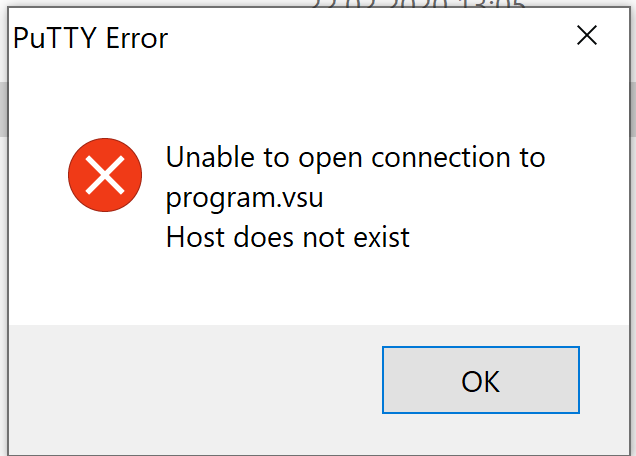
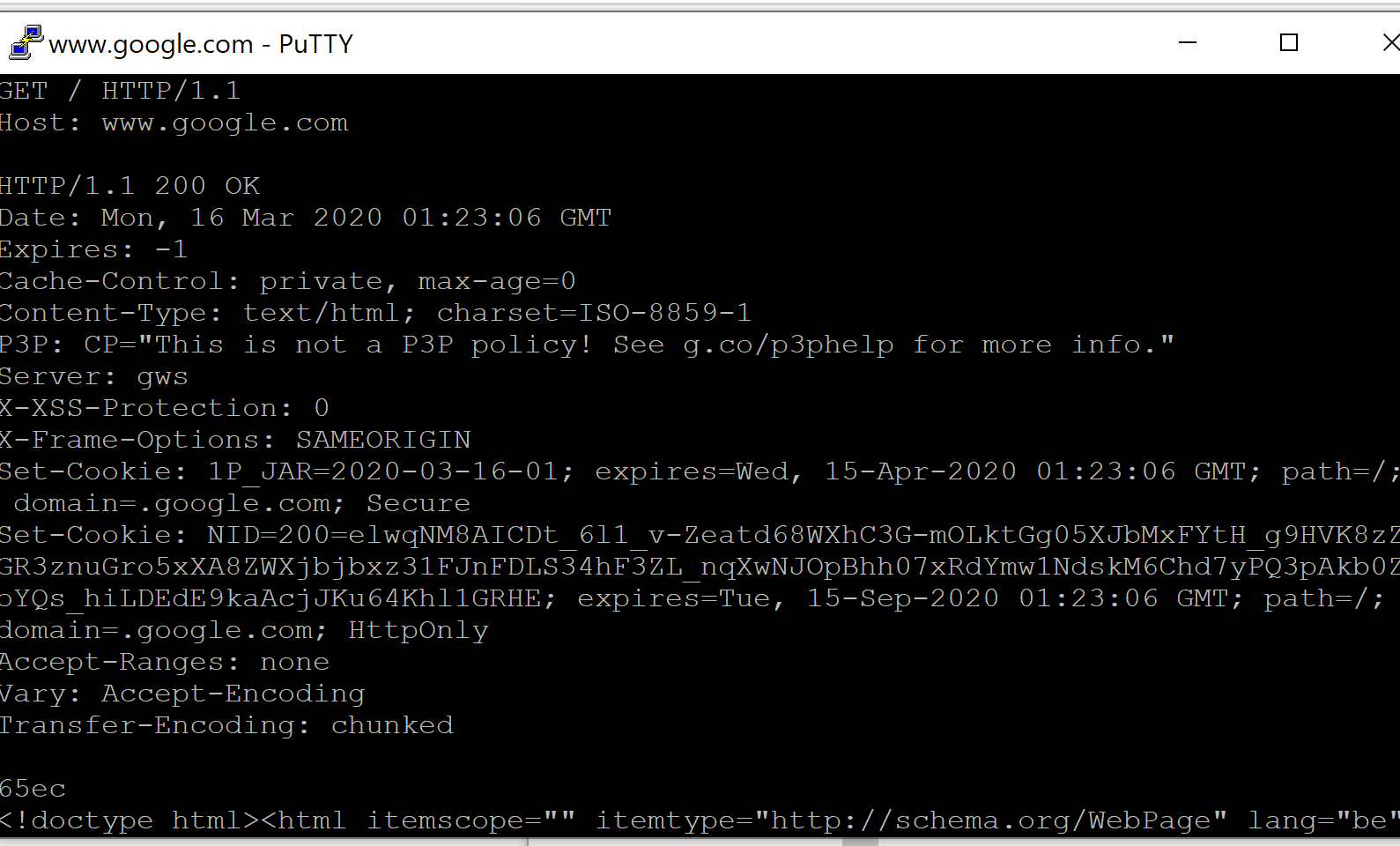


302 – Означает, что запрошенный ресурс был временно перемещён по адресу, указанному в заголовке Location.

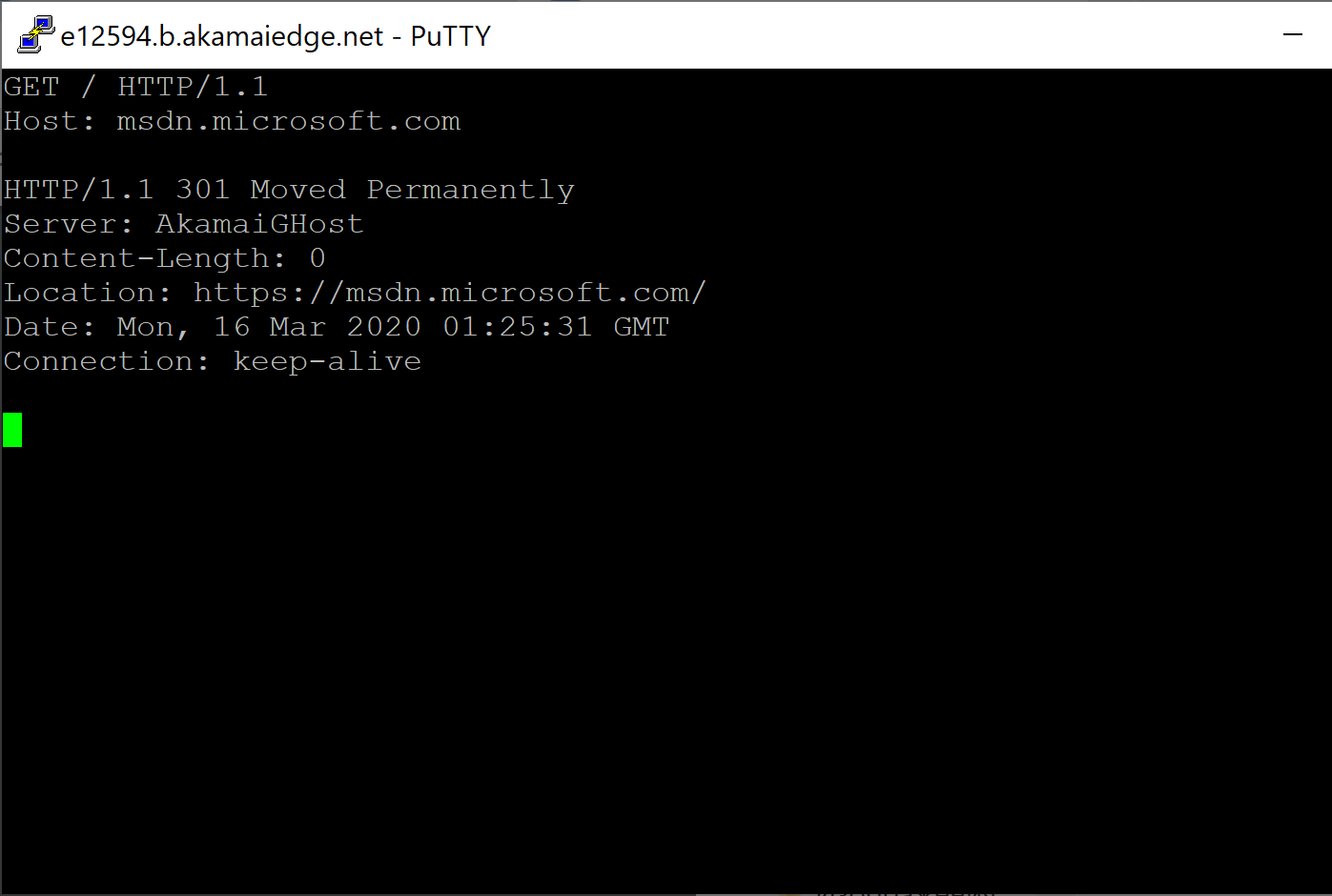


Код состояния HTTP 301 или «Перемещение навсегда» - стандартный код ответа HTTP, получаемый в ответ от сервера в ситуации, когда запрошенный ресурс был на постоянной основе перемещён в новое местоположение, и указывающий на то, что текущие ссылки, использующий данный URL, должны быть обновлены. Адрес нового местоположения указывается в поле Location получаемого ответа.

