



# Веб-приложения

# Веб-приложение

- Клиент-серверное приложение, предоставляющее веб-интерфейс
  - Клиент — браузер
    - MS IE, Mozilla Firefox, Opera, Google Chrome, ...
  - Сервер — веб-сервер
    - обеспечивает обработку протокола передачи гипертекста (Hypertext Transfer Protocol – HTTP)
    - Apache, MS IIS, ...
  - Веб-интерфейс
    - статические HTML-страницы
    - *динамические* HTML-страницы — создаются или изменяются в ответ на действия пользователя

# Выполнение веб-приложений

- **Common Gateway Interface (CGI)**
  - Стандарт связи внешней прикладной программы (CGI-шлюза) с веб-сервером
  - Передача параметров — через командную строку и переменные окружения
  - Вывод результата — в стандартный поток вывода
  - Для обработки каждого запроса создается отдельный процесс ОС
  - Для разработки используются C/C++, Fortran и т.п., а также скриптовые языки Perl, TCL, Unix Shell и пр.
  - Шлюзы размещаются в подкаталоге cgi-bin
  - Недостатки: невысокая производительность и плохая масштабируемость

# Выполнение веб-приложений

- **Сервер веб-приложений**
  - Веб-приложения разрабатываются в виде подключаемых модулей и взаимодействуют с сервером через API
    - MS IIS — веб-приложения реализуются в виде DLL и взаимодействуют с сервером через интерфейс ISAPI
    - Java EE — веб-приложения распространяются в виде WAR-архивов и взаимодействуют с веб-контейнером через Servlet API
  - Сервер предоставляет дополнительные функции для веб-приложений (управление ЖЦ, обеспечение безопасности, организация пулов ресурсов)



# Взаимодействие по протоколу HTTP

# Протокол HTTP

- Протокол прикладного уровня для распределенных гипермедийных информационных систем
  - Используется в WWW, начиная с 1990 года
  - HTTP/0.9 → HTTP/1.0 → **HTTP/1.1** (RFC 2616)
- Протокол типа запрос/ответ
  - Запрос операции с ресурсом (URI) и результат ее выполнения
- Протокол без состояния

# Обобщенный формат сообщения RFC 822

- Запросы/ответы протокола HTTP, сообщения протокола SMTP

```
generic-message = start-line
                  *message-header
                  CRLF
                  [ message-body ]

message-body = entity-body
             | <entity-body, закодированное согласно Transfer-Encoding>

entity-body := Content-Encoding( Content-Type( data ) )
```

  - Тип содержимого (Content-Type) определяет MIME-тип объекта
  - Кодирование содержимого (Content-Encoding) - позволяет сжать данные (GZIP)

# MIME-тип

| | |
|---|---|
| `text/plain` | Простой текст |
| `text/html` | HTML-документ |
| `text/xml` | XML-документ |
| `image/gif`, `image/jpeg`, `image/png` | Изображения в форматах GIF, JPEG, PNG |
| `video/mpeg` | Видео в формате MPEG-x |
| `application/pdf` | PDF-документ |
| `application/x-msword` | Документ MS Word |

# HTTP-запрос

- Первая строка содержит:
  - метод, который нужно применить к ресурсу
  - универсальный идентификатор ресурса (URI)
  - используемую версию протокола

```
GET /servlets-examples/servlet/RequestParamExample HTTP/1.1
Accept: image/gif, image/jpeg, application/msword, */*
Referer: http://localhost:8084/servlets-examples/
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.2)
Host: localhost:8084
Connection: Keep-Alive
```

| ← пустая строка

# HTTP-запрос: URI

- Имена (URN) и локаторы (URL)
- Структура URL:

```
schema://[user:password@]host[:port]/[path][?query][#hash]
```

  - `schema` - схема; определяет правила кодирования URL для конкретного прикладного протокола (для HTTP — `http` и `https`)
  - `user`,`password` - имя пользователя и пароль для доступа к ресурсу
  - `host` - доменное имя или IP-адрес сервера
  - `port` - порт сервера (для HTTP по умолчанию 80)
  - `path` - логический путь и имя ресурса на сервере
  - `query` - параметры запроса
  - `hash` - идентификатор местоположения в документе-ответе

