

## 6.1. Основы IP сетей

Сайт: [Samsung Innovation Campus](https://innovationcampus.ru)  
Курс: Мобильная разработка на Kotlin  
Книга: 6.1. Основы IP сетей

Напечатано:: Murad Rezvan  
Дата: понедельник, 3 июня 2024, 19:14

## Оглавление

- 1. 6.1.1. Об интернете и протоколах TCP/IP
- 2. 5.1.2. Адресация в IP-сетях
- 3. 6.1.3. Версия интернет-протокола IPv4
- 4. 6.1.4. Доменные имена (DNS), URL-ссылки
- 5. 6.1.5 HTTP-протокол

## 1. 6.1.1. Об интернете и протоколах TCP/IP

Прежде чем перейти к изучению тем по клиент-серверной разработке приложений, остановимся кратко на основных понятиях, связанных с интернетом и протоколами, которые обеспечивают его работу.

В июле 1961 года американский ученый Л. Клейнрок опубликовал первую статью по теории пакетной коммутации. Он предложил передавать данные через сети. Для этого он предложил разделить их на небольшие пакеты в несколько десятков байт и потом в адресате из них собрать исходное сообщение. Уже в 1969 году коллектив ученых под его руководством в Калифорнийском университете и ученые Стэнфордского исследовательского института впервые продемонстрировали передачу данных с использованием набора сетевых протоколов TCP (Transmission Control Protocol). Пакет данных прошел по маршруту Сан-Франциско — Лондон — Университет Южной Калифорнии, при этом не потеряв ни одного бита. Этот день, 29 октября 1969 года, многие историки считают днем рождения интернета.

**Интернет-протокол** — это набор правил, по которым компьютеры взаимодействуют между собой. Без протоколов не было бы интернета, потому что устройства в сети не понимали бы друг друга.

В 1978 году TCP был разделен на две отдельные группы:

- за разбивку передаваемого сообщения на пакеты данных и их сборку в пункте получения стал отвечать TCP;
- за передачу пакетов данных с контролем получения — IP-протокол (Internet Protocol).

По отношению к протоколам TCP/IP употребляют понятие «стек протоколов». Так происходит потому, что сейчас это уже большое множество протоколов, которые «слоями» покрывают друг друга: верхние работают на высоком логическом уровне, не вдаваясь в подробности, которые задают протоколы более низкого уровня. Сейчас в стеке протоколов TCP/IP по одной из классификаций выделяют четыре уровня.