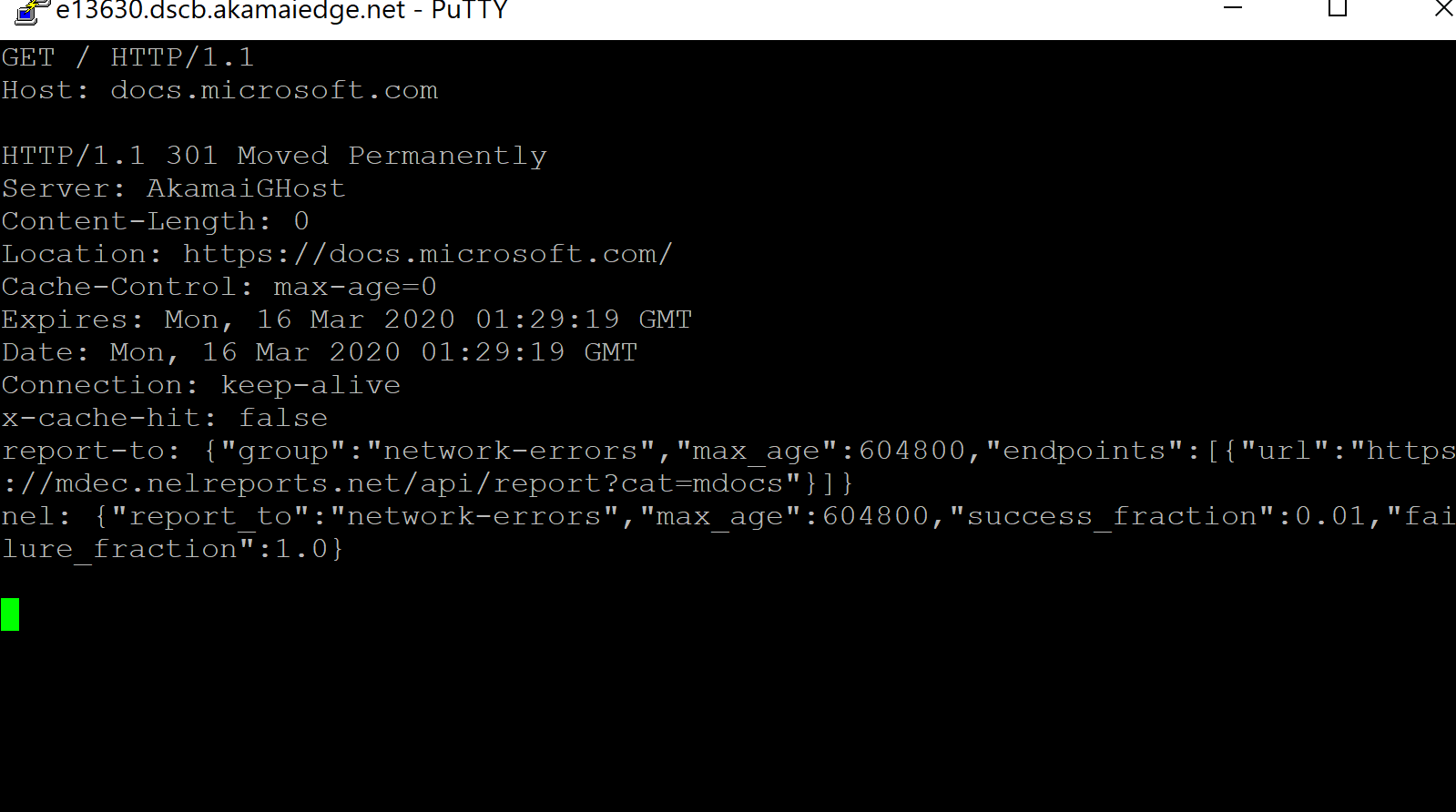


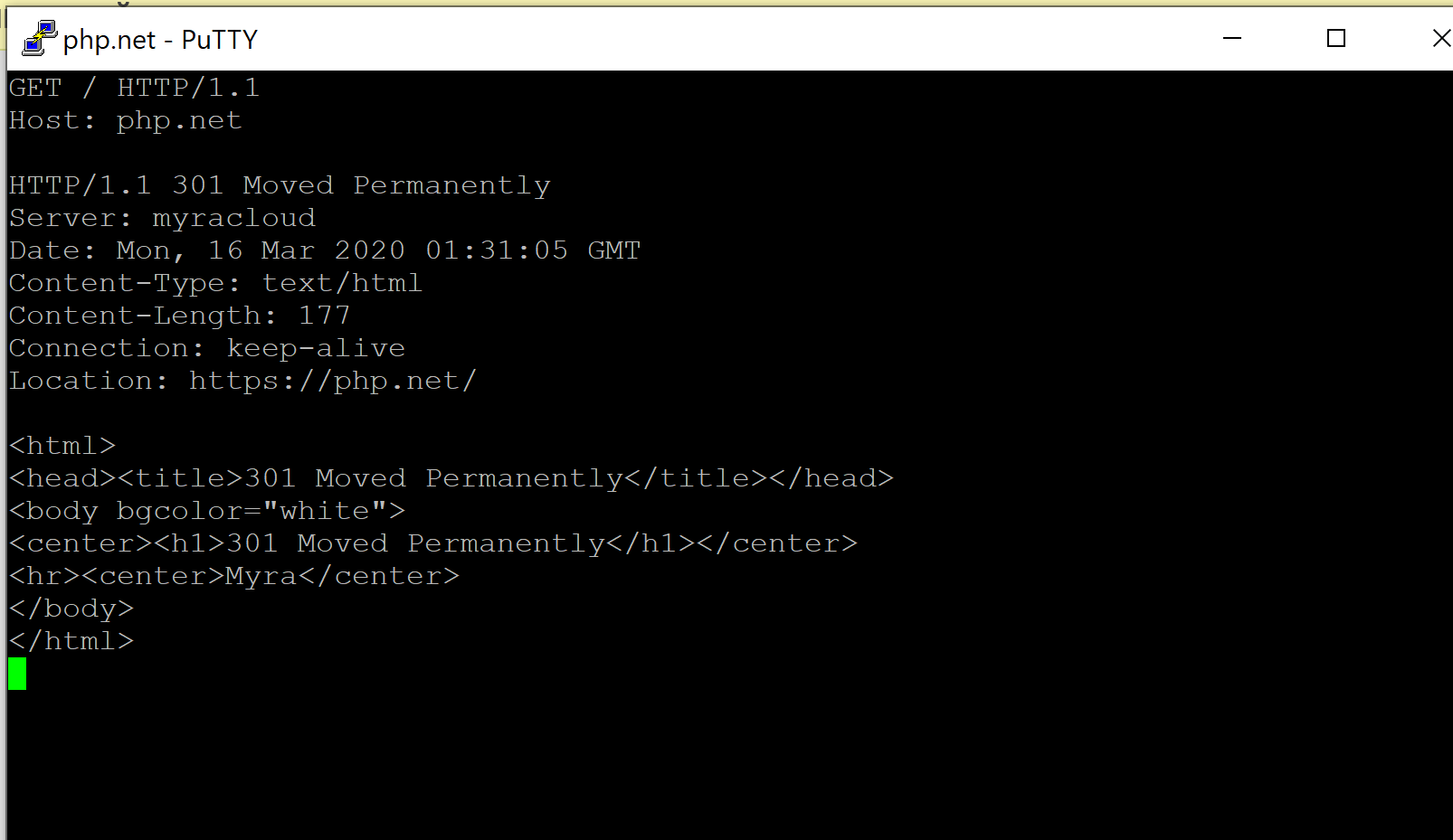


Код 200 ответа об успешном выполнении HTTP указывает на успешное выполнение запроса Ответ 200 кэшируется по умолчанию.



Код состояния HTTP 301 или Moved Permanently (с англ. — «Перемещено навсегда») — стандартный код ответа HTTP, получаемый в ответ от сервера в ситуации, когда запрошенный ресурс был на постоянной основе перемещён в новое месторасположение, и указывающий на то, что текущие ссылки, использующие данный URL, должны быть обновлены. Адрес нового месторасположения ресурса указывается в поле Location получаемого в ответ заголовка пакета протокола HTTP.





1) 6 основных методов: GET, HEAD, POST, OPTIONS, PUT, DELETE; + PATCH, TRACE, LINK, UNLINK, CONNECT.

2) GET, HEAD – наиболее часто используемые, т.к. обязательны для поддерживания всеми серверами. GET – получение ресурса по URL (исполнение скриптов), HEAD – получение заголовка ответа сервера (тестирование гиперссылок и проверка доступа к ресурсам).

3) User-Agent: Mozilla/5.0 (X11; Linux i686; rv:2.0.1) Gecko/20100101 Firefox/4.0.1.

Host: webmail.pisem.net : 8080

Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ= =

From: user@pisem.net

Accept-Encoding: compress, gzip

Accept-Language: ru, en

If-Modified-Since: Mon, 1 Oct 2001 19:43:31 GMT

Referer: http://www.tut.by/default.htm

User-Agent: Mozilla/4.0 (compatible; MSIE 5.0; Windows 98)

4) Accept-Ranges: – если не “none”, то сервер позволяет производить частичную закачку файлов. Пример:

GET /z4d4kWk.jpg HTTP/1.1

Host: i.imgur.com

Range: bytes=0-1023

6) “Location” header field – Благодаря этому полю запроса можно определить (предположить), что запрос выполнен со страниц определенного ресурса.

7) Наиболее известными и распространенными веб серверами являются Microsoft IIS и Apache. Самым популярным из них является Apache. Его программное обеспечение используется более чем на 60% всех существующих. Во многом это стало возможным благодаря тому, что Apache является бесплатным. Разработчики постоянно анализируют ошибки своего продукта и устраняют их. Недостаток программы в том, что по большому счету она предназначена для профессиональных программистов и пользователей, поэтому обслуживать и настраивать ее достаточно тяжело.

Компания Microsoft, наоборот, свой веб сервер IIS стремится сделать доступным для понимания среднестатистического пользователя. Все его настройки можно провести без специалистов. Вполне возможно, что данная программа сможет догнать своего конкурента с индейским именем по числу пользователей.

И Apache и IIS отлично работают со многими операционными системами, например с Mac OS, Linux, различными модификациями Microsoft Windows и многими другими. Эти два веб сервера стабильны в работе, имеют хорошую скорость, к ним можно подключить дополнительные модули.

Apache Web Server, IIS Web Server, Nginx Web Server, Apache Tomcat, LiteSpeed web server.

