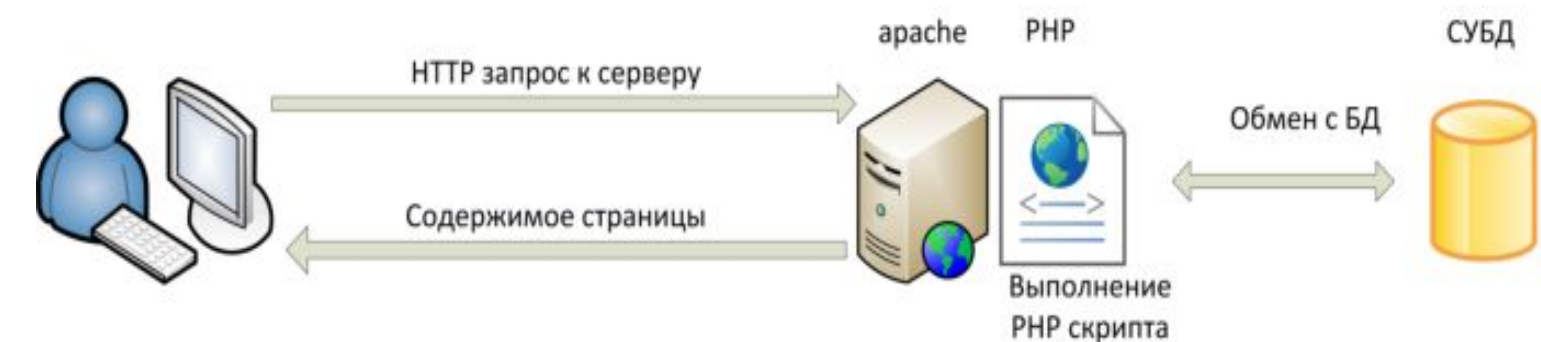


# **Введение в структуру веб-приложений**

# Введение в структуру приложений

Веб-приложение — это клиент-серверное приложение, в котором клиент взаимодействует с сервером при помощи браузера

- **Back-end** (серверная часть) — это часть приложения, которая находится удалённо на любом компьютере где угодно. Серверную часть пишут на высокоуровневых языках программирования, например, PHP, Python, Ruby. Пользователь веб-приложения НЕ видит эту часть приложения
- **Front-end** (клиентская часть) — выполняется в браузере пользователя. Например, страница на сайте. Для написания клиентской части используют JavaScript и языки разметки



# База данных

База данных (БД) — упорядоченный набор структурированной информации или данных

- **Реляционные БД** — наиболее популярный тип базы данных. Информация в такой БД хранится в таблицах, между которыми можно настроить связь по отдельным параметрам (ключам)
- **Иерархическая БД** — хранит информацию в виде древовидной структуры, где есть корневой объект и объекты более низкого уровня (потомки)
- **Сетевая БД** — состоит из наборов записей, связанных между собой так, что записи могут содержать явные ссылки на другие наборы записей

# Популярные базы данных

- MySQL
- PostgreSQL
- Oracle

# Базовые запросы к базе данных

- Создание новой БД

```
CREATE DATABASE db_name
```

- Выбор БД для работы

```
USE db_name
```

- Создание таблицы

```
CREATE TABLE table_name (PRIMARY KEY = (key_name) )
```

# Запросы браузера к серверу

HTTP — протокол передачи данных прикладного уровня

- Запрос GET

Передаёт данные в URL в виде пар «имя-значение». Страница, созданная методом GET, может быть открыта повторно множество раз. Такая страница может быть кэширована браузерами, проиндексирована поисковыми системами и добавлена в закладки пользователем

- Запрос POST

Передаёт данные в теле запроса. Следует использовать в случаях, когда нужно вносить изменение в ресурс (выполнить авторизацию, отправить форму оформления заказа, форму обратной связи, форму онлайн-заявки)