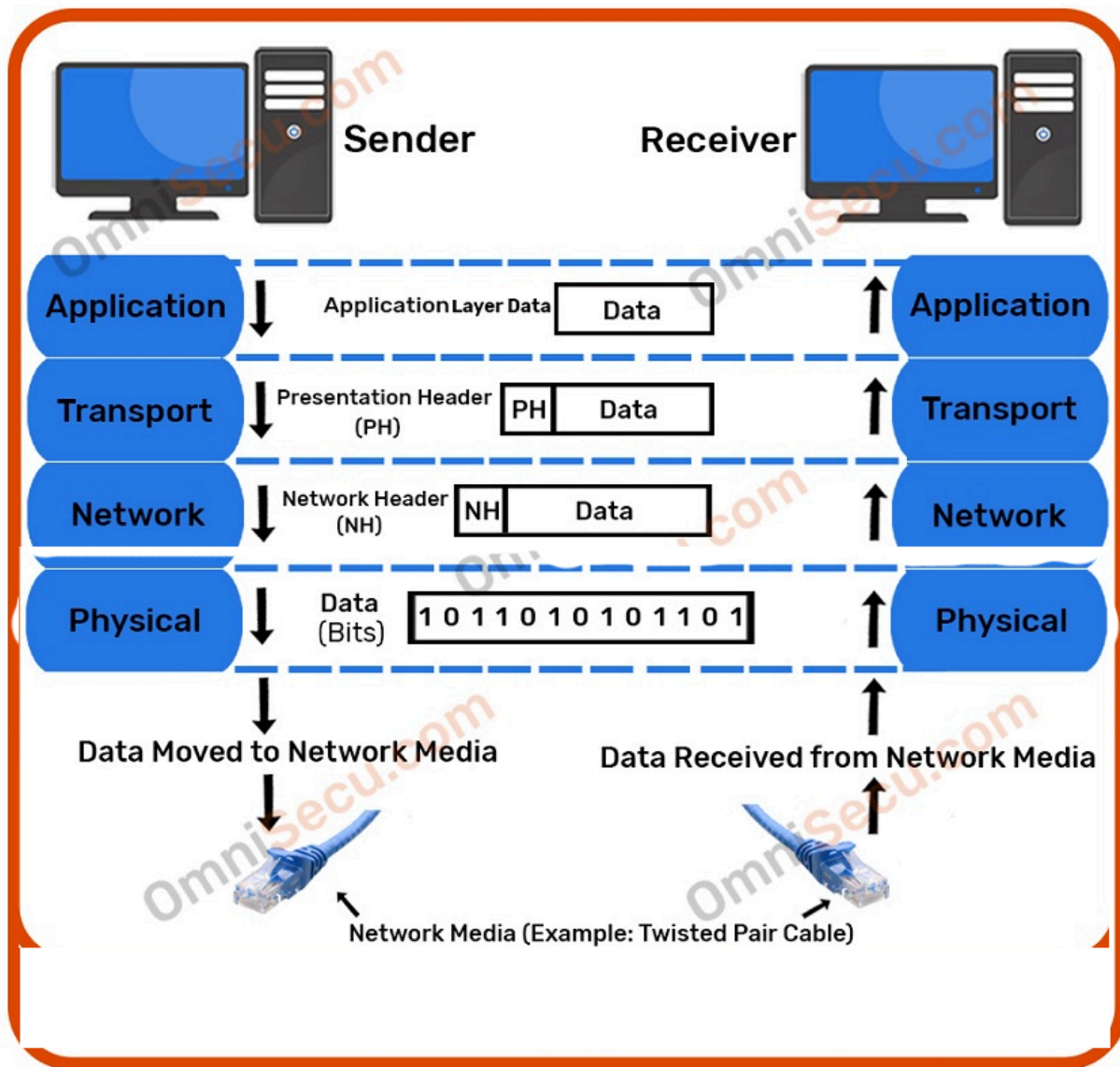
4) Application layer. Самый верхний прикладной уровень. Здесь работают протокол HTTP для WWW, FTP (передача файлов), SMTP (электронная почта), DNS (преобразование символьных имен в IP-адреса) и многие другие.

3) Transport layer. Протоколы транспортного уровня TCP, UDP решают задачу передачи пакетов данных и определяют, для какого конкретно приложения они предназначены.

2) Network layer. IP-протоколы сетевого уровня занимаются определением кратчайшего пути передачи пакетов данных, переводом логических адресов и имен в физические и др. На данном уровне работает сетевое устройство — маршрутизатор.

1) Physical layer. Самый приближенный к физическим устройствам канальный уровень протоколов. Он предназначен для решения задачи передачи данных узлам, находящимся в том же сегменте локальной сети. Одним из примеров такого протокола является Ethernet.

Теперь рассмотрим, как работает передача данных «слоями». Для передачи используется принцип инкапсуляции: сформированный пакет данных (см. Data на рис. 5.1) заворачивается (инкапсулируется) в заголовок протоколом прикладного уровня (например, FTP), затем все это (включая заголовок FTP) инкапсулируется вновь протоколом транспортного уровня (скажем, TCP), затем следующим (например, IP), и, наконец, финальным, низкоуровневым протоколом связи (Ethernet). В конце пакет начинает передаваться через физическую среду передачи. Каждый уровень скрывает данные вышестоящих уровней, но добавляет ту информацию (обычно заголовки), которая необходима для выполнения текущих задач.



Когда другой компьютер получает этот пакет, оборудование (сетевая карта) исключает Ethernet-заголовок (разворачивает пакет), ядро ОС исключает заголовки IP и TCP, программа получатель исключает заголовок FTP, и, наконец, мы получаем исходные данные.

## 2. 5.1.2. Адресация в IP-сетях

Сетевой адрес, также как и почтовый адрес – это информация, которая содержит данные, необходимые для доставки информации адресату. Для работы с сетевыми протоколами различных уровней стека TCP/IP используют три типа адресов:

- MAC-адрес (физический адрес);
- IP-адрес (сетевой адрес);
- DNS-имя (символьное доменное имя).

**MAC-адрес** (от Media Access Control) — это уникальный идентификатор, присваиваемый каждой единице активного оборудования, или некоторым их интерфейсам в компьютерных сетях Ethernet.