8) 1xx информационный ответ - запрос получен, процесс продолжается

2xx успешно - запрос был успешно получен, понят и принят

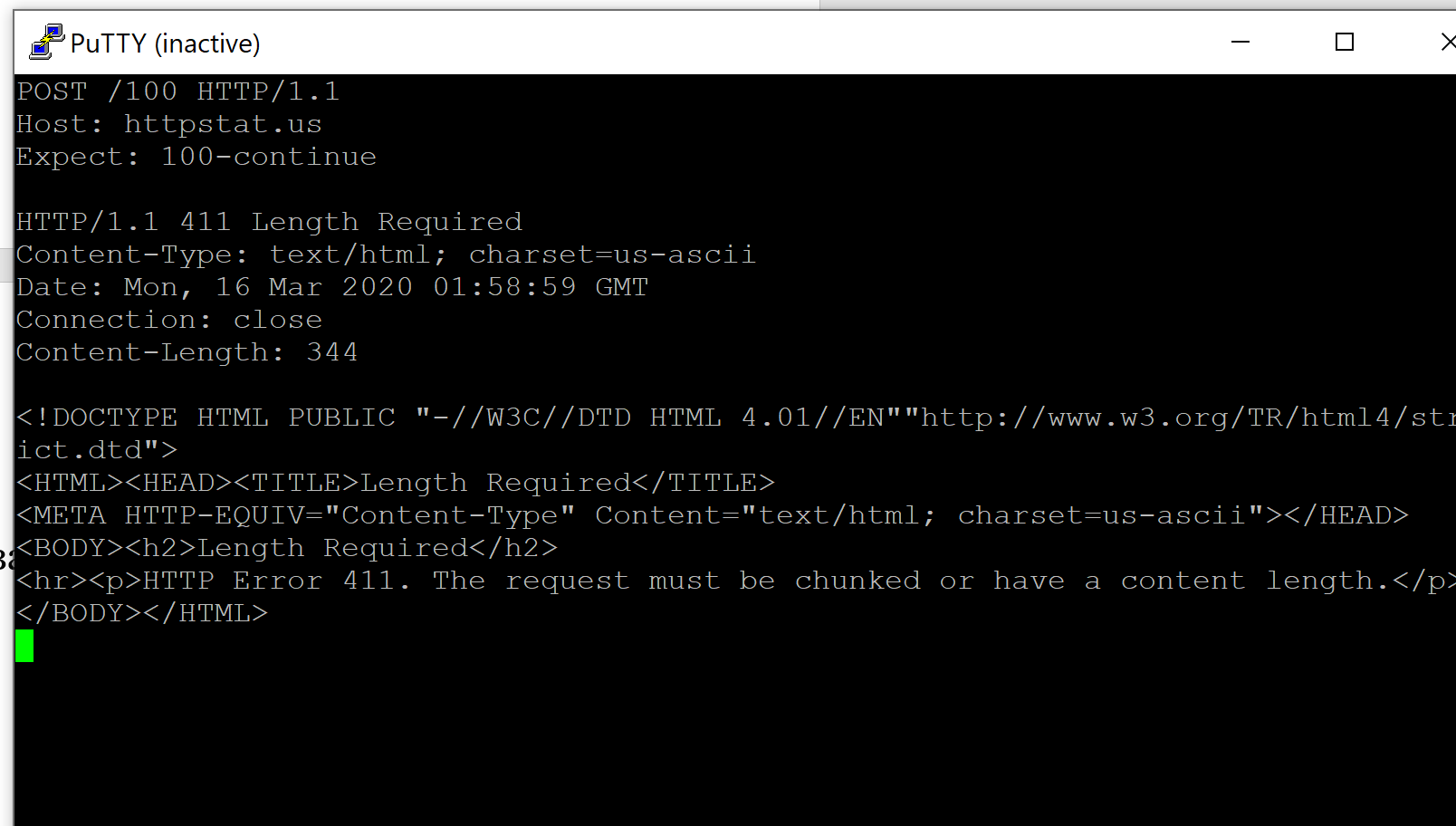
Перенаправление 3xx - для выполнения запроса необходимо предпринять дальнейшие действия

Ошибка клиента 4xx - запрос содержит неверный синтаксис или не может быть выполнен

Ошибка сервера 5xx - серверу не удалось выполнить явно допустимый запрос.

Ошибки клиентом

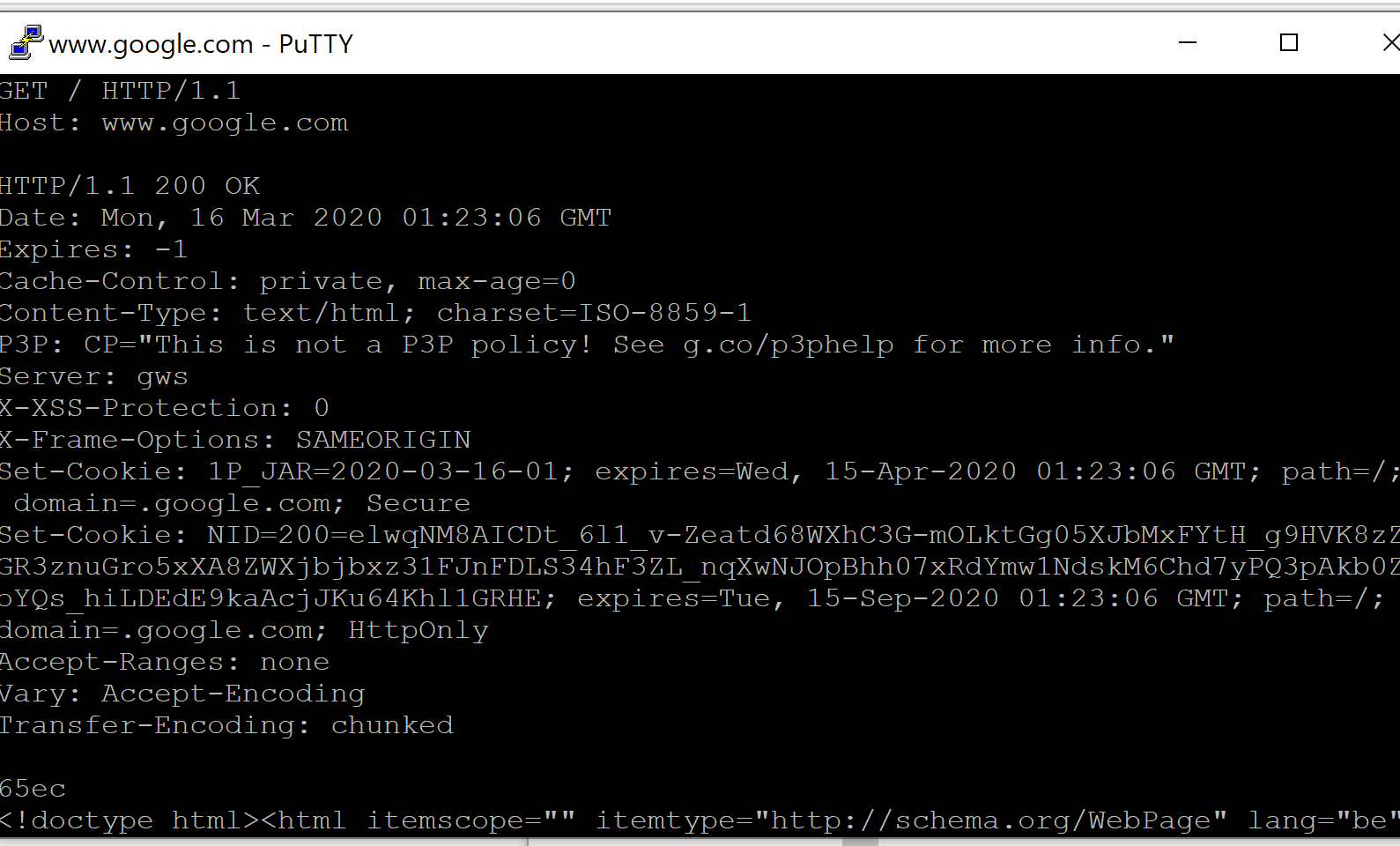
411 длина требуется – В запросе не указана длина его содержимого, которое требуется запрашиваемому ресурсу.

