

Шпаргалка: как устроены сети

Сетевая модель OSI (Open System Interconnection — взаимодействие открытых систем) — схема, которая задаёт правила обмена информацией. Это модель стандартизована ISO (International Organization for Standardization — Международная организация по стандартизации).

Модель **TCP/IP** (Transmission Control Protocol/Internet Protocol) создана на основе протоколов TCP и IP. Она соответствует стандарту OSI.

Вот как соотносятся уровнями модели OSI и TCP/IP:



Каждому уровню соответствуют свои наборы протоколов.

Протокол — это набор правил, которые позволяют уровням общаться между собой. У каждого своя задача.

Запрос на передачу данных от приложения поступает на **прикладной уровень (application layer)**. Он забирает у приложения данные и отправляет на следующий уровень обработки сетевой модели. Один из самых распространённых протоколов прикладного уровня — HTTPS.

Транспортный уровень (transport layer) контролирует и обеспечивает передачу данных по сети. На этом уровне работают протоколы UDP и TCP.

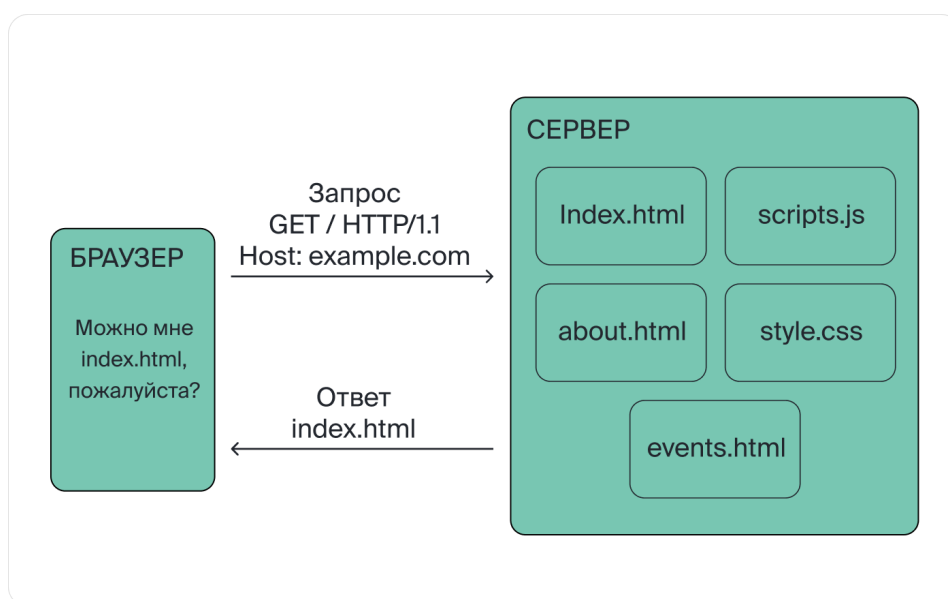
Сетевой уровень (network layer) — на нём данные формируются в пакеты; чтобы точно рассчитать маршрут доставки, добавляются адреса клиента и сервера. На этом уровне работает протокол IP (Internet Protocol), который формирует IP-адрес.

IP address — номер, который присвоен каждому устройству в сети, чтобы передавать данные и их местоположение через IP.

Уровень сетевых интерфейсов (network interface layer) — определяет, как упаковать данные, чтобы передать по сети. На нём также назначается оборудование, по которому данные передадутся в виде электрических или оптических сигналов. Основные протоколы — Ethernet (локальная сеть) и IEEE 802.11 (беспроводная сеть).

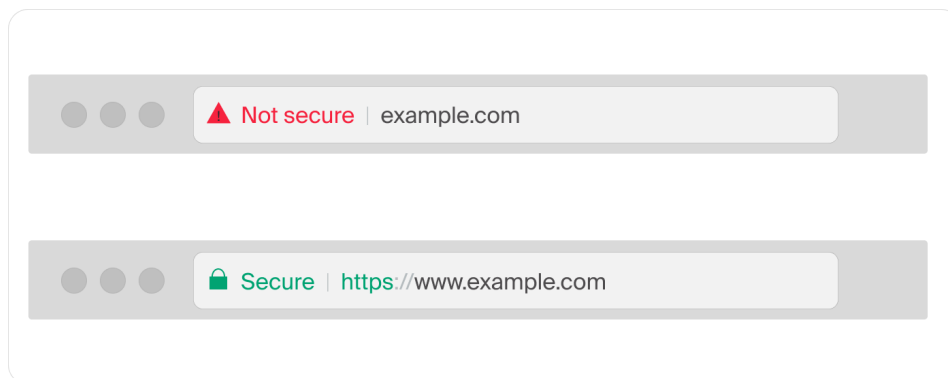
HTTP (HyperText Transfer Protocol — протокол передачи гипертекста) — это протокол прикладного уровня, который позволяет клиенту и серверу обмениваться сообщениями.

