HTTP

Протокол передачи гипертекста (HTTP) — это протокол прикладного уровня для передачи гипермедиа-документов, таких как HTML. Он был разработан для связи между веб-браузерами и веб-серверами, но может использоваться и для других целей. HTTP следует классической модели клиент-сервер , когда клиент открывает соединение для отправки запроса, а затем ждет, пока не получит ответ. HTTP — это протокол без сохранения состояния , что означает, что сервер не сохраняет никаких данных (состояния) между двумя запросами.

Учебники Узнайте, как использовать HTTP, с помощью руководств и руководств.

Обзор HTTP Основные характеристики протокола клиент-сервер: что он может делать и как его использовать.

HTTP-кэш Кэширование очень важно для быстрых сайтов. В этой статье описываются различные методы кэширования и способы управления ими с помощью заголовков HTTP.

HTTP-куки Принцип работы файлов cookie определяется RFC 6265 . При обслуживании HTTP-запроса сервер может отправить Set-CookieHTTP-заголовок с ответом. Затем клиент возвращает значение файла cookie с каждым запросом к тому же серверу в виде Cookieзаголовка запроса. Срок действия файла cookie также может быть установлен на определенную дату или ограничен определенным доменом и путем.

Совместное использование ресурсов между источниками (CORS) Межсайтовые HTTP-запросы — это HTTP-запросы для ресурсов из домена, отличного от домена ресурса, отправляющего запрос. Например, HTML-страница из домена A (http://domaina.example/) делает запрос на изображение в домене B (http://domainb.foo/image.jpg) через imgэлемент. Сегодня веб-страницы очень часто загружают межсайтовые ресурсы, включая таблицы стилей CSS, изображения, сценарии и другие ресурсы. CORS позволяет веб-разработчикам контролировать, как их сайт реагирует на межсайтовые запросы.

Подсказки HTTP-клиента Подсказки клиента — это набор заголовков ответов, которые сервер может использовать для упреждающего запроса информации от клиента об устройстве, сети, пользователе и предпочтениях агента пользователя. Затем сервер может определить, какие ресурсы отправлять, на основе информации, которую клиент выбирает для предоставления.

Эволюция HTTP Краткое описание изменений между ранними версиями HTTP, современным HTTP/2, зарождающимся HTTP/3 и выше.

Руководство по веб-безопасности Mozilla Сборник советов, которые помогут операционным группам создавать безопасные веб-приложения.

HTTP-сообщения Описывает тип и структуру различных видов сообщений HTTP/1.x и HTTP/2.

Типичный сеанс HTTP Показывает и объясняет ход обычного сеанса HTTP.

Управление соединением в HTTP/1.x Описывает три модели управления соединениями, доступные в HTTP/1.x, их сильные и слабые стороны.

Ссылка Просмотрите подробную справочную документацию по HTTP.

HTTP-заголовки Заголовки сообщений HTTP используются для описания ресурса или поведения сервера или клиента. Поля заголовков хранятся в реестре IANA . IANA также ведет реестр предлагаемых новых заголовков HTTP-сообщений .

Методы HTTP-запроса Различные операции, которые можно выполнять с помощью HTTP: GET, POST, а также менее распространенные запросы, такие как OPTIONS, DELETEили TRACE.

Коды ответов состояния HTTP Коды ответа HTTP указывают, был ли успешно выполнен конкретный HTTP-запрос. Ответы сгруппированы в пять классов: информационные ответы, успешные ответы, перенаправления, ошибки клиента и ошибки сервера.

Директивы CSP Поля заголовка ответа Content-Security-Policyпозволяют администраторам веб-сайта контролировать ресурсы, которые пользовательскому агенту разрешено загружать для данной страницы. За некоторыми исключениями, политики в основном включают указание источников серверов и конечных точек сценариев.

From:
https://wwoss.ru/ - **worldwide open-source software**

Permanent link:
https://wwoss.ru/doku.php?id=software:development:web:docs:web:http:http

Last update: **2023/08/21 19:49**