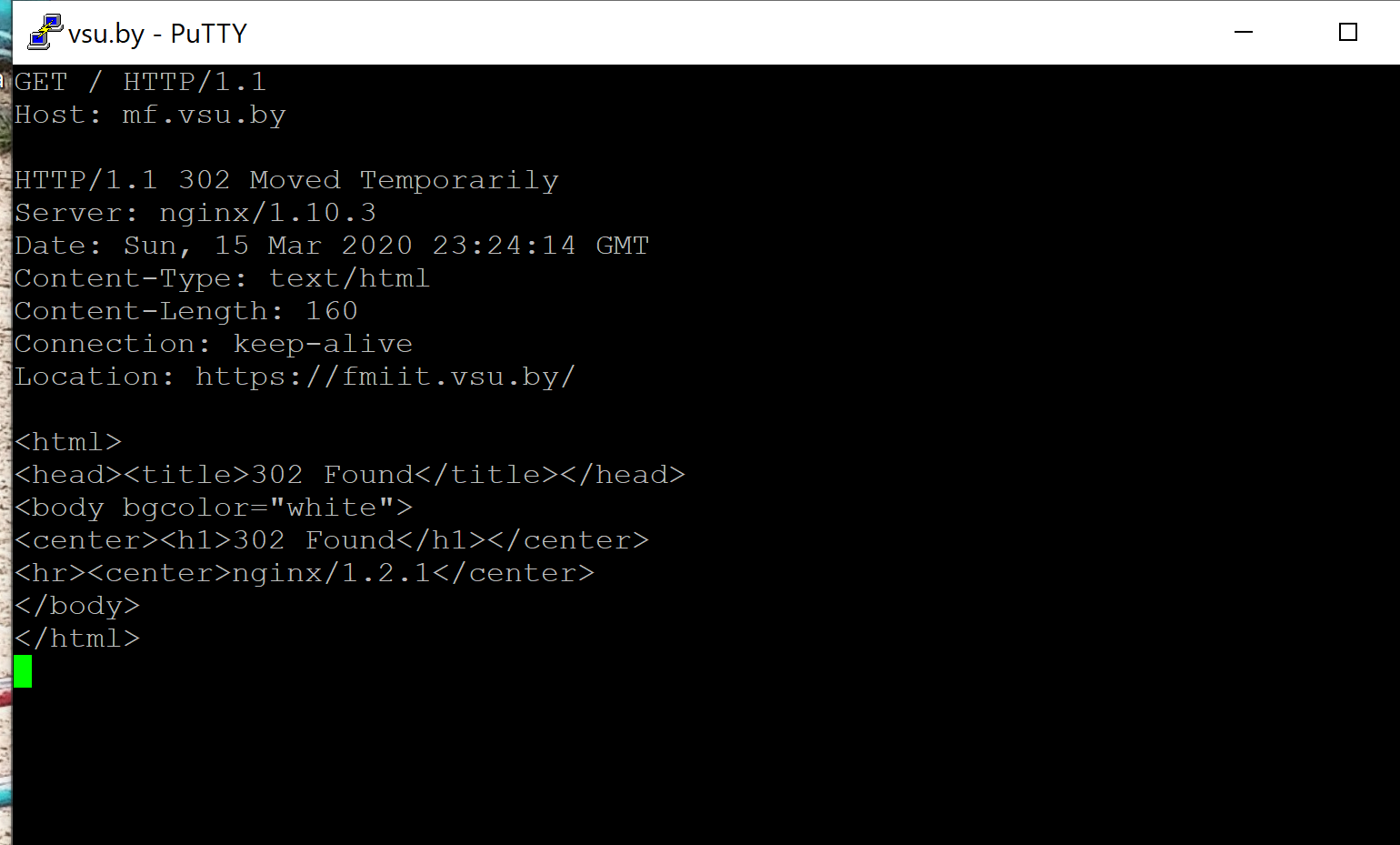


Информационный ответ

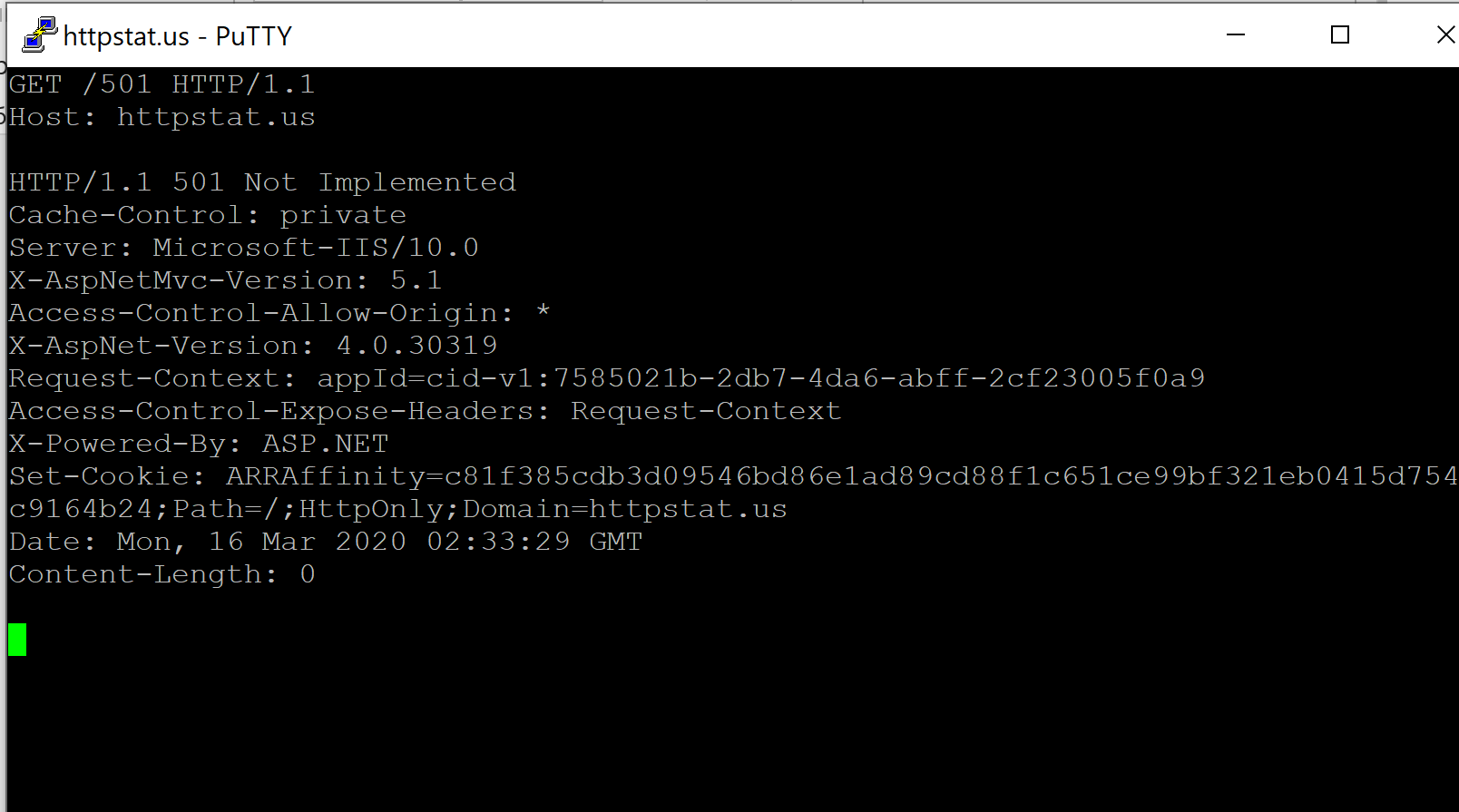


Код 200 ответа об успешном выполнении HTTP указывает на успешное выполнение запроса Ответ 200 кэшируется по умолчанию.



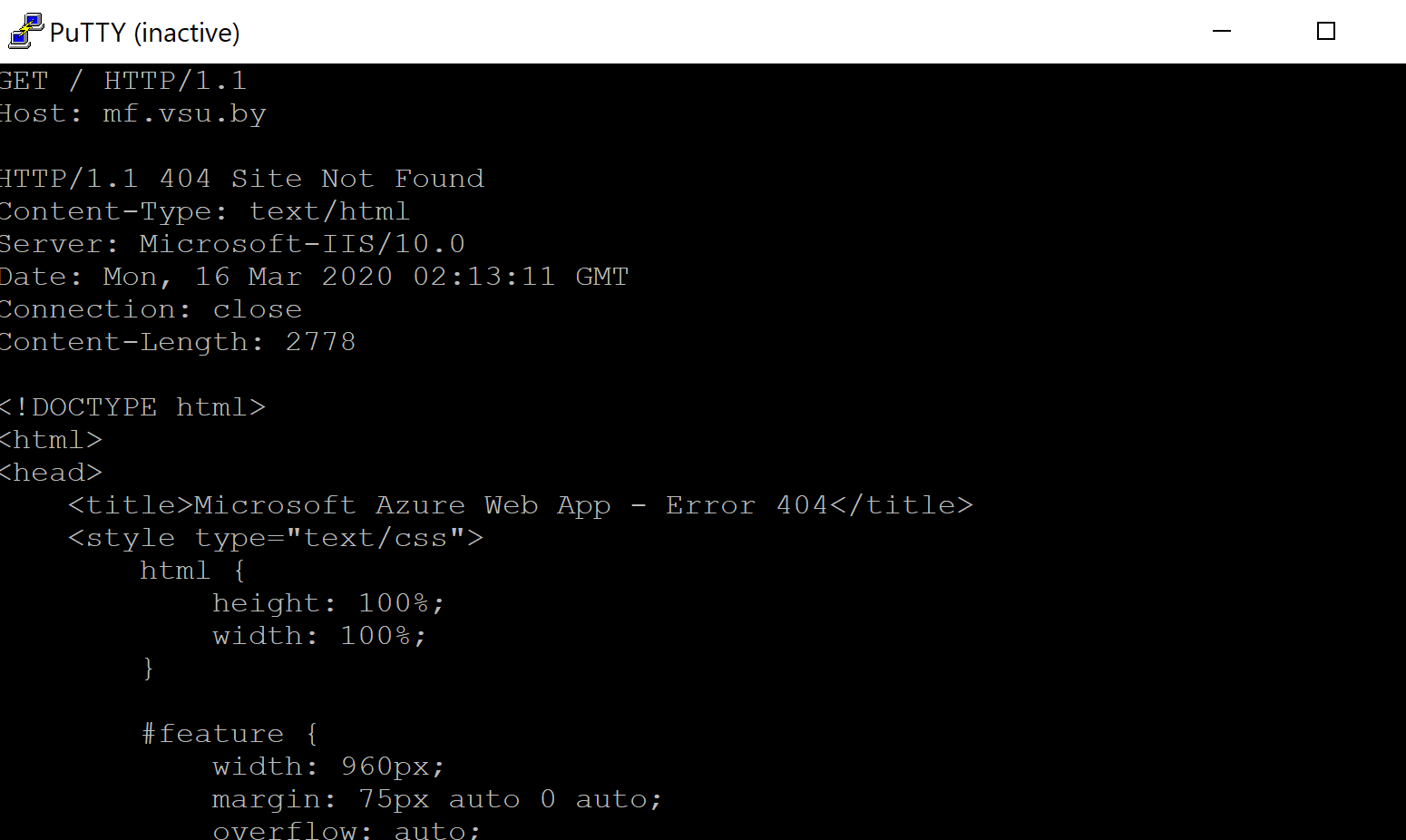


Код 302 – Переехал навсегда.

