

Про навички, необхідні Back-end розробнику

Дорожня карта з опанування потрібних бекенду технічних знань і hard skills має такий вигляд:

- Насамперед потрібно дізнатись, як працює інтернет — вивчити алгоритми роботи HTTP, браузерів, DNS, розібратися, що таке домен і хостинг;
- Обрати одну з операційних систем, спробувати працювати в терміналі, почитати про керування пам'яттю, концепції побудови мережі, керування I/O та POSIX;
- Визначитися з мовою програмування та фреймворком (у мене це C# та ASP.NET Core, зараз також набирають популярність Node.js та Go);
- Опанувати системи контролю версій та їхні хостинги (можете сміливо обирати GIT та GitHub);
- Вивчити реляційні та NoSql бази даних, SQL та ORM;
- Опанувати API — REST, GraphQL, json, gRPC, OpenAPI, способи автентифікації;
- Вивчити кешування — Redis, CDN;
- Опанувати написання тестів;
- Розібратися з CI/CD та Docker;
- Вивчити вебсервери, наприклад Nginx;

- Також не слід забувати про безпеку даних, тому почитайте про CORS, CSP, хеш-алгоритми, HTTPS, SSL/TLS.

план, який може допомогти вам стати бекенд розробником на мові програмування Golang:

1. Поглиблене вивчення мови Go:

- Ретельно вивчайте основи мови Go, включаючи синтаксис, типи даних, змінні, функції та пакети.
- Ознайомтесь з концепціями, такими як горутини (goroutines) та канали (channels), які роблять мову Go потужною для розробки бекенд додатків.

2. Робота зі знаменитими бібліотеками та фреймворками:

- Вивчайте популярні бібліотеки та фреймворки для розробки веб-додатків на мові Go, такі як Gin, Echo, Chi тощо.
- Досліджуйте, як створювати RESTful API та обробляти HTTP запити в Go.

3. Робота з базами даних:

- Ознайомтесь з підключенням до реляційних та нереляційних баз даних в мові Go. Вивчайте бібліотеки, такі як "database/sql" та "gorm" для роботи з SQL базами даних, або "MongoDB" для нереляційних.

4. Робота з автентифікацією та авторизацією:

- Досліджуйте, як реалізувати систему автентифікації та авторизації для вашого бекенд додатку.
- Вивчайте стандартні практики та бібліотеки для роботи з аутентифікацією, такі як JWT (JSON Web Tokens) або OAuth.

5. Тестування коду:

- Вивчайте методи тестування в мові Go, включаючи unit-тести, інтеграційні тести та тести відображення.
- Використовуйте популярні бібліотеки, такі як "testing" та "testify", для створення тестів вашого коду.

6. Оптимізація та швидкість:

- Розгляньте питання оптимізації та вдосконалення продуктивності вашого коду.
- Досліджуйте інструменти профілювання для знаходження та виправлення найповільніших частин вашого додатку.

7. Розгортання та ведення додатку:

- Вивчайте, як розгортати свій бекенд додаток на сервері. Розгляньте варіанти, такі як Docker та Kubernetes.
- Зрозумійте основи моніторингу та журналювання для забезпечення відслідковування стану вашого додатку.

8. Робота з командною розробкою:

- Розгляньте навички співпраці з іншими розробниками, використовуючи системи контролю версій, такі як Git.
- Вивчайте стандарти коду та практики колективного програмування.

9. Проектна робота:

- Розробіть власний проект, який використовує мову Go для створення бекенд частини. Це може бути власний API або інша веб-служба.

10. Навчання і підвищення кваліфікації:

- Продовжуйте навчатися і вдосконалювати свої навички, оскільки IT-сфера постійно розвивається. Вивчайте нові інструменти, технології та практики.

Мікросерверна архітектура - це підхід до розробки програмного забезпечення, в якому додаток розбивається на дрібні компоненти, відомі як мікросервіси, які незалежно функціонують і можуть співпрацювати між собою через API. Якщо ви бажаєте стати бекенд розробником, спеціалізованим на мікросерверній розробці на мові Go, то ось план, який може вам допомогти:

1. Розуміння мікросерверної архітектури:

- Ознайомтеся з основними концепціями мікросерверної архітектури, такими як незалежні мікросервіси, комунікація через API, масштабованість та забезпечення відмовостійкості.

2. Вивчення мови Go:

- Підвищте свої навички програмування на мові Go, особливо в галузі розробки мікросервісів.

3. Робота з мікросервісами на мові Go:

- Розгляньте, як створювати мікросервіси на мові Go.
- Досліджуйте способи комунікації між мікросервісами, такі як HTTP API або використання протоколів, таких як gRPC.

4. Контейнеризація та оркестрація:

- Вивчайте Docker для контейнеризації ваших мікросервісів.
- Розгляньте системи оркестрації, такі як Kubernetes, для управління та розгортання мікросервісів.

5. Моніторинг та журналювання:

- Досліджуйте інструменти моніторингу та журналювання для відстеження працездатності та продуктивності вашого мікросерверного додатку.

6. Безпека:

- Розгляньте питання безпеки мікросерверної архітектури, включаючи захист від атак, автентифікацію та авторизацію.

7. Тестування:

- Розробіть стратегію тестування для ваших мікросервісів, включаючи unit-тести, інтеграційні тести та тести відображення.

8. Розробка API:

- Вивчайте практики розробки API для взаємодії із зовнішніми клієнтами та іншими мікросервісами.

9. Проектна робота:

- Розробіть власний мікросерверний проект на мові Go, щоб отримати практичний досвід у цій галузі.

10. Постійне навчання:

- Постійно оновлюйте свої знання і слідкуйте за новими технологіями та практиками в мікросерверній розробці.