



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	Заочна
Рік підготовки, семестр	3 курс, весняний семестр
Обсяг дисципліни	5 кред./150 год.: 6 год. лекцій, 4 год. лабораторних занять (у тому числі 2 год. на виконання МКР), 140 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

Лекція 1. Основи сервісної взаємодії, HTTP та ресурсне моделювання — 2 год.

Роль програмних інтерфейсів у сервіс-орієнтованих системах. Програмний інтерфейс як контракт взаємодії. Архітектурні підходи до сервісної взаємодії. Протокол HTTP: структура запиту і відповіді, методи, коди стану, безпечність та ідемпотентність операцій. Ресурсна модель сервісу, правила адресації ресурсів, фільтрація, сортування та посторінкове подання результатів.

Лекція 2. Проектування контрактів, надійність і безпека програмних інтерфейсів — 2 год.

Обробка помилок як складова контракту. Повторне виконання запитів, обмеження часу очікування, ідемпотентність і ключі ідемпотентності. Формалізація програмних інтерфейсів засобами OpenAPI та принципи документування. Автентифікація й авторизація, маркерні механізми доступу. Версіювання, зворотна сумісність, життєвий цикл та керування змінами програмного інтерфейсу.

Лекція 3. Еволюція, якість і взаємодія сервісів у розподілених системах — 2 год.

Методи тестування програмних інтерфейсів, контрактні та інтеграційні перевірки, автоматизація тестування. Кешування та обмеження інтенсивності запитів. Синхронна й асинхронна взаємодія сервісів, постійні з'єднання, WebSocket і двосторонній обмін повідомленнями. Програмний інтерфейс як відображення предметної області. Огляд GraphQL, gRPC, потокової та подієво-орієнтованої взаємодії.

5.2 Лабораторна робота (2 год.)

Лабораторна робота 1. Проектування, реалізація та перевірка програмного інтерфейсу сервісу — 2 год.

Проектування ресурсної моделі навчального сервісу; визначення операцій і адрес ресурсів; реалізація основних операцій програмного інтерфейсу; формування коректних кодів стану та повідомлень про помилки; створення специфікації OpenAPI; реалізація базового механізму контролю доступу; перевірка ідемпотентності та повторних запитів; виконання інтеграційних і контрактних перевірок. Робота виконується як єдине комплексне завдання на основі матеріалу трьох лекцій та самостійного опрацювання рекомендованих джерел.

5.3 Контрольні роботи

Передбачено одну модульну контрольну роботу тривалістю 2 академічні години за рахунок часу, відведеного на лабораторні заняття. МКР спрямована на перевірку засвоєння ключових положень проектування, надійності, безпеки, еволюції та тестування програмних інтерфейсів.

6. 6. Самостійна робота студента (140 год.)

Самостійна робота студента (140 год.) є основною формою опанування значної частини матеріалу за заочною формою навчання та передбачає опрацювання теоретичних питань за рекомендованими джерелами і технічною документацією, виконання практичних завдань, підготовку до аудиторних занять, лабораторної роботи, модульної контрольної роботи та заліку. Орієнтовний розподіл часу: самостійне опрацювання тем дисципліни — 105 год.; підготовка та виконання лабораторної роботи — 20 год.; підготовка до МКР — 7 год.; підготовка до заліку — 8 год.

Тема 1. Основи сервісної взаємодії та роль програмних інтерфейсів.

Самостійне опрацювання принципів сервісної взаємодії, типів програмних інтерфейсів, підходів до організації взаємодії між клієнтами та сервісами. Порівняння основних підходів до побудови програмних інтерфейсів та визначення сфер їх застосування.

Тема 2. Протокол HTTP та ресурсна модель сервісів.

Поглиблене вивчення семантики методів HTTP, кодів стану, заголовків, узгодження форматів представлення даних. Опрацювання принципів моделювання ресурсів, побудови адрес ресурсів, організації колекцій і зв'язків між ресурсами.

