

1. Дайте определение понятию «Интернет».

Информационно-коммуникационная сеть и всемирная система объединённых компьютерных сетей для хранения и передачи информации.

- 1) сеть на основе TCP/IP;
- 2) стандарты Internet (RFC, STD);
- 3) службы Интернет (DNS, SMTP/POP3/IMAP, WWW, FTP, Telnet, SSH,...);
- 4) организации, управляющие сетью Internet (ISOC, IETF, W3C, ICANN, IANA, ...).

2. Дайте определение понятию «Служба Интернет».

Это системы, предоставляющие услуги пользователям Интернета. К ним относятся: электронная почта, WWW, телеконференции, списки рассылки, FTP, IRC, а также другие продукты, использующие Интернет как среду передачи информации.

3. Дайте определение понятию «Узел сети Интернет».

Устройство, имеющее IP-адрес и подключенное сети Интернет (обычно к сети Интернет-провайдера). Каждый узел характеризуется своей программно-аппаратной платформой – аппаратурой и операционной системой.

4. Дайте определение понятию «клиент-серверное приложение».

Приложение (программа) с клиент-серверной архитектурой:

- состоит из двух компонент – клиента и сервера;
- клиент и сервер взаимодействуют между собой в соответствии с заданными правилами (протоколами);
- для взаимодействия между клиентом и сервером в соответствии с правилами (протоколом) должно быть установлено соединение;
- инициатором соединения – клиент.

5. Дайте определение понятию «сетевой протокол».

Это набор правил, определяющий принципы взаимодействия устройств в сети.

6. Перечислите основные свойства протокола HTTP.

- версии HTTP/1.1 – действующий (текстовый), HTTP/2 – действующий (бинарный), HTTP/3 (раньше HTTP-over-QUIC, стадия черновика);
- два типа абонентов: клиент и сервер;

- два типа сообщений: request и response;
- от клиента к серверу – request;
- от сервера к клиенту – response;
- на один request всегда один response, иначе ошибка;
- одному response всегда один request, иначе ошибка;
- stateless;
- TCP-порты: 80, 443;
- для адресации используется URI или URL;
- поддерживается W3C, описан в нескольких RFC.

7. Перечислите состав информации, пересылаемой в HTTP-запросе.

- метод;
- URI;
- версия протокола (HTTP/1.1);
- заголовки (пары: имя/значение);
- тело.

8. Перечислите состав информации, пересылаемой в HTTP-ответе.

- версия протокола (HTTP/1.1);
- код состояния (1xx, 2xx, 3xx, 4xx, 5xx);
- пояснение к коду состояния;
- заголовки (пары: имя/заголовков);
- тело.

9. Перечислите группы заголовков HTTP.

General: общие заголовки, используются в запросах и ответах.

Request: используются только в запросах.

Response: используются только в ответах.

Entity: для сущности в ответах и запросах.

10. Дайте определение понятию «web-приложение».

Клиент-серверное приложение, у которого клиент и сервер взаимодействуют по протоколу HTTP.

11. Дайте определение понятиям «frontend» и «backend».

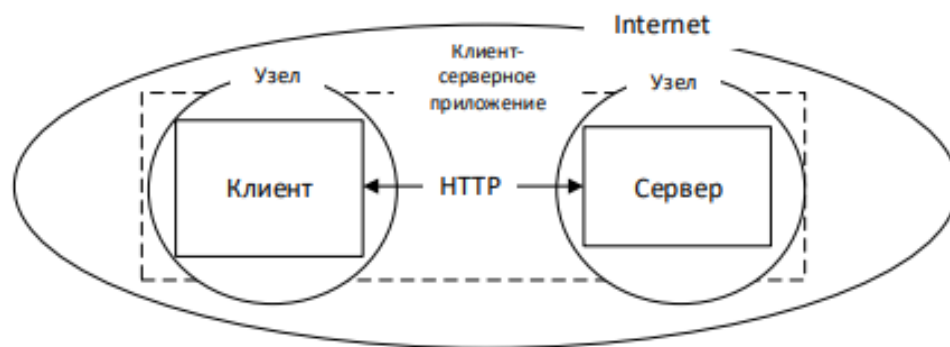
Backend – серверная часть. Frontend – клиентская часть.