# HTTP-запрос: метод

| Метод | Описание |
|---|---|
| OPTIONS | Запрос информации о коммуникационных возможностях сервера. |
| GET | Получение информации с сервера. |
| HEAD | Получение заголовков. Идентичен методу GET (все заголовки должны быть установлены), но ответ не должен содержать тела сообщения. |
| POST | Обработка содержащегося в запросе объекта как подчиненного относительно запрашиваемого ресурса. |
| PUT | Сохранение содержащегося в запросе объекта по указанному адресу. |
| DELETE | Удаление указанного в запросе ресурса. |
| TRACE | Трассировка запроса – возвращение текста запроса в теле ответа. |

# HTTP-ответ

- Первая строка ответа – это строка статуса
  - Версия протокола
  - Код статуса
  - Поясняющая фраза

```
HTTP/1.1 200 OK
Date: Tue, 01 Dec 2001 23:59:59 GMT
Content-Type: text/html
Content-Length: 51

<html>
<body>
<h1>Hello, John!</h1>
</body>
</html>
```

← пустая строка

# HTTP-ответ: код статуса

- 1xx: Информационные коды
- 2xx: Успешные коды
- 3xx: Коды перенаправления
- 4xx: Коды ошибок клиента
- 5xx: Коды ошибок сервера

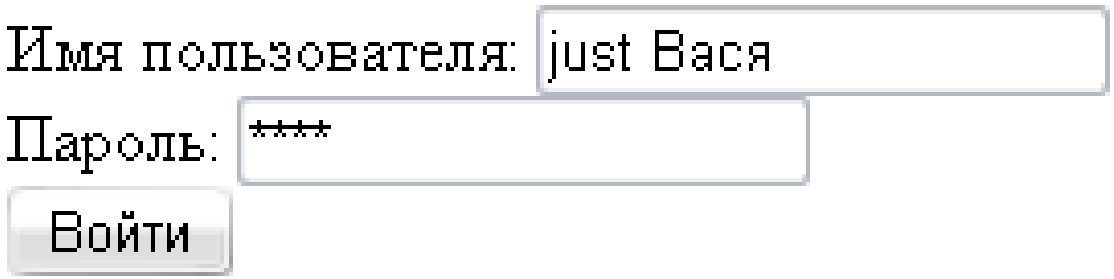
- 200 OK
- 301 Moved Permanently
- 302 Moved Temporarily
- 403 Forbidden
- 404 Not Found
- 500 Internal Server Error



# Организация диалога с пользователем

# HTML-формы: пример

```
<form name="login" action="controller" method="get">
Имя пользователя: <input name="username" type="text"></br>
Пароль: <input name="password" type="password"></br>
<input name="action" type="hidden" value="login">
<input type="submit" value="Войти">
</form>
```



# HTML-формы: элемент <form>

- name — имя формы
- action — URL программы-обработчика
- method — метод передачи данных формы
  - get — внутри URL (по умолчанию)
    - http://www.vstore.com/controller?username=just+%C2%E0%F1%FF&password=1234&action=login
  - post — в теле HTTP-сообщения

```
POST /controller HTTP/1.1
Host: www.vstore.com
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0)
Content-Type: application/x-www-form-ulrencoded
Content-Length: 40

username=just Вася
password=1234
action=login
```

← пустая строка

- enctype — формат кодирования данных для метода POST
  - application/x-www-form-ulrencoded (по умолчанию)
  - multipart/form-data (для передачи на сервер файлов)

# HTML-формы: элементы управления

- <input> — семейство элементов управления, конкретный тип указывается атрибутом type:
  - checkbox – флажок
  - button – обычная кнопка
  - file – элемент выбора локального файла
  - hidden – скрытый элемент формы
  - password – строка редактирования для ввода пароля
  - radio – радио-кнопка
  - reset — кнопка восстановления исходных значений
  - submit — кнопка передачи данных формы на сервер
  - text — текстовое поле (по умолчанию)
- <select> — список выбора (значения — элементы <option>)
- <textarea> — многострочное текстовое поле