Клиент отправляет запрос (request); сервер возвращает ответ (response).



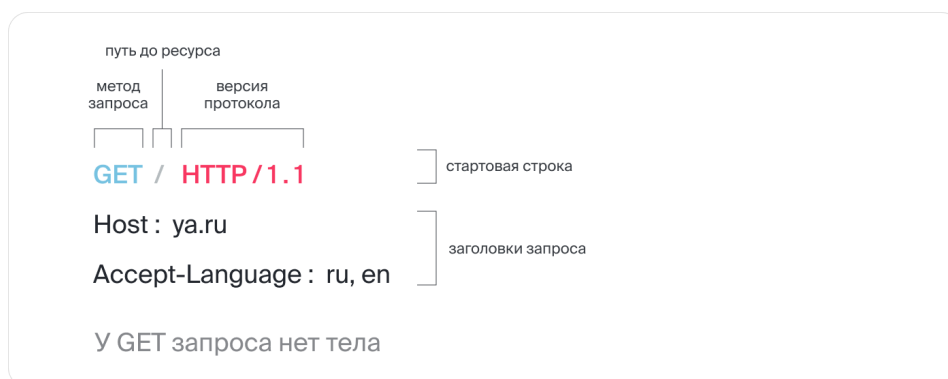
Чтобы обеспечить безопасность, применяют протокол HTTP с расширением защиты — **HTTPS** (HyperText Transfer Protocol Secure — протокол защищённой передачи гипертекста). Он шифрует соединение по криптографическим протоколам: так клиент и сервер смогут безопасно обмениваться данными.

Как браузер отображает подключение по HTTP и HTTPS:



HTTP-запрос состоит из трёх блоков:

- стартовая строка,
- заголовки,
- тело сообщения.



Стартовая строка (start line) состоит из трёх элементов: метода, пути до ресурса и версии протокола. Например, `GET / HTTP/1.1`.

Заголовки запроса (request headers) — это дополнительная информация от клиента серверу.

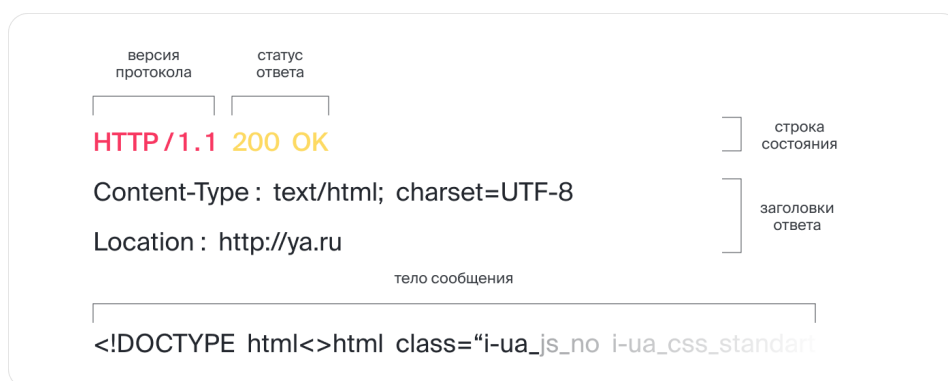
Метод (method) указывает действие: сервер принимает его в обработку. Самые распространённые — GET и POST.

МЕТОД	ДЕЙСТВИЕ
GET	Запросить у сервера данные по определённому адресу
POST	Отправить данные на сервер

Путь до ресурса — адрес запроса в формате URI.

Версия протокола задаёт формат взаимодействия клиента и сервера. Сейчас применяют версию HTTP/1.1.

HTTP-ответ также состоит из нескольких элементов: строки состояния, заголовка, тела сообщения.



Строка состояния состоит из версии протокола, кода состояния и текста состояния. Например, `HTTP/1.1 200 OK`.

Версия протокола — номер версии HTTP, для которой сформирован ответ.

Код состояния (status code) — трёхзначное число, которое указывает код результата: успешно ли сервер обработал запрос; например, 200. Его сопровождает текст состояния «OK».

Описание кодов состояний:

Коды	Описание	Пример
1xx	Информационные сообщения	102 – запрос принят, но обработка ещё не завершена.
2xx	Сообщения об успехе	200 - OK, запрос обработан успешно.
3xx	Перенаправление	302 – запрошенный ресурс временно доступен по другому адресу.
4xx	Клиентские ошибки	404 – страница, которую запросил клиент, не найдена на сервере.
5xx	Ошибки сервера	500 – внутренняя ошибка сервера.