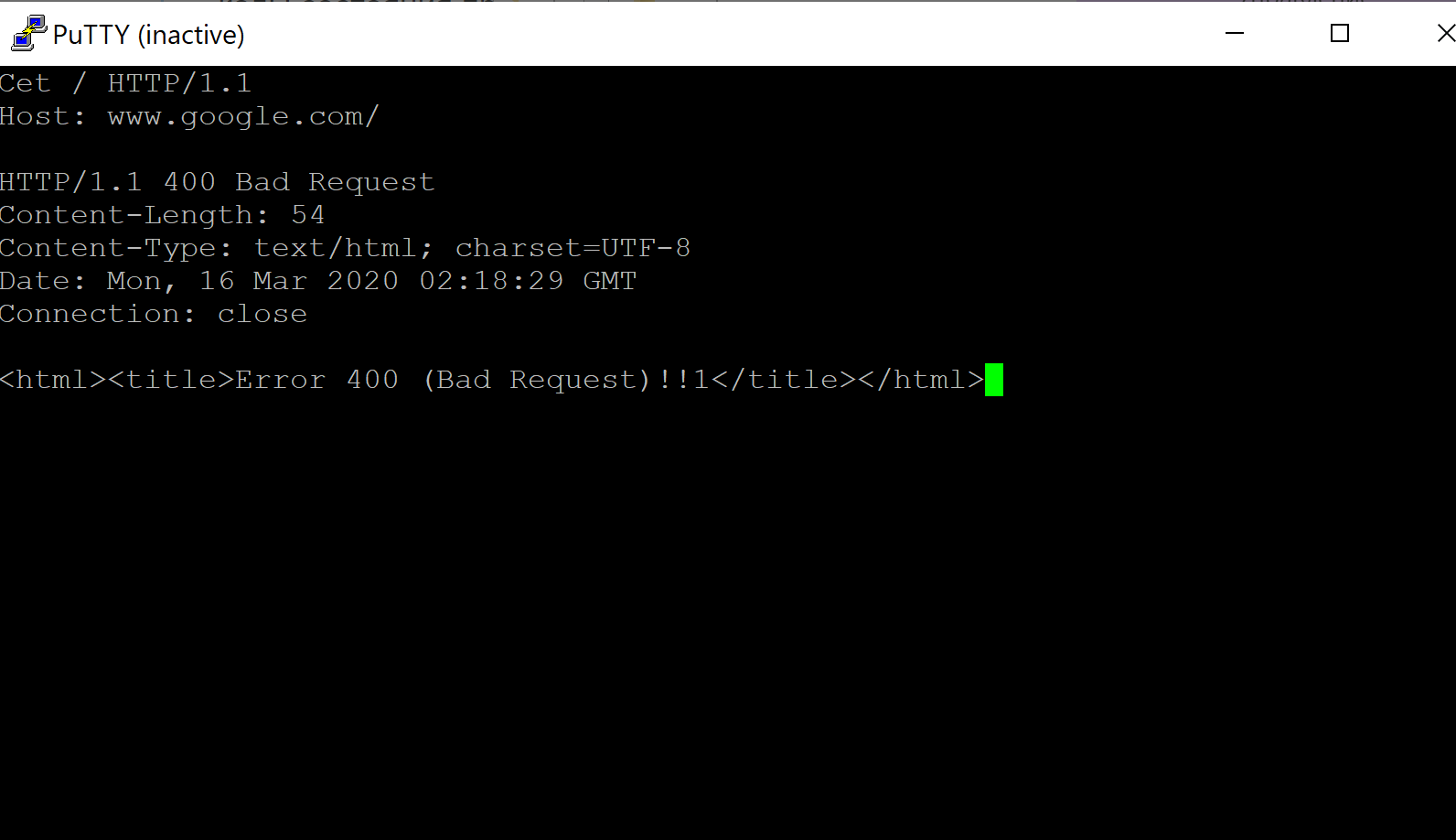


Код 501 - Сервер либо не распознает метод запроса, либо ему не хватает возможности выполнить запрос

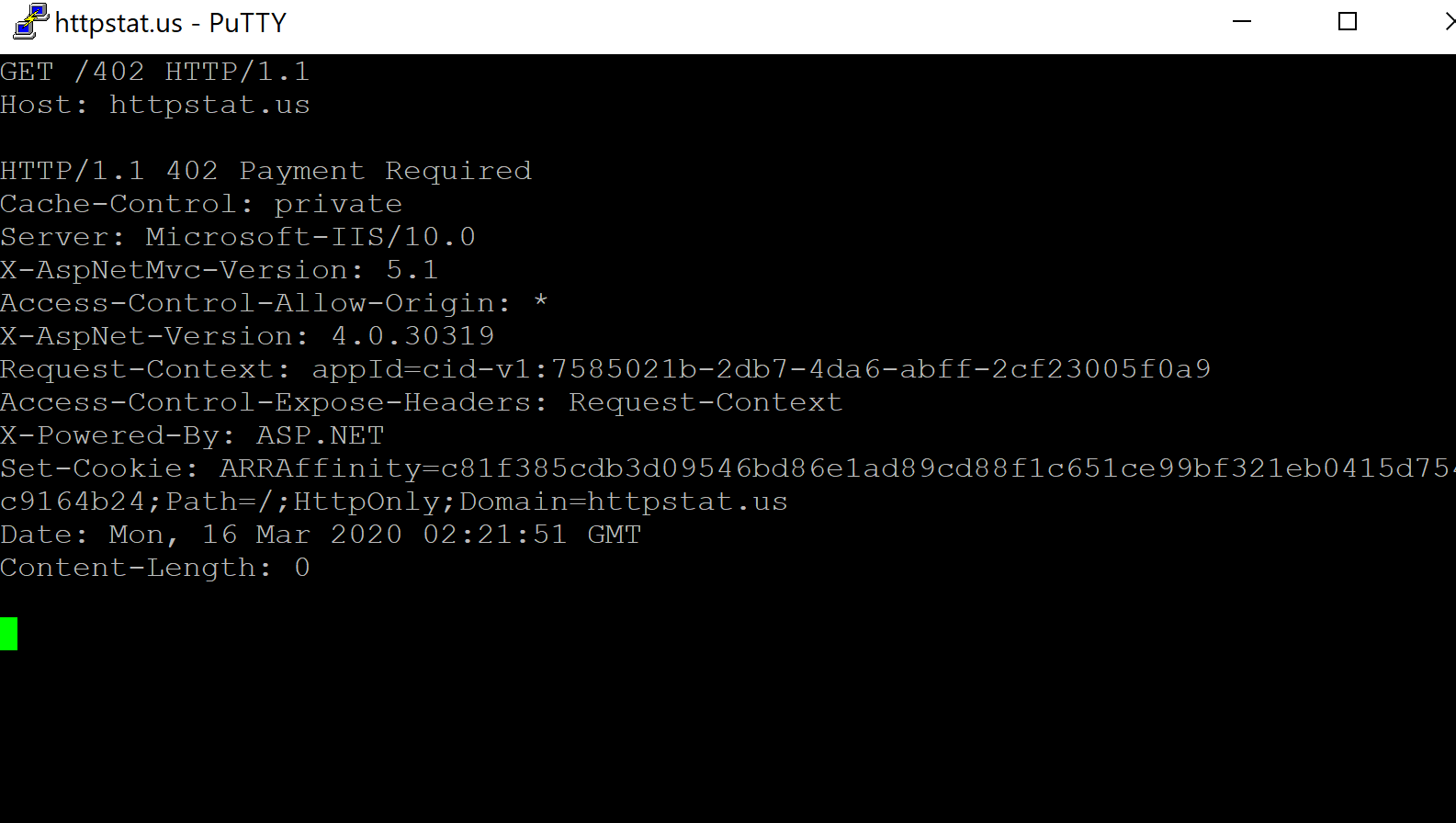
9) Код 404 – Ошибка 404 или Not Found («не найдено») — стандартный код ответа HTTP о том, что клиент был в состоянии общаться с сервером, но сервер не может найти данные согласно запросу. Ошибку 404 не следует путать с ошибкой «Сервер не найден» или иными ошибками, указывающими на ограничение доступа к серверу. Ошибка 404 означает, что запрашиваемый ресурс может быть доступен в будущем, что однако не гарантирует наличие прежнего содержания.



