
Разработка многостраничного сайта на PHP.

ЗАНЯТИЕ № 2
HTTP – ЗАГОЛОВКИ ОТВЕТА СЕРВЕРА

Тема занятия: HTTP – заголовки ответа сервера

- Цель занятия –

Продолжить изучать HTTP – заголовки ответа сервера

Глоссарий :

- URI – это универсальный идентификатор ресурса.
- HOST – определяет адрес домена
- Сервер - это специальное оборудование для выполнения сервисного программного обеспечения.
- Заголовок запроса - это метаданная, отправленная браузером при запросе сценария

Актуализация

- 1) Что такое URI?
- 2) Заголовки. Какова их цель?
- 3) Код ответа. Какие бывают виды?
- 4) Как происходит работа с заголовками РНР?
- 5) Каким образом можно получить тело запроса?
- 6) Каким образом можно получить заголовки запроса?

URI

После того, как мы определимся с методом, мы видим указание на адрес страницы – URI.

URI – это универсальный идентификатор ресурса.

Так как пользователь запрашивает главную страницу сайта, поэтому используется слэш - /. В конце строки будет указана версия протокола, почти всегда будет так:



HTTP/1.1.

Заголовки

После указания основных параметров идет перечисление заголовков.

Они передают серверу дополнительную важную информацию, такую как:


- Название и версия браузера;
- Язык;
- Кодировка;
- Параметры кэширования и так далее...

Заголовки

При этом среди всех заголовков есть лишь один самый важный и обязательный заголовок – заголовок HOST.

HOST – определяет адрес домена, который запрашивает браузер клиента.

Сервер после получения запроса начинает искать у себя сайт с доменом из заголовка HOST, а также указанную страницу. В случае, если запрошенный сайт и страница найдены, то клиенту будет отправлен ответ. Ответ означает, что документ найден и будет отправлен.



```
HTTP/1.1 200 OK
```

Общая структура стартовой строки ответа:

HTTP/ версия код состояние
пояснение

Код ответа

В данном случае, конечно, нам очень интересен код состояния, он же код ответа сервера. На предыдущем слайде код ответа – 200.

200 – означает, что сервер работает, документ найден и будет передан клиенту.

Так как иногда сценарий наших действий может идти не совсем так, как мы запланировали, то бывают еще такие варианты:

400 – если сервер доступен, но запрошенный документ не найден;

503– если сервер не может обрабатывать запросы по техническим причинам;

Спецификация HTTP 1.1 определяет 40 различных кодов HTTP. После стартовой строки следуют заголовки, а затем тело ответа.

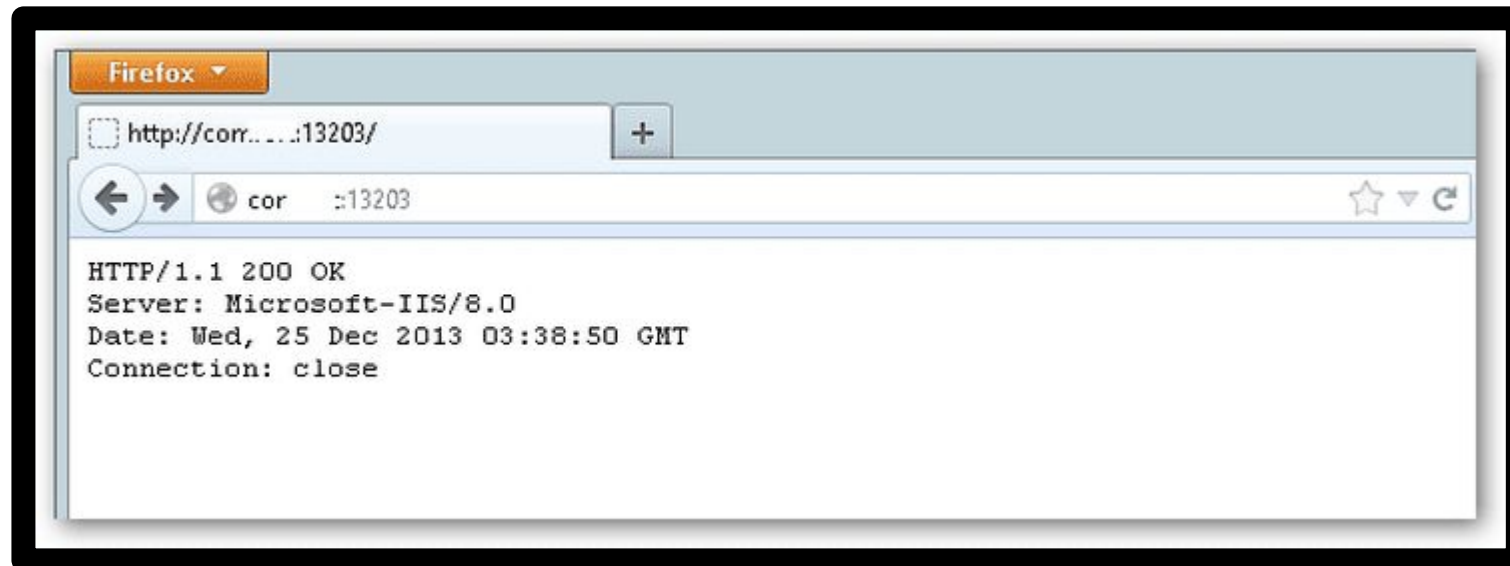
Примеры кода

400 Bad Request

Request Header Or Cookie Too Large

nginx/1.18.0

Примеры кода



Примеры кода

503



503 Service Unavailable

Сервер временно недоступен

Работа с заголовками в PHP

Как PHP взаимодействует с HTTP?

- 1) Получает тело запроса;
- 2) Получает заголовки запроса;
- 3) Добавляет/ изменяет заголовки ответа;
- 4) Управляет телом ответа;

Давайте рассмотрим все по порядку.

Получение тела запроса

Тело запроса — это информация, которую передал браузер при запросе страницы. Но тело запроса присутствует только, если браузер запросил страницу методом POST. Дело в том, что POST — это метод, специально предназначенный для отправки данных на сайт. Чаще всего метод POST браузер задействует в момент отправки формы. В этом случае телом запроса будет содержимое формы.

В PHP-сценарии все данные отправленной формы будут доступны в специальном массиве `$_POST`. Более подробно об этом написано в следующей главе, посвящённой формам.

Получение заголовков запроса

PHP автоматически извлекает такие заголовки и помещает их в специальный массив — `$_SERVER`. Стоит отметить, что в этом массиве, помимо заголовков, есть и другая информация. Значения заголовков запроса находятся под ключами, которые начинаются с `HTTP_`.

Пример, как получить предыдущую страницу, с которой перешёл пользователь:

```
print($_SERVER['HTTP_REFERER']);
```

Рефлексия

- Что мы прошли сегодня на занятии?
- Какие ваши впечатления от новой темы?
- Какие основные моменты вы можете выделить?
- Как вы поняли, что такое URI?
- Какую информацию заголовки передают серверу?
- Что определяет HOST?

A decorative border with floral and scrollwork motifs in the corners and along the sides of the text area.

**Спасибо
за
внимание**