12. Дайте определение понятию «кроссплатформенное приложение».

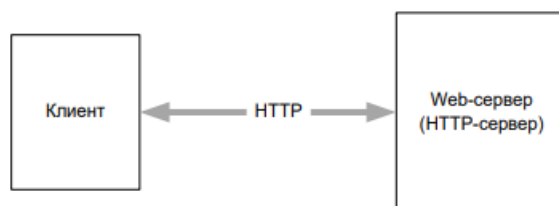
Приложение, способное работать на более чем одной программно-аппаратной (аппаратура + операционная система) платформе.

13. Изобразите и поясните общую схему web-приложения.

Курс посвящен разработке серверной части web-приложения или иначе разработке web-сервера (backend).

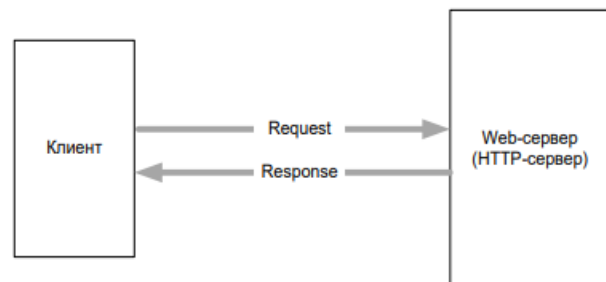


Архитектура web-приложения



Web-приложение состоит из клиента и сервера, взаимодействующих по протоколу HTTP.

Взаимодействие по протоколу HTTP происходит посредством запросов и ответов.



14. Назовите основные технологии разработки серверных кроссплатформенных приложений.

- PHP / Apache, LAMP;
- Java / JVM / Application Server;
- C# / ASP.NET CORE;
- Python / Django;
- Ruby on Rails;
- JS / Node.js.

15. Поясните понятие «асинхронная операция».

Запрос, при котором поток, выдавший http-запрос, не блокируется до поступления запроса; для обработки ответа применяется функция обратного вызова.

16. Поясните основное назначение NODE.JS.

Node.js — это среда выполнения, которая позволяет разработчикам запускать код JavaScript вне веб-браузера на стороне сервера. Основная цель Node.js — дать разработчикам возможность создавать быстрые, масштабируемые и эффективные сетевые приложения, которые могут обрабатывать большое количество одновременных подключений с высокой пропускной способностью.

17. Поясните основные свойства NODE.JS.

Управляемый событиями и неблокирующий ввод-вывод: Node.js построен на управляемой событиями и неблокирующей модели ввода-вывода, что позволяет ему обрабатывать большое количество одновременных подключений, не блокируя выполнение других задач. Это делает его хорошо подходящим для создания приложений, работающих в режиме реального времени, с интенсивным использованием данных, требующих высокой производительности.

Кроссплатформенность: Node.js предназначен для работы на нескольких платформах, включая Windows, Linux и macOS. Это упрощает разработку и развертывание приложений в различных операционных системах.

Однопоточный, но хорошо масштабируемый: Node.js является однопоточным, что означает, что он может выполнять только одну задачу за раз. Однако он использует цикл обработки событий и пул потоков для одновременной обработки нескольких запросов, что делает его высоко масштабируемым и эффективным.

Легкий и быстрый: Node.js построен на движке JavaScript V8, известном своей скоростью и эффективностью. Это делает Node.js легким и быстрым, позволяя

обрабатывать большое количество запросов с высокой пропускной способностью.

NPM: Node.js поставляется с мощным менеджером пакетов под названием NPM, который позволяет разработчикам легко устанавливать сторонние пакеты и модули и управлять ими. Это способствовало созданию динамичной экосистемы пакетов с открытым исходным кодом, которые можно использовать для создания приложений Node.js.

Полный стек JavaScript: Node.js позволяет разработчикам использовать JavaScript как на стороне клиента, так и на стороне сервера своих приложений, что упрощает совместное использование кода и поддерживает согласованность на протяжении всего процесса разработки.