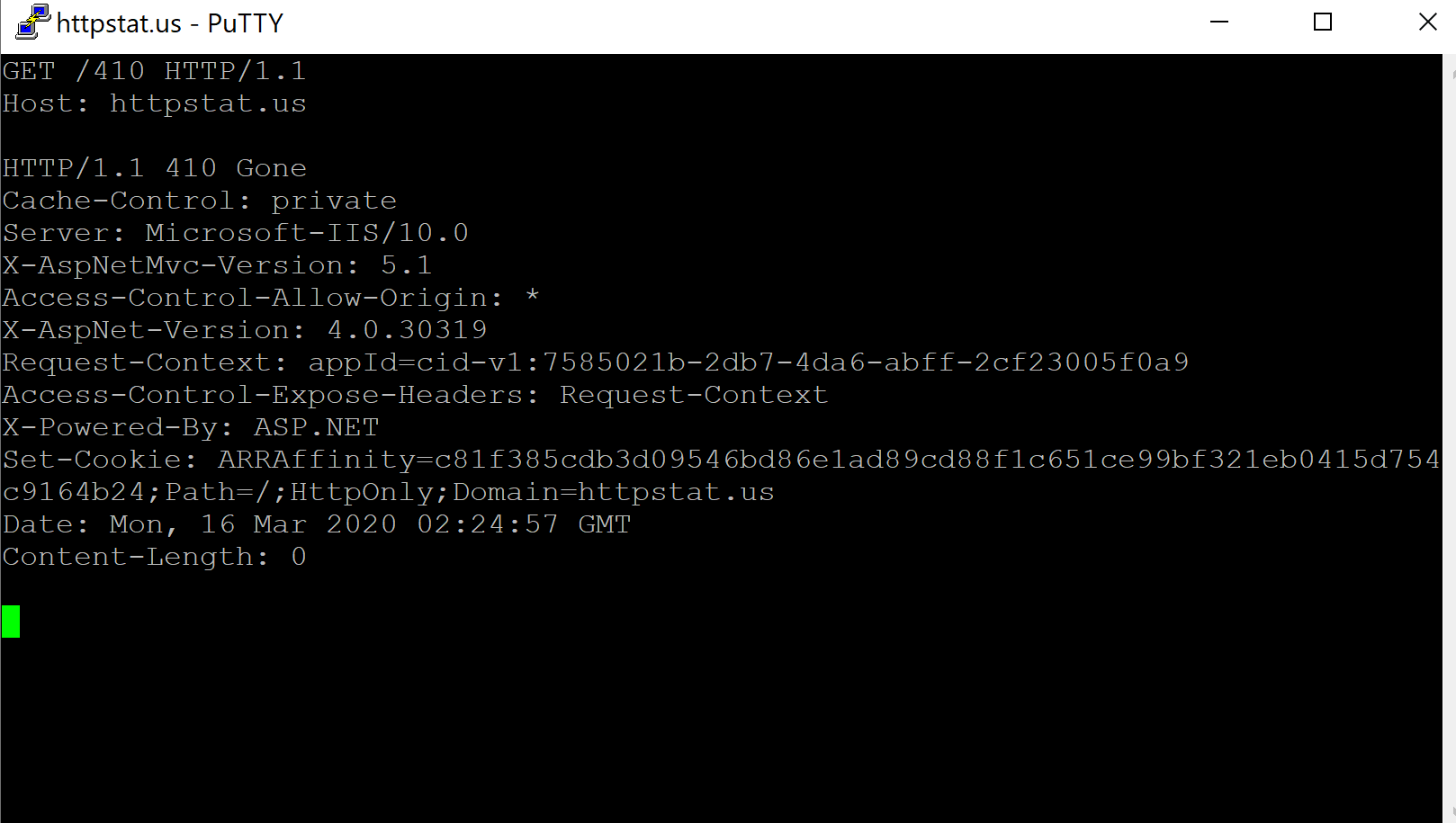
400 - Сервер не может или не будет обрабатывать запрос из-за явной клиентской ошибки (например, синтаксис неверно сформированного запроса, слишком большой размер, неверное формирование кадра запроса или обманчивая маршрутизация запроса).



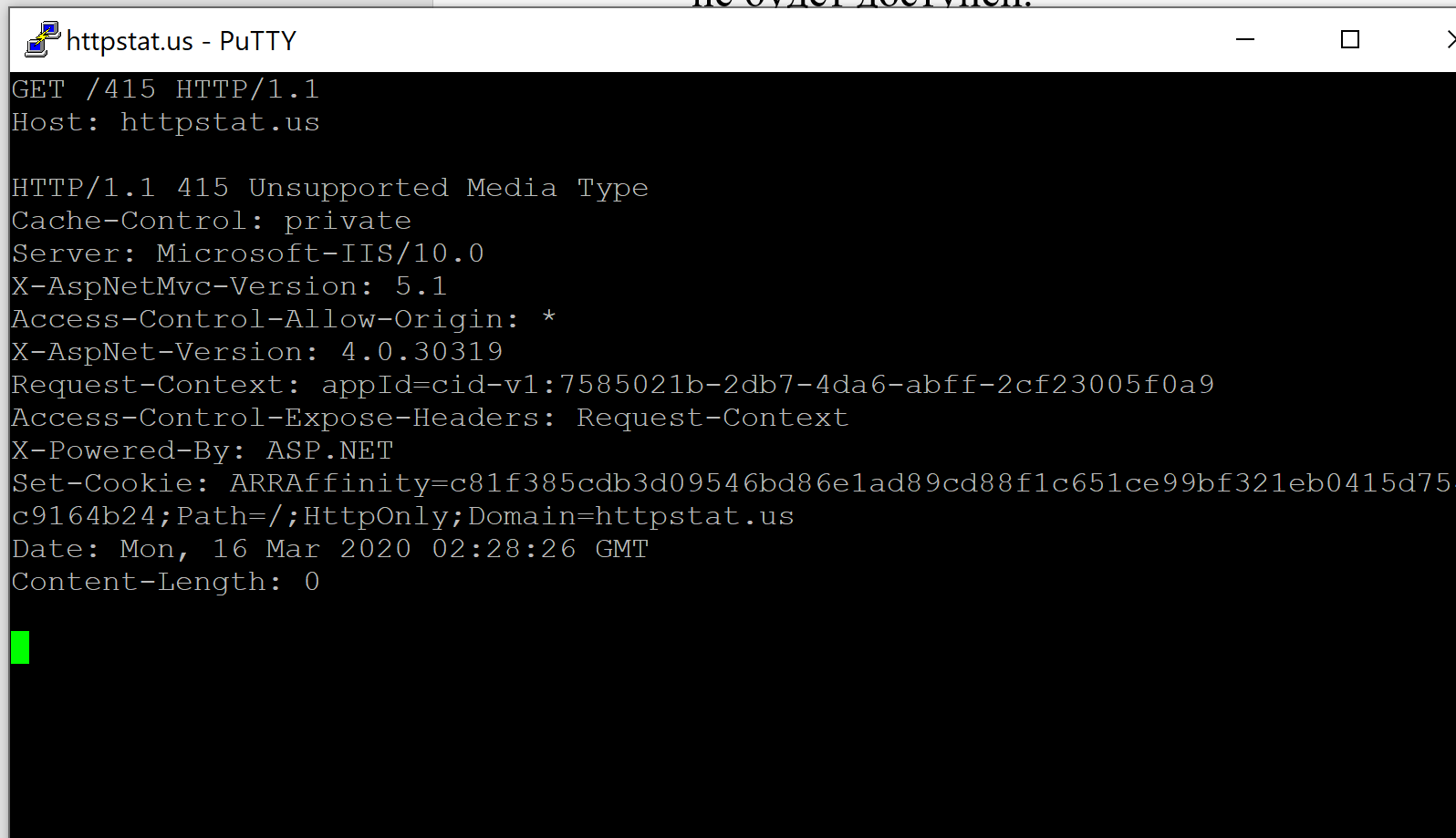
Код 402 Зарезервировано для будущего использования. Первоначально предполагалось, что этот код может использоваться как часть какой-либо формы цифровой наличности или схемы микроплатежей , как предложено, например, GNU Taler , [34], но этого еще не произошло, и этот код обычно не используется. Google Developers API использует этот статус, если конкретный разработчик превысил дневной лимит запросов.



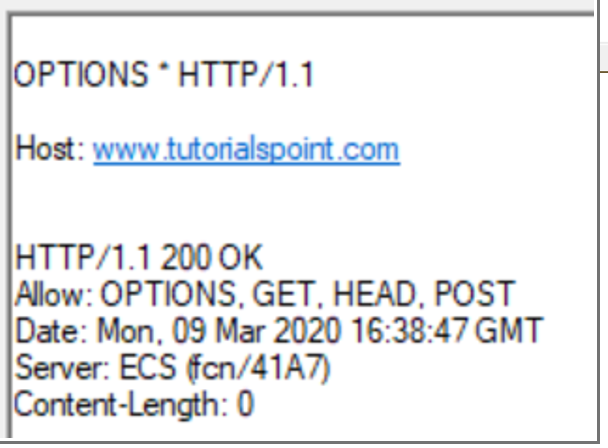
Код 410 - Указывает, что запрошенный ресурс больше не доступен и больше не будет доступен.