Свой MAC-адрес есть как у адаптера сети Wi-Fi, так и у адаптера проводной сети Ethernet. MAC-адрес — это шестибайтный номер. Обычно он записывается в шестнадцатиричной системе счисления, например, следующим образом: E1:39:F5:7A:E2:22. MAC-адреса устройств являются уникальными. Это обусловлено тем, что каждый производитель оборудования получает в пользование диапазон из шестнадцати миллионов адресов в комитете IEEE Registration Authority. Три старших байта идентифицируют производителя, три младших назначаются самим производителем. MAC-адреса формируют основу уровней каналов. Затем ее используют протоколы сетевого уровня.

Вы можете узнать MAC-адрес сети на ваших устройствах двумя способами:

Меню -> Настройки -> Сведения об устройстве (Об устройстве) -> Состояние -> MAC-адрес Wi-Fi. Меню -> Настройки -> Wi-Fi -> Меню -> Дополнительно -> MAC-адрес.

**IP -адрес** (от Internet Protocol Address) — представляет собой основной тип адресов, с помощью которых происходит обмен пакетами на сетевом уровне. Такие адреса состоят из 4 байт и записываются четырьмя десятичными числами от 0 до 255, разделенными точками, например, 195.223.68.11. IP-адрес назначается администратором во время конфигурации компьютеров и маршрутизаторов. Если в локальной сети много компьютеров, то настройка адресов становится существенной проблемой и в этой ситуации системному администратору приходит на помощь служба автоматической настройки адресов - DHCP.

**DNS -имя** (Domain Name System) — символьный имя-идентификатор, такой как, например, myitschool.ru. Доменное имя представляет собой буквенные адреса. Они гораздо удобнее для восприятия и использования, чем последовательность цифр IP-адреса.

### 3. 6.1.3. Версия интернет-протокола IPv4

Сегодня IP-адрес может быть в двух форматах: IPv4, который имеет длину 4 байта, например, 192.168.0.1, и протокола IPv6, состоящего из 16 байт, например, 1021:0:6ed2:7a1b:2182:20:4db5:a1d4.

Протокол IPv4 появился еще в 1981 году. В настоящий момент это самая часто используемая версия. Однако в связи с постоянным ростом числа устройств в сети интернет, количества адресов, состоящих из 4 байт, уже не хватает. Для этого была запущена версия протокола IPv6. Она способна поддерживать значительно большее количество уникальных адресов. В настоящее время наблюдается рост использования протокола IPv6. Если на конец 2013 года доля IPv6 в мировом сетевом трафике составляла около 3%, то на сегодняшний день она превышает 15% и продолжает расти.

Далее в учебнике будет рассматриваться только версия протокола IPv4. Проанализируем, например, адрес 192.168.104.115. Несмотря на то что адрес визуально один, он состоит из двух частей: адрес (номер) сети и адрес (номер) компьютера внутри сети. Для того чтобы выделить эти части из IP-адреса, используют маски подсети.

**Маска подсети** — битовая маска, определяющая, какая часть IP-адреса узла сети относится к адресу сети, а какая — к адресу самого узла в этой сети. При этом, в отличие от IP-адреса, маска подсети не является частью IP-пакета и не передается с прочей информацией адресату.

Так же, как и IP-адрес, маска состоит из четырех чисел в диапазоне от 0 до 255 включительно. Однако она устроена особым образом, по принципу: «п единиц, потом — нули» в двоичном коде. Для получения адреса подсети необходимо выполнить поразрядную конъюнкцию маски сети и IP-адреса устройства. В разрядах маски, где стоят 1, значение соответствующих разрядов в IP-адресе не изменится, а там, где 0, — обнулится. Проиллюстрируем это на примерах.

#### Пример 6.1.1

Пусть имеем IP-адрес 192.168.105.123 и маску подсети 255.255.254.0, которая в двоичном виде имеет вид:

```
11000000.10101000.01101001.01111011 - адрес
& 11111111.11111111.11111110.00000000 - маска
-----
11000000.10101000.01101000.00000000 - сеть
00000000.00000000.00000001.00000000 - хост
```

Это значит, что первые 23 бита адреса — это адрес сети (192.168.104.0), а оставшиеся 9 битов — номер узла (компьютера) в этой сети (1.123 или 379).

