

ОБРАБОТКА ФОРМ



АЛЕКСЕЙ САПРОНОВ / IZet



АЛЕКСЕЙ САПРОНОВ

Начальник отдела программных разработок в IZet



mathehac-as@mail.ru



[mathehac-as](https://t.me/mathehac-as)



vk.com/phpgod



ОСНОВЫ ПРОТОКОЛА HTTP

- HyperText Transfer Protocol
- Является протоколом прикладного 7-ого уровня (по спецификации OSI)
- Режим «запрос-ответ»





OSI



УРОВНИ МОДЕЛИ OSI



```
GET / HTTP/1.1
Host: xbb.uz
User-Agent: Mozilla/5.0 ...
Accept: text/html,application ...
Accept-Language: ru-ru,r ...
Accept-Encoding: gzip,de ...
...
```

Запрос

Ответ

```
HTTP/1.0 200 OK
Server: nginx/0.7.67
Date: Tue, 08 Feb 2011 08: ...
Content-Type: text/html; c ...
X-Powered-By: PHP/5.2.12
Expires: Thu, 19 Nov 1981 ...
...
```

ЗАГОЛОВКИ ЗАПРОСА И
ОТВЕТА

СТАРТОВАЯ СТРОКА ЗАПРОСА HTTP 1.1



<Метод> <URI> HTTP/<Версия>

- Метод – определяет операцию (GET, POST, PUT, DELETE и др.)
- URI (Uniform Resource Identifier) – путь до ресурса
- Версия



СТАРТОВАЯ СТРОКА ОТВЕТА HTTP 1.1

HTTP/<Версия> <Код состояния>
<Пояснение>

- Версия
- Код состояния (Status Code) — три цифры
- Пояснение к коду состояния (Reason Phrase) — текстовое.



ПРОЧИЕ ТОНКОСТИ И РАСШИРЕНИЯ HTTP 1.1

- Конец заголовка – дважды: возврат каретки (\r) + символ перевода строки (\n).
- Расширение HTTPS (Secure)



POST И GET

- GET передает данные серверу используя URL
- POST передает данные, используя тело HTTP запроса
- GET - 1024 символа





ОБРАБОТКА POST И GET

- Суперглобальные переменные `$_GET`, `$_POST`, `$_REQUEST`

```
if (isset($_GET['name'])) {  
    echo 'Hello ' . htmlspecialchars($_GET['name']);  
}
```



ПРОСТАЯ HTML ФОРМА

```
<form action="/" method="post">  
  <input name="fio" type="text" placeholder="Введите ФИО">  
  <input type="submit" value="Отправить">  
</form>
```



ВАЛИДАЦИЯ ДАННЫХ

- Приводим к нужному типу
- Очищаем код от вредоносных вставок
- Делаем проверки на разрешенное содержимое





ПРИВОДИМ К НУЖНОМУ ТИПУ

- (тип)\$variable
- **Типы:** bool или boolean, float, string, array, int или integer

```
$sVariable = '911a';  
$iVariable = (int)$sVariable;  
// $iVariable имеет значение 911
```

ПРИНИМАЕМ, ФИЛЬТРУЕМ И ПРОВЕРЯЕМ



- **filter_input** – принимает переменную извне и при необходимости фильтрует ее

```
$options = [  
    'options' => [  
        'min_range' => 10,  
        'max_range' => 200  
    ]  
];  
$f = FILTER_VALIDATE_INT;  
$r = filter_input(INPUT_GET, 'name',  
$f, $options);
```



ФИЛЬТРУЕМ И ПРОВЕРЯЕМ

- `filter_var` – проверка на соответствующий фильтр и получение значение в случае успеха

```
$options = [  
    'options' => [  
        'min_range' => 10,  
        'max_range' => 200  
    ]  
];  
$f = FILTER_VALIDATE_INT;  
$r= filter_var('123', $f, $options); // 123  
$r= filter_var('1a', $f, $options); //false  
$r= filter_var('2', $f, $options); //false
```



ОЧИЩАЕМ ПРИ ВЫВОДЕ

- **htmlspecialchars—**
Преобразует спец символы в HTML сущности.

```
htmlspecialchars ( '<script></script>' ) ;
```




нетология

ПЕРЕРЫВ

10
МИНУТ



ОБРАБОТКА ФАЙЛОВ НА СЕРВЕРЕ

- `enctype=«multipart/form-data»`
- Глобальный массив `$_FILES`
- `move_uploaded_file($fname, $dest)`



ФОРМА ДЛЯ ОТПРАВКИ ФАЙЛОВ



```
<!-- Тип кодирования данных, enctype, ДОЛЖЕН БЫТЬ указан ИМЕННО так -->
<form enctype="multipart/form-data" action="/2/index.php" method="POST">
  ФИО: <input name="fio" type="text" placeholder="Введите ФИО">
  <br />
  <!-- Название элемента input определяет имя в массиве $_FILES -->
  Аватар: <input name="userfile" type="file" />
  <br />
  <input type="submit" value="Отправить">
</form>
```

ДОМАШНЕЕ ЗАДАНИЕ



- Генератор тестов на PHP и JSON.
- Создаём файл [admin.php](#) с формой через которую на сервер можно загрузить JSON файл с тестом.
- Создаём файл [list.php](#) со списком загруженных тестов.
- Создаём файл [test.php](#), который принимает в качестве GET параметра номер теста, и отображает форму теста. Если форма отправлена, проверяет и показывает результат.





Спасибо за внимание!

АЛЕКСЕЙ САПРОНОВ



matheac-as@mail.ru



[matheac-as](https://t.me/matheac-as)



vk.com/phpgod