

Программно-аппаратные средства WEB

Cookies и сессия

Сергей Геннадьевич Синица

КубГУ, 2020

sin@kubsu.ru

Печеньки

Cookies – расширение протокола HTTP, предназначенное для того, чтобы сохранять на стороне клиента (в браузере) значения некоторых переменных сайта, выдаваемых сервером, и передавать эти значения при каждом последующем HTTP-запросе на этот сайт.

Примеры?

Как выглядит?

Set-Cookie:

SESSac86aef0d3122aba837189cae94ead37=2
fgz5249We0HGd5y3Mwlr75SRUo5fk1DW2Bfab1
LvEM; expires=Fri, 25-Apr-2014
08:27:53 GMT; path=/; HttpOnly

Как выглядит?

```
Cookie: Drupal.toolbar.collapsed=0;  
Drupal.tableDrag.showWeight=0;  
SESSac86aef0d3122aba837189cae94ead37=i  
zgr0qqr8MH10hRgGLp0eXKacqjFaeQkGylSjhk  
DTFk; has_js=1
```

Как выглядит?

Set-Cookie:

```
SESSac86aef0d3122aba837189cae94ead37=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; path=/; httponly
```


Пример на PHP

```
// Записать Cookie.
```

```
setcookie($name, $value, 0, $path,  
$domain, $secure, $httponly);
```

```
// Читать Cookie.
```

```
$var = $_COOKIES['name']
```

Пример на JS

// Можно писать и читать
document.cookies, но проще
использовать плагин JQuery.

```
$.cookie('name', 'value', { expires:  
30, path:  
window.location.pathname });
```

```
var cookie = $.cookie('name');
```

Задача

Реализовать проверку заполнения обязательных полей формы в предыдущей задаче с использованием Cookies, а также заполнение формы по умолчанию ранее введенными значениями.

Сессия

Механизм сессий позволяет сохранять на сервере некоторые переменные (состояние) при работе конкретного пользователя с веб-приложением.

Чем отличается от печенек?

Как сессии связаны с Cookies?

Каковы ограничения cookies и сессий?

Для чего можно использовать сессию/печеньки?

Что с XMLHttpRequest?

Какие угрозы безопасности?

Какие есть альтернативы?

Как работает сессия?

- 1) На сервере проверяется наличие идентификатора сессии в запросе, если есть, то данные по нему загружаются из БД или файла, если нет, то создается пустая сессия.
- 2) Веб-приложение пишет данные в объект сессии. По окончании запроса данные сериализуются и сохраняются на диск, клиент получает идентификатор сессии через Cookie или параметр URL.

Пример на PHP

```
// Начало сессии. Что происходит?  
session_start();
```

```
// Запись в сессию. Как сохраняются данные?  
$_SESSION['key'] = 'value';  
$_SESSION['array'] = array('value1', 'value1');  
...
```

```
// Уничтожение сессии. Что происходит?  
session_destroy();
```

```
// TODO: параметры сессии в настройках сервера, имя  
сессии, пользовательские обработчики сессии.
```

Безопасность сессии

- 1) Привязка к IP-адресу.
- 2) Устаревание сессии.
- 3) Регенерация идентификатора сессии.

Как еще обезопасить сессию?

Как работают системы системы, где важна повышенная безопасность передачи данных?

Какие уязвимости бывают в веб-приложениях и как их избежать?

Задача

Реализовать возможность входа с паролем и логином с использованием сессии для изменения отправленных данных в предыдущей задаче, пароль и логин генерируются автоматически при первоначальной отправке формы.

Литература

1. Веб-программирование и веб-сервисы / С.Г. Сеница. Краснодар: КубГУ, 2013.
2. Плагин JQuery Cookies.
3. Документация PHP.