Можно использовать другую запись, которая значит то же самое:

192.168.105.123/23 — «/23» говорит о том, что в маске 23 единицы.

В такой сети может быть 510 узлов, а не 512, как можно было бы ожидать. Дело в том, что младший адрес (192.168.104.0) используется для обозначения всей сети, а старший (192.168.104.255) — для так называемой широковещательной рассылки (сообщение отправляется всем компьютерам данной сети). Все узлы с адресами от 192.168.104.1 до 192.168.104.254 находятся в той же сети, что и данный компьютер.

#### Пример 6.1.2

Рассмотрим IP-адрес из примера 6.1, но с другой маской 255.255.255.248. В двоичной системе она содержит 29 единиц и 3 нуля.

Адрес:	192	.	168	.	105	.	123
	11000000.	10101000.	01101001.	01111		011	
Маска:	11111111.	11111111.	11111111.	11111		000	
	255	.	255	.	255	.	248

Таким образом узел с адресом 192.168.105.123/29 — это узел номер 3 (011<sub>2</sub>) в сети 192.168.105.120 (11000000.10101000.01101000.01111000<sub>2</sub>).

Так как на адрес узла выделено три бита (в маске три нуля), в такой сети доступно только  $2^3 = 8$  адресов. Кроме того, учитывая, что два из них специальные (первый — номер сети, последний — широковещательный адрес), то в такую сеть может входить не более 6 узлов.

Необходимо помнить, что IP-адрес назначается не компьютеру, а каналу передачи данных (проводной сетевой карте, Wi-Fi-адаптеру, модему). Таким образом, один компьютер может иметь множество IP-адресов, например, если на нем установлены две сетевые карты.

Ниже приведены диапазоны IP-адресов (так называемые «серые» адреса), которые не используются в сети интернет. Они рекомендованы для локальных сетей:

- 10.0.0.0–10.255.255.255(10.0.0.0/8)
- 172.16.0.0–172.31.255.255(172.16.0.0/12)
- 192.168.0.0–192.168.255.255(192.168.0.0/16)

Несколько особое значение имеет IP-адрес, который начинается со 127. Его используют при взаимодействии процессов и тестировании программ в пределах одного устройства. Когда данные передаются по адресу 127.0.0.1, образуется «петля» (loopback). При этом не происходит передачи данных по сети, они сразу же возвращаются как только что принятые. При этом можно использовать не только 127.0.0.1, но и любой адрес из диапазона от 127.0.0.1 до 127.255.255.254.

Адреса сетей назначаются либо централизованно (сеть является частью Internet), либо произвольно (сеть работает автономно). Номера узлов в сети в обоих случаях назначаются по усмотрению администратора, не выходя из разрешенного для сети диапазона. Роль координатора в централизованном распределении IP-адресов ранее выполняла организация InterNIC (Internet Network Information Center). Однако в настоящее время в связи с ростом сети эта задача адресов стала очень сложной, и InterNIC передала часть своих полномочий другим организациям и крупным поставщикам интернет-услуг. Зачастую поставщики интернет-услуг получают диапазоны адресов у компании InterNIC, а затем распределяют их между своими абонентами.

Практически во всех операционных системах есть сборник консольных утилит которые позволяют показать состояние сети и стека TCP/IP а также проверить работоспособность сети на компьютере:

- ipconfig(windows)/ifconfig(linux) - утилита показывает состояние и настройки сети на текущем устройстве,
- ping - делает тестовые сетевые запросы по протоколу ICMP, и этим проверяется доступность сетевого устройства, на экран выводится время, затраченное на доставку пакета до узла,
- tracert(windows)/traceroute(linux) - утилита предназначена для печати сетевого маршрута, по которому происходит доставка пакетов,
- nslookup - утилита водит на экран отображение (resolving) доменных интернет адресов в числовые IP адреса.
- whois - утилита выводит на экран информацию о владельце доменного интернет имени.

## 4. 6.1.4. Доменные имена (DNS), URL-ссылки

Нам уже известно, что для идентификации узла в сети TCP/IP используют IP-адрес. Например, укажем в адресной строке браузера адрес <http://85.192.33.245>, мы попадем на страницу учебного портала IT ШКОЛЫ SAMSUNG. Однако пользователям не очень удобно работать с числовыми адресами. И что делать, если сайт необходимо перенести на другой сервер? Ведь это значит, что IP-адрес сменится и пользователи не смогут найти нашу страницу.