Ситуация	GET	POST
Фильтр товаров	<div>✓</div> <div>Пользователь получит страницу с подходящими ему товарами, сможет сохранить ее в закладки, переслать ссылку на страницу с параметрами другим и вернуться к странице позже без повторного заполнения формы фильтра.</div>	<div>✗</div> <div>Для повторного посещения страницы пользователь должен будет повторно заполнить форму фильтра, страницей с параметрами нельзя будет поделиться, сохранить в закладки и вернуться к странице позже без повторного заполнения формы фильтра.</div>
Форма авторизации	<div>✗</div> <div>Отсутствует защита конфиденциальной информации. Введенный пароль будет виден в адресной строке браузера, будет сохранен в истории посещенных сайтов.</div>	<div>✓</div> <div>Хотя данные могут передаваться в незашифрованном виде, увидеть их можно только через инструменты разработчика или с помощью другого специального программного обеспечения.</div>
Онлайн заявка Оформления заказа Форма обратной связи	<div>✗</div> <div>При повторном обращении по конечной ссылке на сервере будет произведена повторная обработка, например, будет создана повторная заявка, оформлен еще один заказ, создан еще один запрос на обратную связь.</div>	<div>✓</div> <div>Повторное обращение по конечной ссылке не приведет к повторной обработке запроса с введенными ранее параметрами.</div>
Через гиперссылку	<div>✓</div> <div>Переход по гиперссылке с параметрами равнозначен отправке запроса через HTML форму.</div>	<div>✗</div> <div>Нет технической возможности поместить POST запрос в гиперссылку.</div>
AJAX запросы	Используются оба метода. Выбор зависит от контекста. Принципы выбора метода такие же, как и для HTML форм.	

# Запросы браузера к серверу

Свойство	GET	POST
Способ передачи данных	Через URL	В теле HTTP запроса
Защита данных	Данные видны всем в адресной строке браузера, истории браузера и т.п.	Данные можно увидеть только с помощью инструментов разработчика, расширений браузера, специализированных программ.
Длина запроса	Не более 2048 символов	Не ограничена <i>Примечание: ограничения могут быть установлены сервером.</i>
Сохранение в закладки	Страница с параметрами может быть добавлена в закладки	Страница с параметрами не может быть добавлена в закладки.
Кэширование	Страница с параметрами может быть кэширована	Страница с параметрами не может быть кэширована
Индексирование поисковыми системами	Страница с параметрами может быть индексируется	Страница с параметрами не может быть индексируется
Возможность отправки файлов	Не поддерживается	Поддерживается
Поддерживаемые типы кодирования	application/x-www-form-urlencoded	application/x-www-form-urlencoded multipart/form-data text/plain
Использование в гиперссылках <a>	Да	Нет
Использование в HTML формах	Да	Да
Использование в AJAX запросах	Да	Да

# HTML-код, генерирующий GET запрос

```
<form method="GET" name="filter" action="http://example.com/catalog/">
  <p>Диагональ экрана</p>
  <label><input type="radio" name="screen" value="4" checked> 4</label><br>
  <label><input type="radio" name="screen" value="4.5"> 4.5</label><br>
  <label><input type="radio" name="screen" value="5"> 5</label>
  <p>Цвет</p>
  <label><input type="checkbox" name="color_black" checked> Черный</label><br>
  <label><input type="checkbox" name="color_white" checked> Белый</label><br>
  <label><input type="checkbox" name="color_golden"> Золотой</label><br>
  <input type="submit" value="Применить фильтр">
</form>
```



# HTML-код, генерирующий POST запрос

```
<form method="POST" name="authorization" action="http://example.com/profile.php">  
  Логин: <input type="text" name="username"><br>  
  Пароль: <input type="password" name="user_password"><br>  
  <input type="submit" value="Войти">  
</form>
```

Заголовки запроса:

```
POST /profile.php HTTP/1.1  
Host: example.com  
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8  
Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.5,en;q=0.3  
Accept-Encoding: gzip, deflate  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 35  
Origin: http://example.com  
Connection: keep-alive  
Referer: http://example.com/profile.php  
Upgrade-Insecure-Requests: 1  
Pragma: no-cache  
Cache-Control: no-cache
```

Тело запроса:

```
username=admin&user_password=123456
```