

# PHP: обработка HTTP запросов



---

Краткий обзор

# Передача параметров запроса через суперглобальные переменные

- `$_GET` - массив параметров переданных методом GET
- `$_POST` – массив параметров переданных методом POST
- `$_COOKIE` – массив переданных ключиков
- `$_REQUEST` – объединенный массив переданных параметров
- `$_SERVER` – массив, содержащий информацию, переданную веб-сервером
- `$_FILES` – массив, содержащий сведения о переданных методом POST файлах

# Работа с \$\_FILES

- Суперглобальный массив \$\_FILES содержит всю информацию о загруженных файлах:
  - \$\_FILES['userfile']['name'] - оригинальное имя файла на компьютере клиента
  - \$\_FILES['userfile']['type'] - Mime-тип файла, в случае, если браузер предоставил такую информацию. Пример: "image/gif"
  - \$\_FILES['userfile']['size'] - размер в байтах принятого файла
  - \$\_FILES['userfile']['tmp\_name'] - временное имя, с которым принятый файл был сохранен на сервере
  - \$\_FILES['userfile']['error'] - код ошибки, которая может возникнуть при загрузке файла

# \$\_SERVER

- Переменная `$_SERVER` - это суперглобальный массив, содержащий информацию, такую как заголовки, пути и местоположения скриптов. Записи в этом массиве создаются веб-сервером.
- Индексы:
  - `'SERVER_ADDR'` - IP адрес сервера, на котором выполняется текущий скрипт
  - `'SERVER_NAME'` - имя хоста, на котором выполняется текущий скрипт
  - `'SERVER_SOFTWARE'` - строка идентификации сервера, указанная в заголовках, когда происходит ответ на запрос
  - `'SERVER_PROTOCOL'` - имя и версия информационного протокола, через который была запрошена страница; к примеру `'HTTP/1.0'`;
  - `'REQUEST_METHOD'` - Какой метод был использован для запроса страницы; к примеру `'GET'`, `'HEAD'`, `'POST'`, `'PUT'`.
  -

# \$\_SERVER

- 'QUERY\_STRING' - строка запросов, если есть, с помощью которой была получена страница
- 'DOCUMENT\_ROOT' - директория корня документов, в которой выполняется текущий скрипт
- 'HTTP\_ACCEPT' - содержимое заголовка Accept: из текущего запроса, если он есть
- 'HTTP\_ACCEPT\_CHARSET' Содержимое заголовка Accept-Charset: из текущего запроса, если он есть. Например: 'iso-8859-1,\* ,utf-8'
- 'HTTP\_REFERER' - адрес страницы (если есть), которая привела браузер пользователя на эту страницу. Этот заголовок устанавливается веб-браузером пользователя.
- 'HTTP\_USER\_AGENT' - содержимое заголовка User-Agent: из текущего запроса, если он есть.
-

# Установка ключиков функцией `setcookie()`

- `bool setcookie ( string $name , string $value, int $expire = 0 , string $path, string $domain, bool $secure = false)` - задает cookie, которое будет передано клиенту вместе с другими HTTP заголовками. Все передаваемые параметры соответствуют параметрам HTTP заголовка, устанавливающего ключик.
- Как и любой другой заголовок, cookie должны передаваться до того как будут выведены какие-либо другие данные скрипта (это ограничение протокола). Это значит, что в скрипте вызовы этой функции должны располагаться прежде остального вывода, включая вывод тэгов `<html>` и `<head>`, а также пустые строки и пробелы.
- После передачи клиенту cookie станут доступны через массивы `$_COOKIE` при следующей загрузке страницы. Значения cookie также есть в `$_REQUEST`

# Установка ключиков функцией `setcookie()`

+

- `expire` - время, когда срок действия cookie истекает в количестве секунд с начала эпохи. Можно задавать это время с помощью функции `time()`, прибавляя время в секундах, через которое срок действия cookie должен истечь. Если задать 0 или пропустить этот аргумент, срок действия cookie истечет с окончанием сессии (при закрытии браузера). Преобразование к необходимому формату функция выполняет автоматически.

# Отправка HTTP заголовка с помощью `header()`

- `void header ( string $hstr [, bool $replace = true] )` - используется для отправки HTTP заголовка
- `hstr` – строка заголовка  
`Content-Type: text/plain`
- `replace` - флаг замены. Если установить в `FALSE`, можно задать несколько однотипных заголовков
- в скрипте должна следовать до любого вывода, формирующего тело сообщения!!!
- `header ("location: file.html");`



# Чтение заголовков через `headers_list()`

- `array headers_list ( void )`  
возвращает список заголовков, передаваемых браузеру / клиенту.
- возвращает нумерованный массив заголовков
- для того чтобы определить были ли переданы уже переданы заголовки используется функция `headers_sent()`. Функция обычно используется без параметров и возвращающая значение булева типа

# Загрузка файлов с помощью `move_uploaded_file ()`

- `bool move_uploaded_file ( string $filename , string $destination )` – проверяет, является ли файл `filename` загруженным на сервер (переданным по протоколу HTTP POST) и перемещает его в место, указанное в аргументе `destination`
- `filename` – временный путь к загруженному файлу
- `destination` – конечное назначение перемещаемого файла
- В случае успеха возвращает `TRUE`, иначе `FALSE`