**1. Что такое API?**

API (Application Programming Interface) — это интерфейс программирования приложений, который позволяет различным программным компонентам взаимодействовать друг с другом. API определяет набор правил и протоколов, которые позволяют одной программе запрашивать данные или функционал у другой программы или сервиса. Это помогает абстрагировать сложные детали реализации и предоставляет простой и стандартизированный способ взаимодействия.

**2. Как оно работает?**

API работает как посредник между различными программами. Когда одна программа хочет получить данные или выполнить действие, она отправляет запрос к API другой программы. API принимает запрос, обрабатывает его, взаимодействует с нужными ресурсами (например, базой данных) и возвращает результат в виде ответа.

Примерно так это выглядит:

* Приложение А отправляет запрос к API Приложения Б.
* API Приложения Б получает запрос и передаёт его внутреннему механизму обработки.
* Приложение Б выполняет необходимые действия и отправляет результат обратно через API.
* Приложение А получает ответ от API и использует его для своих целей.

**3. Что такое Fetch API / AJAX и как они работают?**

**AJAX (Asynchronous JavaScript and XML)** — это метод асинхронного взаимодействия с сервером, который позволяет загружать данные на веб-страницу без необходимости её перезагрузки. AJAX использует объект XMLHttpRequest для отправки запросов на сервер и получения данных в формате JSON, XML, HTML или простого текста.

**Fetch API** — это современная замена AJAX, предоставляющая более удобный и гибкий способ выполнения запросов. Fetch API возвращает промис (promise), который облегчает работу с асинхронными запросами.

**4. Что такое REST API, как они работают и примеры на PHP?**

**REST (Representational State Transfer)** — это архитектурный стиль взаимодействия в веб-приложениях, который основан на использовании стандартных HTTP-методов, таких как GET, POST, PUT, DELETE, для взаимодействия с ресурсами. REST API — это API, которое следует принципам REST. Основная идея REST заключается в том, что все операции (запросы) представляют собой манипуляции с ресурсами, которые идентифицируются по URI (Uniform Resource Identifier).

REST API работает следующим образом:

1. Клиент отправляет HTTP-запрос к REST API.
2. API обрабатывает запрос и выполняет действие над соответствующим ресурсом.
3. API возвращает ответ в формате, таком как JSON или XML, который клиент может использовать.

**Принципы REST**

1. **Статусное представление (Stateless):**
   * Каждый запрос от клиента к серверу должен содержать всю необходимую информацию для обработки запроса. Сервер не должен хранить состояние между запросами.
2. **Клиент-серверная архитектура:**
   * Клиенты и серверы должны быть разделены, что позволяет им развиваться независимо друг от друга. Клиент отвечает за интерфейс и взаимодействие с пользователем, а сервер — за логику приложения и управление данными.
3. **Код по требованию (необязательно):**
   * Сервер может предоставлять исполняемый код (например, JavaScript) для выполнения на клиенте. Это опционально и не всегда используется.
4. **Единый интерфейс (Uniform Interface):**
   * Использование стандартных HTTP-методов и URL для взаимодействия с ресурсами:
     + **GET** — получение данных.
     + **POST** — создание новых данных.
     + **PUT** — обновление существующих данных.
     + **DELETE** — удаление данных.
5. **Модульность и гибкость (Layered System):**
   * Архитектура может быть организована в виде слоев, где каждый слой выполняет определенную функцию, что позволяет скрывать сложность системы и улучшать масштабируемость.
6. **Кэширование (Caching):**
   * Ответы на запросы могут быть кэшированы для повышения производительности и уменьшения нагрузки на сервер.

**Где применяется REST API?**

**REST API** используется во многих областях и для различных целей:

1. **Веб-разработка:**
   * Взаимодействие между клиентскими приложениями (например, веб-сайтами и мобильными приложениями) и сервером для выполнения операций CRUD (создание, чтение, обновление, удаление) над данными.
2. **Мобильные приложения:**
   * Мобильные приложения часто используют REST API для общения с сервером, получения данных и отправки пользовательского ввода.
3. **Интернет вещей (IoT):**
   * REST API применяется для взаимодействия между устройствами IoT и серверами для обмена данными и управления устройствами.
4. **Интеграция сервисов:**
   * REST API позволяет интегрировать различные веб-сервисы и системы, предоставляя стандартизированный способ обмена данными.
5. **Облачные сервисы:**
   * Облачные платформы и сервисы (например, Amazon Web Services, Google Cloud) предоставляют REST API для управления ресурсами и выполнения различных операций.

**Как работает REST API?**

1. **Определение Ресурсов:**
   * Ресурсы (например, пользователи, задачи, товары) идентифицируются с помощью URL. Например, https://api.example.com/users может представлять коллекцию пользователей.
2. **HTTP Методы:**
   * **GET**: Получение данных (например, список пользователей).
   * **POST**: Создание нового ресурса (например, добавление нового пользователя).
   * **PUT**: Обновление существующего ресурса (например, изменение информации о пользователе).
   * **DELETE**: Удаление ресурса (например, удаление пользователя).
3. **Запрос и Ответ:**
   * Клиент отправляет HTTP-запрос на сервер. Запрос может содержать параметры, заголовки и тело (для POST и PUT запросов).
   * Сервер обрабатывает запрос, выполняет нужные операции и отправляет ответ обратно клиенту. Ответ обычно содержит данные в формате JSON или XML и код состояния HTTP.

**В примерах используются общедоступные API не требующие авторизации и т.д. Все проверяем через Postman**