Для решения этих проблем в 1984 году была разработана система доменных имен (англ. DNS — Domain Name System), которая позволила установить в соответствие с определенными IP-адресами некоторые символьные имена, например, [myitschool.ru](http://myitschool.ru). Это значит, что мы можем менять IP-адрес и при этом доменное имя останется прежним.

В конфигурации доменных имен применяется древовидная иерархия. При этом имена, у которых совпадают старшие составные части образуют домен имен (domain). Например, имена [myitschool.ru](http://myitschool.ru), [yandex.ru](http://yandex.ru) и [mail.ru](http://mail.ru) входят в домен ru (все эти имена имеют одну общую старшую часть — ru). А вот сайт [www.samsung.com](http://www.samsung.com) в домен ru не входит, так как он находится в домене com.

Понятие «домен» следует трактовать в рамках определенного контекста, так как он имеет множество значений. Кроме вышеупомянутого значения в компьютерной литературе можно встретить домен коллизий, домены Windows NT и др. Объединяет эти термины то, что они предназначены для описания некоторого подмножества компьютеров с каким-либо определенным свойством.

Если один домен является составной частью другого домена, то его можно назвать поддоменом (subdomain). Так как домены конструируются по принципу древовидной структуры, поддомен в свою очередь также является доменом (поддерево является деревом). Зачастую поддомен называют по имени, отличительной от других поддоменов его старшей составляющей. Устройства, включенные в один домен, при этом могут иметь IP-адреса, принадлежащие к разным сетям.

В сети Internet корневой домен управляется компанией InterNIC. Для каждой страны назначаются домены верхнего уровня, следуя стандарту ISO 3166. Для обозначения стран используются двухбуквенные сокращения (например, ru — Россия, de — Германия и т. п.). Кроме того, домены верхнего уровня назначаются на организационной основе. Для различных типов организаций приняты следующие обозначения:

- com — коммерческие организации;
- edu — образовательные организации;
- org — некоммерческие организации;
- net — организации, поддерживающие сети;
- biz — организации, связанные с бизнесом.

Подробнее со списком доменов верхнего уровня можно ознакомиться [здесь](#).

После доменов верхнего уровня зачастую в доменных именах присутствуют поддомены. Каждый поддомен может администрироваться в отдельно взятых регионах, организация и т. п. Далее поддомен может быть разбит еще на несколько поддоменов и т. д.

Большинство продаваемых доменов — это домены второго уровня. Имя домена второго уровня регистрируется на конкретное лицо или организацию. Такие услуги оказывают специальные организации — регистраторы доменных имен. Домены уровнем ниже второго чаще всего не обладают особой ценностью и их можно арендовать бесплатно.

Ранее в доменных именах допускалось использование только символов латинского алфавита, цифр и дефиса. В настоящее время возможна регистрация домена, содержащего другие знаки, входящие в кодировку UNICODE. Например, в России закреплен домен рф, в котором можно зарегистрировать домены второго уровня.

Резюмируя вышеизложенное, можно сделать вывод о том, что в сети интернет используется две системы адресов: доменные имена и IP-адресация. Для определения соответствия между ними на специальных DNS-серверах хранятся таблицы пар IP-адрес — доменное имя. По запросу доменного имени они возвращают ответ в виде IP-адреса клиента или наоборот. При вводе в браузер доменного имени (адреса сайта) сначала на DNS-сервер отправляется запрос с целью определить IP-адрес сервера. В случае положительного результата направляется запрос на получение веб-страницы, при этом драйвер протокола IP использует именно полученный IP-адрес, а не доменное имя.

Следует заметить, что одному доменному имени может соответствовать несколько IP-адресов (и наоборот). Данный подход применяется для распределения нагрузки на популярные ресурсы (например, [www.google.com](http://www.google.com), [www.vk.com](http://www.vk.com) и т. д.).

Адрес имеет не только каждый компьютер в интернете, но и каждый документ. Для обозначения такого адреса используется английское сокращение URL — Uniform Resource Locator — универсальный указатель ресурса. Типичный URL-адрес состоит из четырех частей: протокола, имени сервера (или его IP-адреса), каталога и имени документа (файла). Например, адрес

<http://myitschool.ru/is/Education/Markbook.aspx>



включает:

- протокол HTTP — протокол для обмена гипертекстовыми документами (это веб-страница);
- доменное имя сервера myitschool.ru;
- каталог на сервере /is/Education/;
- имя файла Markbook.aspx.

