**I. Въведение в API и REST**

1. **Какво е API (Application Programming Interface)?**
   * Определение и роля в софтуерната разработка.
   * Видове API: Web API, OS API, Library API.
   * Примери за популярни API-та и тяхното приложение.
2. **REST (Representational State Transfer):**
   * Произход и основни принципи.
   * Архитектурни ограничения на REST: Client-Server, Stateless, Cacheable, Layered System, Uniform Interface.
   * Предимства на RESTful архитектурата.
3. **RESTful Web Services:**
   * Как REST се прилага в уеб услугите.
   * Сравнение с други архитектури (SOAP, GraphQL).
   * Защо REST е предпочитан за съвременни уеб приложения.

**II. Елементи на RESTful API**

1. **Ресурси:**
   * Какво представляват ресурсите в REST.
   * URL структура и именуване на ресурси.
   * Примери за ресурси и техните представяния (representations).
2. **HTTP Методи:**
   * GET, POST, PUT, DELETE, PATCH – обяснение и употреба.
   * Семантика на HTTP методите и тяхното съответствие с CRUD операциите.
   * Idempotency на методите.
3. **Формати на данни:**
   * JSON (JavaScript Object Notation):
     + Синтаксис и структура.
     + Предимства и недостатъци.
     + JSON Schema.
   * XML (eXtensible Markup Language):
     + Синтаксис и структура.
     + Предимства и недостатъци.
     + XML Schema (XSD).
   * Сравнение между JSON и XML.
4. **HTTP Status Codes:**
   * Класификация на кодовете за състояние.
   * Най-често използвани кодове (200 OK, 201 Created, 400 Bad Request, 404 Not Found, 500 Internal Server Error).
   * Семантично използване на кодовете за състояние.

**III. Client-Side работа с REST API**

1. **HTTP Клиенти:**
   * Вградени HTTP клиенти в езици за програмиране (например, XMLHttpRequest в JavaScript).
   * Библиотеки и frameworks за HTTP заявки:
     + JavaScript: fetch API, Axios.
     + Python: requests.
     + Java: HttpClient.
2. **Изпращане на HTTP заявки:**
   * Конфигуриране на HTTP заявки: URL, методи, headers, body.
   * Асинхронни заявки и Promises.
   * Обработка на отговори (responses): status code, headers, body.
3. **Обработка на JSON данни:**
   * Парсване на JSON отговори.
   * Сериализация на JSON данни за изпращане.
   * Валидиране на JSON данни.
4. **Работа с Headers:**
   * Content-Type, Accept, Authorization и други важни headers.
   * Как да задаваме и четем headers в HTTP заявки.

**IV. Сигурност на REST API**

1. **Удостоверяване (Authentication):**
   * Basic Authentication: Предимства и недостатъци.
   * API Keys: Как се генерират и използват.
   * OAuth 2.0:
     + Как работи OAuth 2.0 flow.
     + Roles и Grants.
     + Примери за OAuth 2.0 имплементации.
   * JWT (JSON Web Tokens):
     + Как работят JWT.
     + Структура на JWT.
     + Подписване и верификация на JWT.
2. **Оторизация (Authorization):**
   * Роли и права на достъп.
   * RBAC (Role-Based Access Control) и ABAC (Attribute-Based Access Control).
3. **HTTPS (Hypertext Transfer Protocol Secure):**
   * Важност на HTTPS за сигурност.
   * SSL/TLS сертификати.
4. **CORS (Cross-Origin Resource Sharing):**
   * Как работи CORS.
   * CORS политика и headers.
   * Преодоляване на CORS ограничения.
5. **API Rate Limiting:**
   * Защо е необходимо rate limiting.
   * Методи за имплементиране на rate limiting.
6. **Input Validation и Sanitization:**
   * Валидиране на данните, изпратени към API.
   * Предотвратяване на SQL Injection, XSS атаки и други заплахи.

**V. Документация и Тестване на REST API**

1. **API Документация:**
   * Важност на добрата документация.
   * Swagger/OpenAPI Specification:
     + Как се пише OpenAPI спецификация.
     + Генериране на документация от OpenAPI спецификация.
     + Swagger UI.
2. **Тестване на REST API:**
   * Unit Testing:
     + Как да пишем unit тестове за клиентски код, който използва REST API.
   * Integration Testing:
     + Как да тестваме интеграцията между клиентската част и API.
   * Инструменти за тестване (Postman, Insomnia, Jest, Mocha, Chai).

**VI. HATEOAS (Hypermedia as the Engine of Application State)**

1. **Какво е HATEOAS?**
2. **Предимства и недостатъци на HATEOAS.**
3. **Примери за HATEOAS имплементации.**

**VII. Заключение**

* Обобщение на основните концепции.
* Бъдещи тенденции в развитието на REST API.