Тема 3. Проектування програмних інтерфейсів і забезпечення надійності взаємодії.

Опрацювання правил проектування операцій програмного інтерфейсу, безпечності та ідемпотентності операцій, моделей представлення помилок. Вивчення обмежень часу очікування, стратегій повторного виконання запитів, ключів ідемпотентності та підходів до забезпечення стійкості сервісної взаємодії.

Тема 4. Формалізація контрактів та документування програмних інтерфейсів.

Самостійне вивчення структури специфікації OpenAPI, способів опису операцій, параметрів, моделей даних, відповідей і помилок. Опрацювання принципів створення, автоматизованого формування, актуалізації та супроводу документації програмних інтерфейсів.

Тема 5. Безпека сервісної взаємодії.

Опрацювання основних загроз безпеці програмних інтерфейсів, принципів автентифікації та авторизації, механізмів використання маркерів доступу. Самостійне вивчення JWT, OAuth 2.0, принципу мінімальних привілеїв, перевірки вхідних даних та обмеження доступу до операцій сервісу.

Тема 6. Еволюція та супровід програмних інтерфейсів.

Вивчення причин і видів змін контрактів програмних інтерфейсів, сумісних і несумісних змін, стратегій версіювання та забезпечення зворотної сумісності. Опрацювання життєвого циклу програмного інтерфейсу, процедур оголошення застарівання та виведення версій з експлуатації.

Тема 7. Забезпечення якості та тестування програмних інтерфейсів.

Поглиблене опрацювання методів модульного, інтеграційного та контрактного тестування програмних інтерфейсів, позитивних і негативних сценаріїв та автоматизації перевірок. Самостійне вивчення засобів оцінювання продуктивності, кешування, обмеження інтенсивності запитів, а також особливостей синхронної, асинхронної та двосторонньої взаємодії сервісів.

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Протягом семестру студенти:

- виконують та захищають одну комплексну лабораторну роботу;
- пишуть модульну контрольну роботу.

Виконання та захист комплексної лабораторної роботи є обов'язковою складовою опанування дисципліни.

Несвоєчасне виконання лабораторної роботи 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>.

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

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

Максимальна кількість балів за виконання та захист комплексної лабораторної роботи становить 76 балів.

Для оцінки роботи використовується наступні критерії оцінювання:

- роботу виконано в повному обсязі, рішення обґрунтовано, студент здатний пояснити реалізацію — 69–76 балів;
- за неповне виконання, помилки в реалізації або недостатнє обґрунтування рішень кількість балів зменшується відповідно до характеру недоліків;
- у разі порушення академічної доброчесності робота оцінюється відповідно до чинних правил університету;
- невиконана або незахищена робота — 0 балів.

Модульна контрольна робота проводиться під час навчальної сесії. Для її проведення виділяються 2 академічні години за рахунок часу, передбаченого для лабораторних занять. Максимальна кількість балів за МКР — 24.

Якість виконання роботи оцінюється за наступними критеріями:

- повні, правильні та обґрунтовані відповіді — 22–24 бали,

- – правильний підхід за наявності окремих несуттєвих неточностей — 18–21 бал,
- – основні положення засвоєно, але відповіді неповні або містять суттєві неточності — 12–17 балів,
- – фрагментарне засвоєння матеріалу — 1–11 балів,
- – роботу не виконано або відповіді відсутні — 0 балів.

МКР може проводитися у формі письмової роботи або тестування; отриманий результат нормується до 24 балів.

Залік виставляється за результатами роботи студента протягом семестру відповідно до набраного рейтингу. Умовою отримання заліку є виконання та захист комплексної лабораторної роботи, МКР та отримання не менше 60 рейтингових балів.

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

$$R = r_{\text{практ}} + r_{\text{мод}} = 76 + 24 = 100 \text{ балів.}$$

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.)