Иногда каталог и имя файла не указывают, например: <http://myitschool.ru>. Это означает, что мы обращаемся к главной странице сайта. Она может иметь разные имена в зависимости от настроек сервера (чаще всего — index.htm, index.html, index.php).

## 5. 6.1.5 HTTP-протокол

Сегодня любой потребитель интернет-ресурсов сталкивается с HTTP-протоколом. Этот сетевой протокол, будучи разработанным в 90-х годах, и до сегодняшнего дня является основным для передачи информации в мировой сети. **HTTP (HyperText Transfer Protocol)** — протокол передачи гипертекста, протокол прикладного уровня по классификации ISO-OSI. Сегодня чаще всего используется версия 1.1 протокола. Полное описание протокола можно увидеть в документе RFC 2616.

При взаимодействии в интернете участников взаимодействия обычно разделяют на две роли — клиент и сервер. Где сервер (serv — обслуживание) — поставщик информации/услуг, а клиент — потребитель. HTTP также предполагает клиент-серверное взаимодействие. То есть программа-клиент (например, браузер) отправляет HTTP-запрос, сервер его принимает, обрабатывает и отправляет клиенту HTTP-ответ. Таким образом, взаимодействие клиента и сервера по HTTP похоже на обмен почтовыми посылками через службу доставки.



Несмотря на то что в WWW протокол используется для передачи текстовой и медиаинформации, многие API используют его для передачи данных. При этом данные чаще всего кодируются как строковые, или в более универсальном представлении как JSON/XML. Рассмотрим, каким образом происходит обмен между клиентом и сервером по HTTP. При этом каждый запрос порождает следующую последовательность шагов.

1. DNS-запрос — поиск ближайшего DNS-сервера, чтобы из URL (например, yandex.ru) получить реальный IP-адрес.
2. Установка TCP-соединения с сервером по полученному IP-адресу.
3. Отправка данных HTTP-запроса.
4. Ожидание ответа, пока пакеты дойдут до сервера, он их обработает и вернет ответ назад.
5. Получение данных HTTP-ответа. Первые две операции, как правило, занимают больше всего времени.

### Структура запроса

Каждый HTTP-запрос состоит из трех частей, которые следуют в указанном порядке:

- Строка запроса — указан метод запроса (HTTP-метод), URI, версия протокола.
- Заголовки — характеризуют тело сообщения, параметры передачи и прочие сведения.
- Тело запроса — данные сообщения.

HTTP-спецификация определяет более строгое описание структуры запроса или ответа:

```
message = <start-line>
          *(<message-header>)
          CRLF
          [<message-body>]
<start-line> = Request-Line | Status-Line
<message-header> = Field-Name : Field-Value
```

CRLF — обязательная «новая» строка между секцией заголовка и телом. Заголовок (message-header) и тело (message-body) запроса может отсутствовать, но строка запроса (start-line) есть всегда.

### Строка запроса. Методы HTTP.

Строка запроса состоит из трех разделов, разделенных пробелом — Method URI Protocol. Рассмотрим пример запроса, который может использовать браузер при обращении к странице <http://myitschool.ru/book> - GET /book HTTP/1.1. В этом примере через пробел указываются метод GET, URI (нужный раздел сайта) и версию протокола:

Метод URI    Протокол

GET    /book HTTP/1.1

В поле метод указывается операция, которую нужно осуществить с указанным ресурсом. Сервер может выполнять тот набор методов, который в нем запрограммирован. Однако в большинстве фреймворков реализованы следующие методы:

- GET — получить ресурс. При этом URI содержит информацию, позволяющую найти ресурс;
- POST — создать новый ресурс;
- PUT — обновить существующий ресурс;
- DELETE — удалить существующий ресурс.

Рассмотрим два наиболее распространенных метода более подробно.

Метод **GET** предназначен для получения требуемой информации и передачи данных в адресной строке. Удобство использования метода GET заключается в том, что адрес со всеми параметрами можно использовать неоднократно, сохранив его, например, в закладки браузера, а также менять значения параметров прямо в адресной строке.

