіз формалізованої специфікації програмного інтерфейсу.

Тема 5. Аналіз типових загроз програмним інтерфейсам і засобів контролю доступу.

Тема 6. Дослідження стратегій версіювання, сумісності та виведення версій з експлуатації.

Тема 7. Розроблення набору автоматизованих перевірок програмного інтерфейсу.

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Протягом семестру студенти:

- виконують та захищають лабораторні роботи у відповідні терміни (на кожну роботу відводиться два тижні для здачі),
- пишуть модульну контрольну роботу.

Виконання та захист лабораторних робіт є обов'язковою складовою опанування дисципліни. На виконання кожної лабораторної роботи відводиться два тижні. У разі пропуску заняття з поважної причини студенту надається можливість виконати та захистити відповідну роботу в узгоджений із викладачем термін.

Лабораторні роботи виконуються в межах наскрізного семестрового проєкту. Кожна наступна лабораторна робота розвиває результати попередніх робіт. Студент повинен своєчасно виконувати завдання, демонструвати отримані результати та захищати прийняті проєктні й програмні рішення.

Несвоєчасне виконання лабораторних робіт без поважної причини може враховуватися під час оцінювання відповідно до рейтингової системи. Під час виконання робіт студент повинен дотримуватися принципів академічної доброчесності. Використання зовнішніх програмних засобів, бібліотек та генеративних систем допускається відповідно до завдання і вимог викладача із розумінням та здатністю пояснити створене рішення.

Політика та принципи академічної доброчесності визначені у розділі 3 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.

Норми етичної поведінки Норми етичної поведінки студентів і працівників визначені у розділі 2 Кодексу честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». Детальніше: <https://kpi.ua/code>.

Передбачена можливість зарахування результатів навчання за окремими темами дисципліни за результатами успішного проходження онлайн-курсів. Визнання таких результатів здійснюється відповідно до «Положення про визнання в КПІ ім. Ігоря Сікорського результатів навчання, набутих у неформальній/інформальній освіті» (<https://osvita.kpi.ua/node/179>).

Оскільки лабораторні роботи виконуються в межах наскрізного семестрового проєкту, зарахування результатів онлайн-навчання не звільняє студента від виконання тих складових проєкту, які необхідні для забезпечення його цілісності та виконання наступних лабораторних робіт.

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Система рейтингових (вагових) балів та критерії оцінювання

1) Лабораторні роботи

Максимальна кількість балів за лабораторну роботу визначається її обсягом і складністю та становить 10 або 15 балів відповідно до таблиці вагових коефіцієнтів.

Для оцінки роботи використовується наступні критерії оцінювання:

- виконаний своєчасно (протягом двох тижнів з моменту видачі), у повному обсязі – відповідний бал згідно номеру лабораторної роботи;
- виконаний із запізненням – знімається 10 – 30% від максимальної кількості балів в залежності від терміну запізнення;
- виконаний не самостійно, із запізненням – знімається 50% від максимальної кількості балів;
- невиконаний протягом відведеного часу – 0 балів.

Максимальна кількість балів за всі виконані лабораторні роботи становить 90 балів.

Розподіл рейтингових балів за лабораторні роботи наведено в таблиці.

№ з/п	Назва лабораторної роботи	Ваговий коефіцієнт
1	Проектування сервісу та його ресурсної моделі	10
2	Реалізація ресурсно-орієнтованого програмного інтерфейсу	15
3	Формалізація контракту та документування програмного інтерфейсу	15
4	Реалізація механізмів безпеки програмного інтерфейсу	15
5	Версіювання, сумісність та надійність сервісної взаємодії	15
6	Тестування та контроль якості програмного інтерфейсу	10
7	Інтеграція сервісів, продуктивність та стійкість сервісної взаємодії	10

Передбачено систему штрафних та заохочувальних балів:

активність на лабораторних роботах	+ 1 бали
виконання лабораторної роботи: • з використанням власного архітектурного рішення з дотриманням сучасних стандартів програмування; • застосовано обґрунтоване архітектурне або компонентне рішення, що виходить за межі базових вимог лабораторної роботи; • реалізовано додаткову функціональність Angular-застосунку, що виходить за межі базового завдання та відповідає сучасним практикам розробки; • покриття модульними тестами виконано на високому рівні.	+ 0,5 бали
несвоєчасна здача лабораторної роботи без поважної причини	-0,5 балів- -1 бал
обидві наскрізні атестації позитивні	+2 бал

Загальна сума рейтингових балів з урахуванням заохочувальних балів не може перевищувати 100 балів.

2) Календарний контроль

Проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

Для отримання «зараховано» з першої проміжної атестації студент повинен захистити не менше 2 лабораторних робіт.

Для отримання «зараховано» з другої проміжної атестації студент повинен захистити не менше 4 лабораторних робіт та набрати не менше 30 балів.

3) Модульна контрольна робота

Модульна контрольна робота (МКР) проводиться на лекційному занятті 18-го тижня з метою контролю рівня засвоєння матеріалу дисципліни.

Максимальна кількість балів за МКР дорівнює 10 балів.

Якість виконання роботи оцінюється за наступними критеріями:

- усі відповіді вірні та повні – 10 балів,
- у відповідях допущені несуттєві неточності – 8 балів,
- половина відповідей вірна – 5 балів,
- відповіді з суттєвими неточностями, але без критичних помилок – 2 бали,
- менше половини відповідей вірна – 0 балів.

Можливе також проведення модульної контрольної роботи у формі тестування, в такому випадку студент отримує бал, який відповідає частці правильних відповідей тесту з розрахунку, що 10 балів становлять 100%.

4) Семестровий контроль (залік)

Залік виставляється за результатами роботи студента протягом семестру відповідно до набраного рейтингу. Умовою отримання заліку є виконання та захист усіх лабораторних робіт, виконання модульної контрольної роботи та отримання не менше 60 рейтингових балів. За умови виконання вимог допуску додаткове залікове випробування не проводиться.

5) Розрахунок шкали (R) рейтингу:

Сума вагових балів контрольних заходів протягом семестру (шкала рейтингу) складає:

$$R = r_{\text{практ}} + r_{\text{мод}} = 90 + 10 = 100 \text{ балів.}$$

Таким чином, рейтингова шкала з кредитного модуля складає

$$R = 100 \text{ балів.}$$

Для отримання студентом відповідних оцінок рейтингова оцінка студента переводиться згідно таблиці:

Бали	Оцінка
95 - 100	Відмінно
85 - 94	Дуже добре
75 - 84	Добре
65 - 74	Задовільно
60 - 64	Достатньо
Менше 60	Незадовільно
R < 40 є незараховані лабораторні роботи або не виконані інші умови допуску до семестрового контролю	Не допущено

Робочу програму навчальної дисципліни (силабус):

Складено к.т.н., доцент каф. ЦТЕ, Тихоход Володимир Олександрович

Ухвалено кафедрою ЦТЕ (протокол № 22 від 17.06.26)

Погоджено науково-методичною комісією НН ІАТЕ КПІ ім. Ігоря Сікорського (протокол № 9 від 26.06.2026 р.)