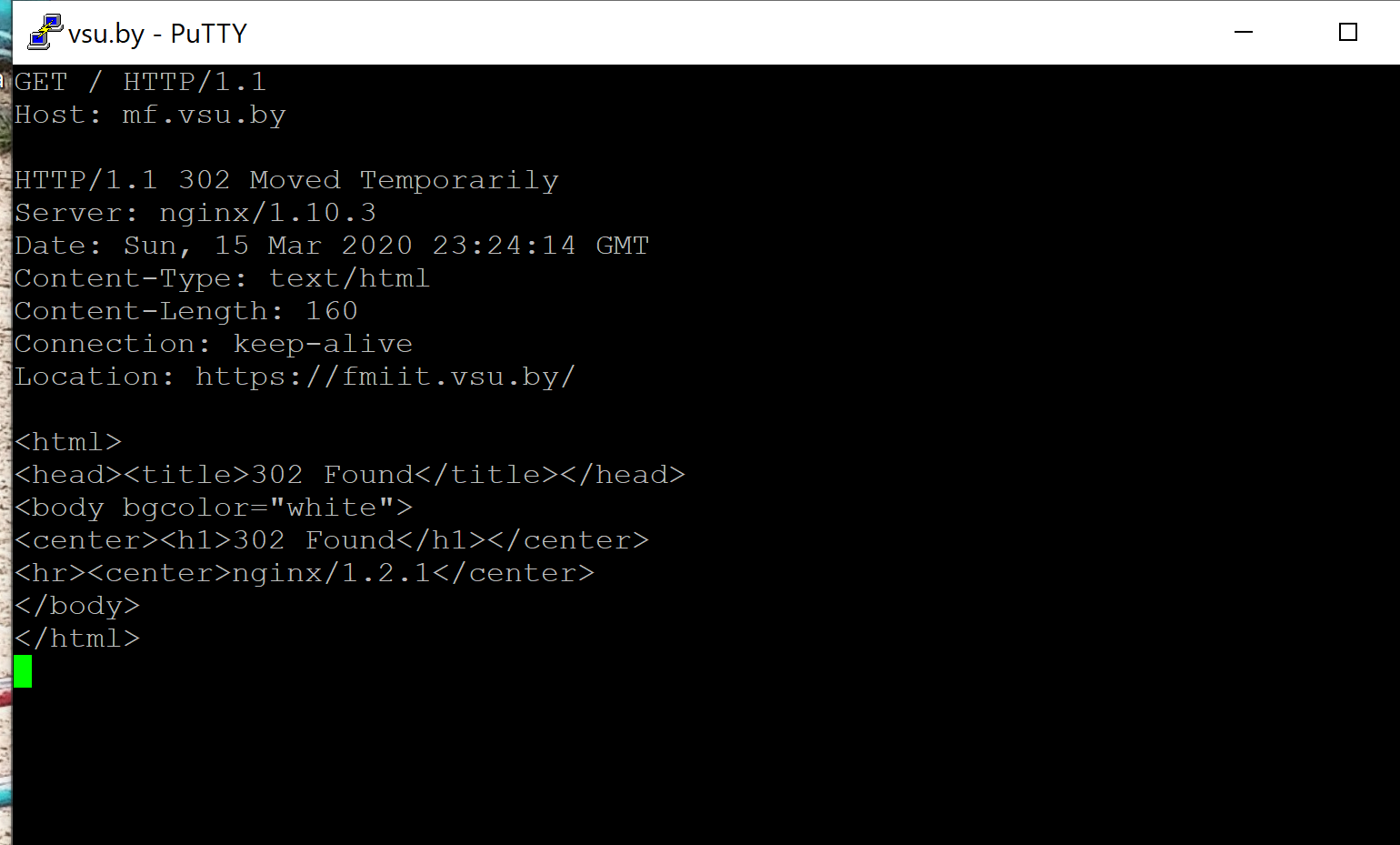
415 неподдерживаемый тип носителя



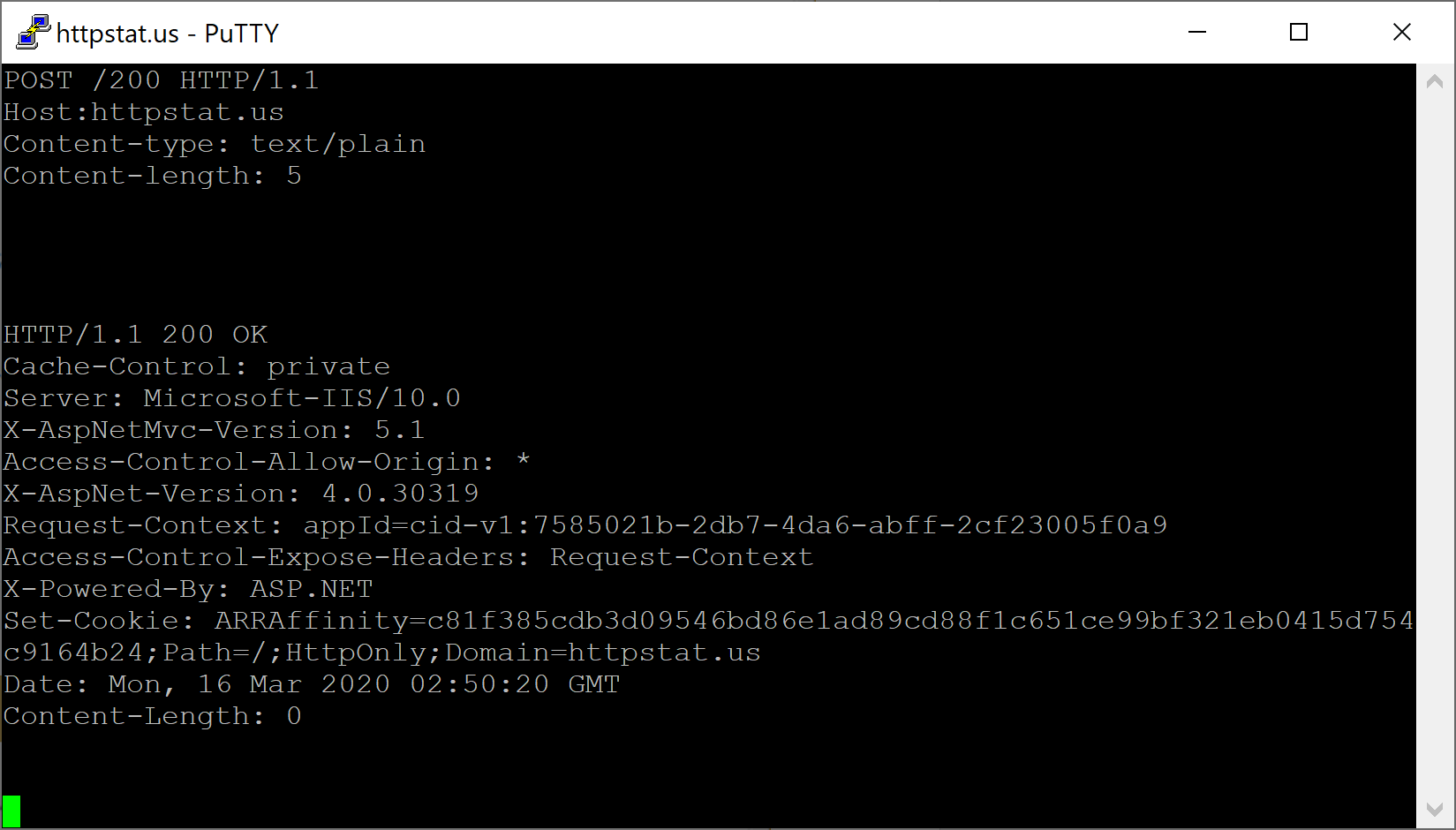
10) Метод ”OPTIONS” используется для определения возможностей Webсервера или проверки его работоспособности. Чаще всего в виде: «OPTIONS\* HTTP/1.1»