Метод **POST** посылает на сервер данные в теле запросе. Это позволяет отправлять большее количество данных, чем доступно методу GET, поскольку у него установлено ограничение в 4 Кб. Большие объемы данных используются в форумах, почтовых службах, заполнении базы данных, при пересылке файлов и др. Таким образом между методами POST и GET следующие различия (см. табл. 5.1).

Характеристика	GET	POST
Тело запроса	Отсутствует	Содержит данные
Максимальный объем	255 символов 8 КБ	
Кэширование	Да	Нет

Кроме того, следует помнить, что передача дополнительных параметров у GET и POST запросов происходит по-разному. У POST-запроса дополнительные параметры передаются в теле запроса, а у GET-запроса дополнительные параметры передаются в URL следующим образом: `URI[?param1=value1[&paramN=valueN]]` Например, если в браузере набрать адрес <https://www.google.ru/search?q=myitschool+Samsung&ie=utf-8>, то браузер отправит серверу google.ru следующий GET-запрос:

```
GET /search?q=myitschool+Samsung&ie=utf-8 HTTP/1.1
```

Параметры этого запроса следующие:

Имя параметра	Значение
q	myitschool+rostov
ie	utf-8

Не следует передавать в параметрах GET-запросов конфиденциальных данных, так как они присутствуют в запросе в открытом виде и будут легкодоступны посторонним. Например, в истории просмотра браузера или в каких-либо других журналах.

Заголовки запроса

Заголовки запроса задают его параметры. В зависимости от назначения заголовки HTTP-запросов разделяют на 4 группы:

- general;
- request specific;
- response specific;
- entity.

Из General-заголовков (применяются как для запроса, так и ответа сервера) наиболее распространенные:

- Date указывает дату запроса, пример: Date: Fri, 29 Jun 2016 09:20:45 GMT;
- MIME-version указывает версию MIME (по умолчанию 1.0), пример: MIME-version: 1.0;
- Pragma содержит указания для таких промежуточных агентов, как прокси и шлюзы, пример: Pragma: no-cache.

Request-заголовки:

- Host содержит доменное имя вебсервера, например: Host: [www.google.ru](http://www.google.ru);
- Authorization содержит информацию об аутентификации, например: Authorization: Basic QWxhZGRpbjpvGVuHnI2FtZQ==;
- From — поле, в котором браузер может посылать полный e-mail адрес пользователя серверу, например: From: info@myitschool.ru;
- User-Agent указывает наименование и версию браузера, ОС, например: User-Agent: Mozilla/3.0.

HTTP сегодня считается небезопасным, так как не предполагает использование шифрования при передаче информации. Однако для HTTP есть распространенное расширение, которое реализует упаковку передаваемых данных в

криптографический протокол SSL или TLS. Название этого расширения — HTTPS (HyperText Transfer Protocol Secure). HTTPS широко используется для защиты информации от перехвата, а также, как правило, обеспечивает защиту от атак вида man-in-the-middle. На данный момент HTTPS поддерживается всеми популярными веб-браузерами и большинство интернет ресурсов уже переключились с HTTP на HTTPS.