Заголовки ответа (response headers) — дополнительная информация о сервере и типе данных для клиента.

URL (Uniform Resource Locator — унифицированный указатель ресурса) говорит, где находится веб-ресурс и как к нему обратиться.

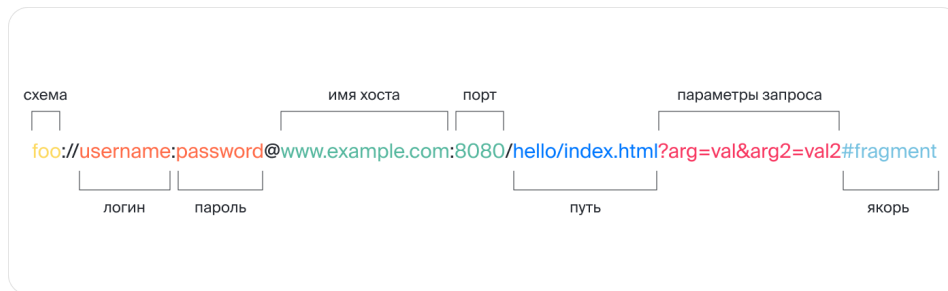


Схема (scheme) — протокол, по которому передаются данные. Например, HTTP, HTTPS.

Логин:пароль (username:password) указывают серверу, какой пользователь обратился к нему. К логину пользователя привязаны права доступа. Если данные передают по HTTPS, логин и пароль могут не указывать.

Символ @ (at коммерческое, «собака») отделяет логин:пароль (username:password) и имя хоста:порт (hostname:port). Если логин:пароль не применяют, «собаку» — тоже.

Имя хоста:порт (hostname:port) — доменное имя или IP-адрес сервера, к которому обращается пользователь.

- **Хост** (host) — доменное имя или IP-адрес, к которому обращается браузер. Хранит данные веб-приложения.
- **Порт** (port) — виртуальный номер, который сообщает серверу, в какую службу отправить запрос.

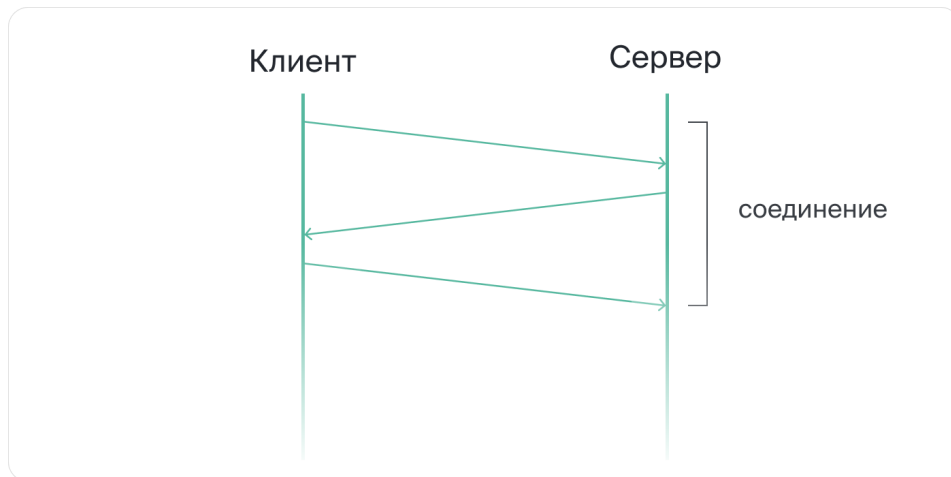
Путь (path) — месторасположение ресурса. Похоже на путь к файлу или папке в файловой системе.

Параметры запроса (query) — дополнительные параметры вида `ключ=значение`. Перечисляются после знака `?` и разделяются через `&`.

Фрагмент текста (fragment), или якорь (anchor) — дополнительный указатель; позволяет сразу попасть в нужную часть веб-страницы: например, к заголовку или абзацу в тексте.

TCP и UDP — протоколы транспортного уровня. Применяются во всех операциях в интернете.

Протокол **TCP** (Transmission Control Protocol) обеспечивает надёжную доставку данных. Перед отправкой данных между клиентом и сервером устанавливается соединение.



TCP:

- устанавливает соединение между клиентом и сервером до начала передачи данных,
- отправляет фрагменты данных в определённом порядке,
- сообщает клиенту о статусе доставки данных.

Протокол **UDP** (User Datagram Protocol), в отличие от TCP, не устанавливает соединение перед передачей данных, а сразу отправляет их. К тому же он не ждёт ответ от сервера о статусе доставки. Это не обеспечивает надёжность доставки данных, но ускоряет их передачу.

UDP:

- не гарантирует доставку данных,
- не сохраняет порядок фрагментов данных,
- обеспечивает ускорение передачи данных.