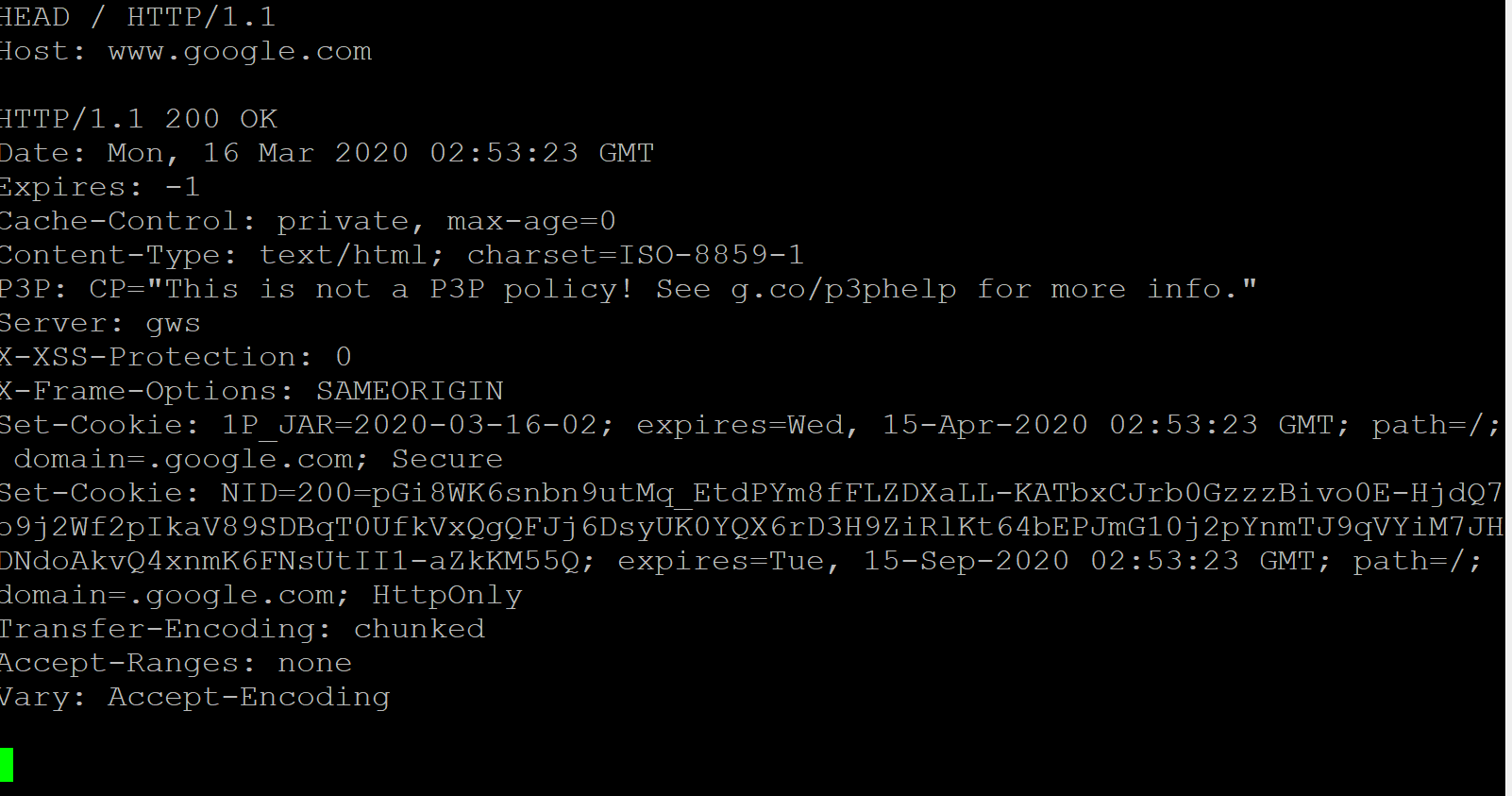


Метод Get



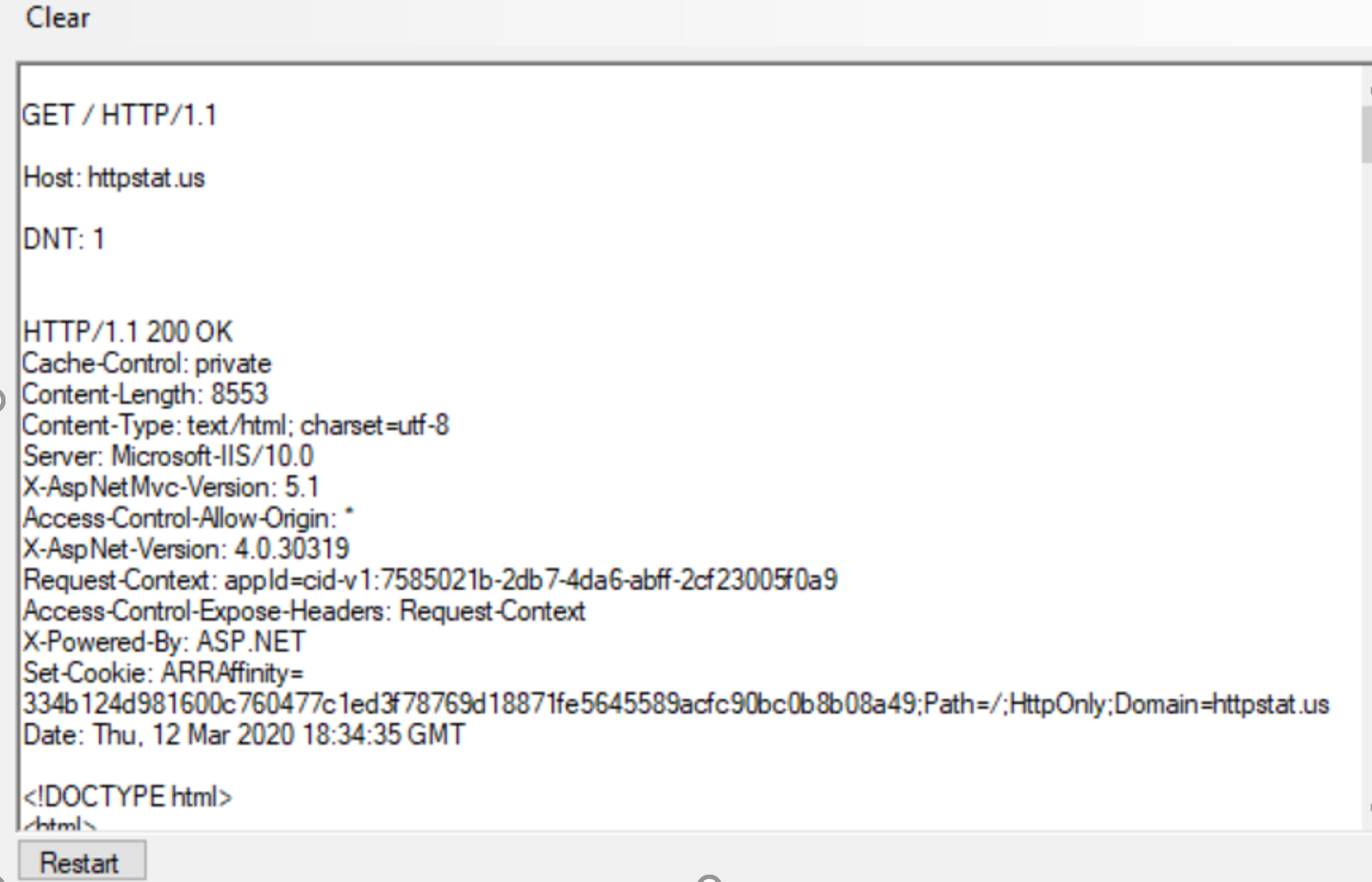
Метод Post



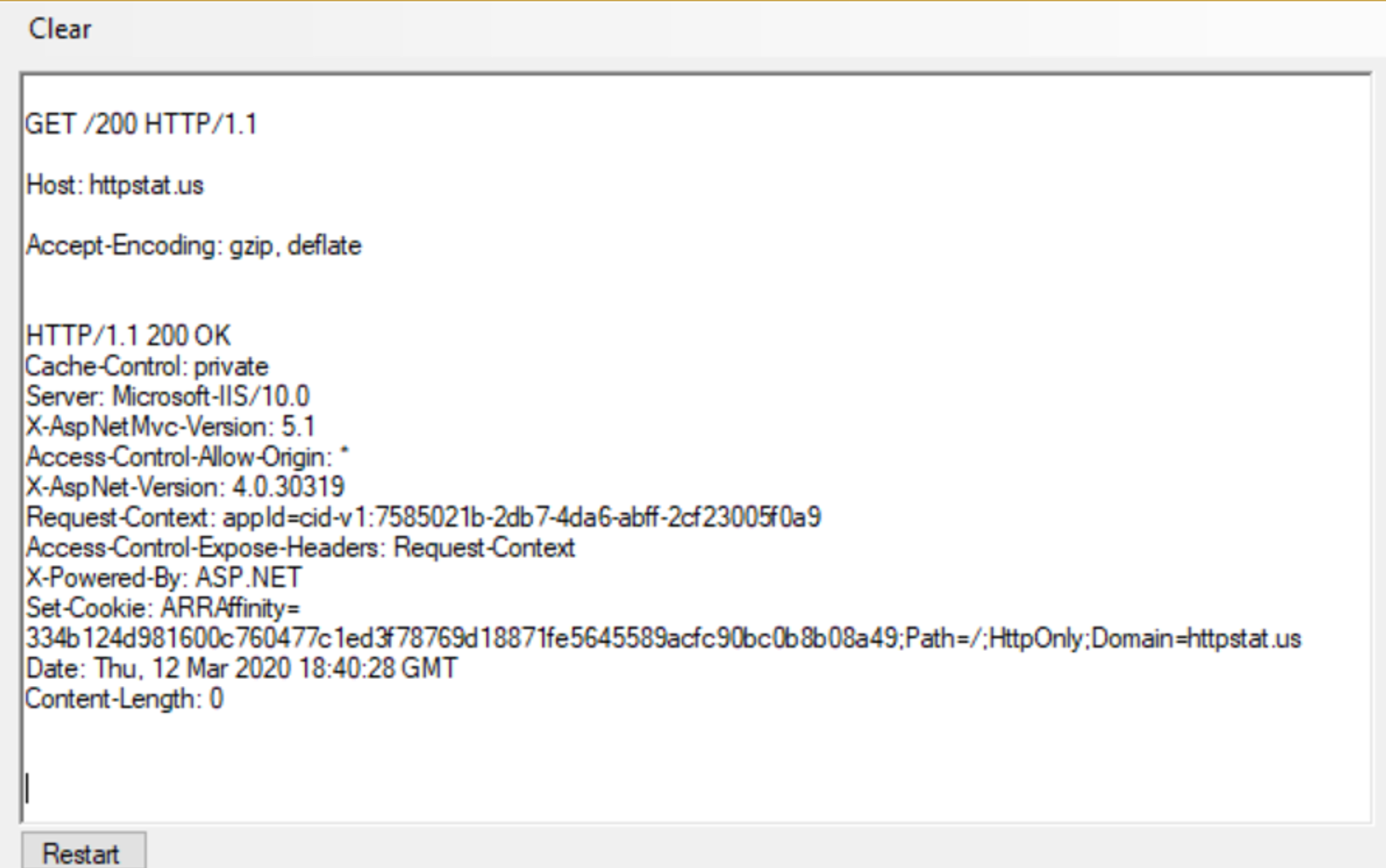


11) 





DNT( Д о N OT T стойки) заголовка запроса указывает на то предпочтение отслеживания пользователя. Это позволяет пользователям указывать, предпочитают ли они конфиденциальность, а не персонализированный контент.





5) В сфере вычислительной обработки данных кэш – это высокоскоростной уровень хранения, на котором требуемый набор данных, как правило, временного характера. Доступ к данным на этом уровне осуществляется значительно быстрее, чем к основному месту их хранения. С помощью кэширования становится возможным эффективное повторное использование ранее полученных или вычисленных данных.

no-cache и no-store

Директива no-cache означает, что при повторном запросе к тому же URL ответ можно использовать только после проверки изменений. Таким образом, если указан соответствующий маркер подтверждения (ETag), будет выполнена повторная проверка. Однако при отсутствии изменений данные не будут скачиваться ещё раз.

Действие директивы no-store намного проще. Она запрещает браузеру и всем промежуточным кешам сохранять какую-либо версию ответа. Директива используется, если в ответе содержится личная или банковская информация. Каждый раз, когда пользователь хочет использовать этот ресурс, на сервер отправляется запрос и ответ полностью скачивается заново.

public и private

Если в ответе содержится директива public, его можно кешировать, даже если с ним связана HTTP-аутентификация и код статуса ответа обычно нельзя сохранять. Эта директива требуется редко, потому что другая информация, заданная для кеширования, (например, max-age) показывает, что ответ можно сохранить.

Директива private, наоборот, используется для ответов, которые можно сохранить в кеше браузера, но не в промежуточных кешах. Это происходит из-за того, что подобная информация предназначается для одного пользователя. Например, HTML-страницу с личными данными пользователя можно кешировать в браузере, но не в сетях доставки контента.

max-age

Эта директива указывает период времени в секундах, в течение которого скачанный ответ может быть повторно использован. Отсчет начинается с момента запроса. Например, max-age=60 означает, что ответ может быть кеширован и использован в течение 60 секунд.

Выбор правил Cache-Control



Используйте схему выше, чтобы определить оптимальную политику кеширования для одного или нескольких ресурсов в приложении. Постарайтесь кешировать как можно больше ресурсов на максимально долгий период времени и предоставьте маркеры подтверждения для каждого ответа. Они нужны для проверки актуальность данных.

Кеширование - этот метод, заключающийся в сохранении копии полученного ресурса, чтобы вернуть ее по запросу.