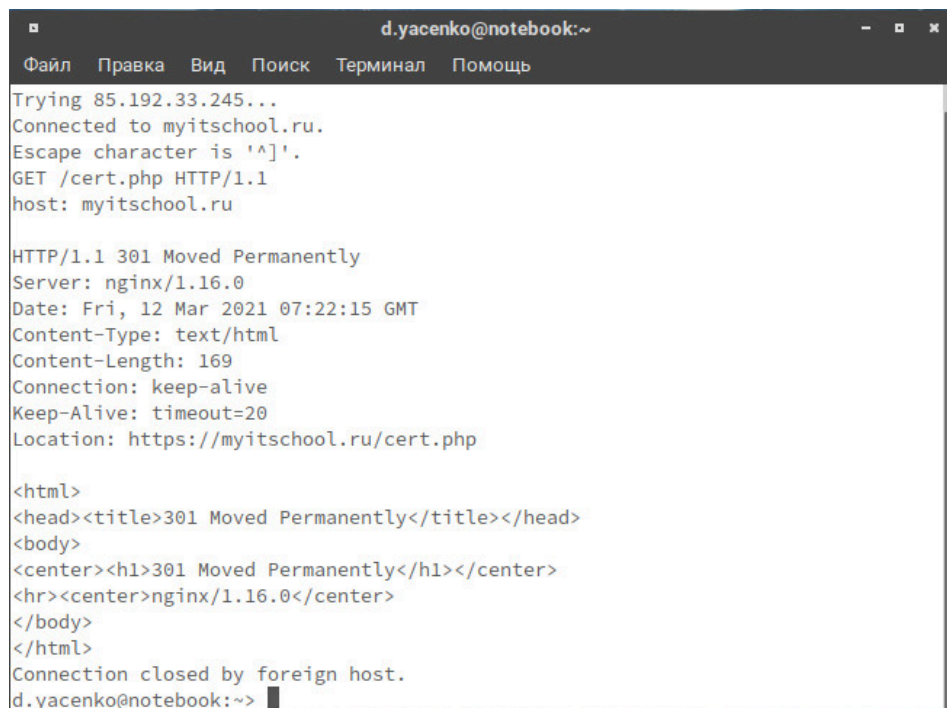
### Пример 6.1.3

Хороший способ познакомиться с протоколом HTTP — это поработать с каким-нибудь веб-ресурсом вручную. Для этого необходимо воспользоваться любой подходящей утилитой командной строки, например, telnet. В ОС Windows, возможно, придется ее установить. Как это сделать описано здесь: <http://windows.microsoft.com/ru-ru/windows/telnet-faq>.

В качестве примера симулируем запрос браузером страницы IT School Samsung — <http://myitschool.ru/cert.php>. Для этого откроем окно командной оболочки (в Windows WIN+r -> cmd -> ). В открывшемся окне оболочки запустим команду telnet myitschool.ru 80. В открывшемся соединении к серверу введем следующие строки HTTP протокола:

```
GET /cert.php HTTP/1.1 <Enter>
host: myitschool.ru <Enter>
<Enter>
```

В рассматриваемом случае ответ сервера будет код 301 и в консоль будет выведен небольшой HTML-документ. Для закрытия соединения необходимо ввести CTRL+] и далее exit.



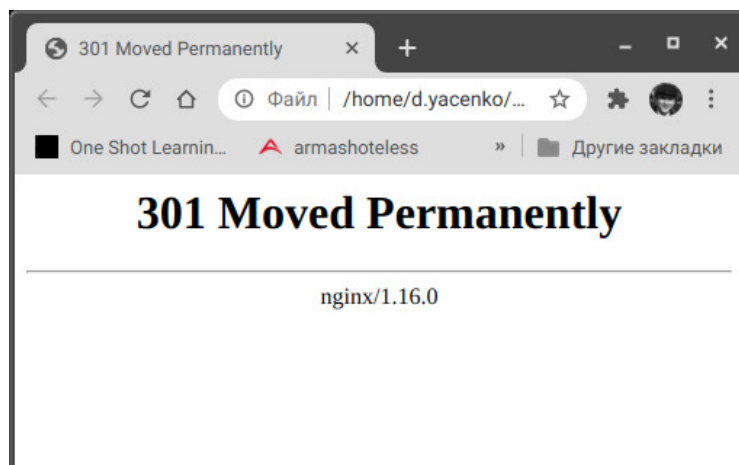
```
d.yacenko@notebook:~
Файл  Правка  Вид  Поиск  Терминал  Помощь

Trying 85.192.33.245...
Connected to myitschool.ru.
Escape character is '^]'.
GET /cert.php HTTP/1.1
host: myitschool.ru

HTTP/1.1 301 Moved Permanently
Server: nginx/1.16.0
Date: Fri, 12 Mar 2021 07:22:15 GMT
Content-Type: text/html
Content-Length: 169
Connection: keep-alive
Keep-Alive: timeout=20
Location: https://myitschool.ru/cert.php

<html>
<head><title>301 Moved Permanently</title></head>
<body>
<center><h1>301 Moved Permanently</h1></center>
<hr><center>nginx/1.16.0</center>
</body>
</html>
Connection closed by foreign host.
d.yacenko@notebook:~>
```

Если рассмотреть ответ сервера, то в его теле присутствует HTML-текст. Если этот текст просматривать в браузере, то он будет выглядеть следующим образом:



Полное описание протокола HTTP можно прочитать в документе [RFC2616](#) на сайте открытого международного сообщества Internet Engineering Task Force (IETF)

[Начать тур для пользователя на этой странице](#)