



УПРАВЛЕНИЕ ДАННЫМИ 2024



ЛЕКЦИЯ 7

Поддержка разработки веб приложений баз данных.

ОПРЕДЕЛЕНИЕ URI

- ◆ Uniform Resource Identifiers – специальная схема, которая идентифицирует ресурсы в сети Интернет. В качестве ресурсов выступают страницы Интернет, музыка, видео, картинки и т.д.
- ◆ URL — это подвид URI, который, помимо идентификации ресурса, предоставляет ещё и информацию о местонахождении этого ресурса.
- ◆ URI могут быть реализованы в разном виде:
 - ◇ <https://msuniversity.ru>
 - ◇ <mailto:smirnovmgupi@gmail.com>
 - ◇ <file:///Users/John/Documents/Projects/Web/MyWebsite/about.html>
 - ◇ <tel:+1-816-555-1212>

HYPertext TRAnSFER PROTOCOl

- ◆ Это набор стандартов, которые определяют структуру передаваемых по сети сообщений. За правильную интерпретацию и передачу сообщений обычно отвечает браузер. Реже – пользовательское приложение.
- ◆ Типовая схема взаимодействия с сервером:
 - ◇ клиент (веб-браузер) отправит HTTP запрос серверу
 - ◇ сервер получит запрос и ответит на него
 - ◇ клиент получит ответ, создаст новые запросы

HYPERTEXT TRANSFER PROTOCOL. HTTP ЗАПРОС.

- ◆ Строка запроса GET ~/index.html HTTP/1.1
 - ◇ GET: поле HTTP метода (GET, POST и т.д.)
 - ◇ ~/index.html: поле URI
 - ◇ HTTP/1.1 – поле версии HTTP
- ◆ Тип клиента: User-agent: Opera/...
- ◆ Тип принимаемых клиентов файлов: Accept: text/html, image/gif, image/jpg...

HYPERTEXT TRANSFER PROTOCOL. HTTP ЗАПРОС, ПРИМЕР.

POST / HTTP/1.1

Host: example.com

User-Agent: Mozilla/5.0 (X11;...) Firefox/91.0

Accept: text/html, application/json

Accept-Language: ru-RU

Accept-Encoding: gzip, deflate

Connection: keep-alive

Upgrade-Insecure-Requests: 1

Content-Type: multipart/form-data; boundary=b4e4fbd93540

Content-Length: 345

Заголовки
запроса

Заголовки общего
назначения

Заголовки
представления

HYPERTEXT TRANSFER PROTOCOL. HTTP ОТВЕТ.

- ◆ Строка статуса: HTTP/1.1 200 OK
 - ◇ Версия HTTP: HTTP/1.1
 - ◇ Код статуса: 200
 - ◇ HTTP/1.1 – поле версии HTTP
 - ◆ 200 OK запрос успешен
 - ◆ 400 Bad Request запрос с ошибками
 - ◆ 404 Not found запрошенный объект отсутствует на сервере
 - ◆ 500 Internal error сервер не может обработать запрос
- ◆ Дата создания объекта: Last-Modified: Mon, 04 May 2020...
- ◆ Количество отправляемых байтов: Content-Length: 1024
- ◆ Тип отправляемого объекта: Content-Type: image/jpg

HYPERTEXT TRANSFER PROTOCOL. HTTP ОТВЕТ, ПРИМЕР.

```
HTTP/1.1 200 OK
Server: nginx/1.2.1
Date: Sat, 08 Mar 2014
    22:53:46 GMT
Content-Type: application/json
Content-Length: 500
Last-Modified: Sat, 08 Mar
    2014 22:53:30 GMT
Connection: keep-alive
```

```
{
  "status": 200,
  "message": "Success",
  "data" : {
    "name": "John Doe",
    "age": 25,
    "email":
      "john.doe@example.com"
  }
}
```

ОСОБЕННОСТИ РАБОТЫ HTTP.

- ◆ HTTP не умеет отслеживать состояние рабочей сессии клиента:
 - ◆ нет сессионности (начало-конец)
 - ◆ протокол не запоминает предыдущих взаимодействий и их результатов
 - ◆ протокол не имеет возможности вести конфигурацию (настройки)
- ◆ Самый популярный инструмент помощи HTTP – это печенье ^_^ (cookies)

COOKIE В ЗАПРОСАХ HTTP.

```
HTTP/1.0 200 OK
Content-type: text/html
Set-Cookie: yummy_cookie=choco
Set-Cookie:
tasty_cookie=strawberry
```

```
[page content]
```

```
GET /sample_page.html HTTP/1.1
Host: www.example.org
Cookie: yummy_cookie=choco;
tasty_cookie=strawberry
```

ИНСТРУМЕНТАРИЙ ПРЕДСТАВЛЕНИЯ ДАННЫХ В ВЕБ ПРИЛОЖЕНИИ.

- ◆ HTML (HyperText Markup Language) – стандартизованный язык разметки документов
- ◆ XML (eXtensible Markup Language) – расширяемый язык разметки документов
- ◆ JSON (JavaScript Object Notation) - стандартный текстовый формат для хранения и передачи структурированных данных
- ◆ YAML (Yet Another Markup Language, но это неточно ;))

КРАТКОЕ ОПИСАНИЕ HTML.

- ◆ В качестве команд используются тэги.
 - ◇ Начальный тэг и конечный тэг
 - ◇ Например:
 - ◆ `<HTML>...</HTML>`
 - ◆ `...`
- ◆ Чаще всего разметка создается в автоматическом режиме с помощью прикладных программ (например, MS Word)

КРАТКОЕ ОПИСАНИЕ HTML. ПРИМЕР ОФОРМЛЕНИЯ ТАБЛИЦЫ.

```
<table>
  <tr>
    <td>&nbsp;</td>
    <td>Knocky</td>
    <td>Flor</td>
  </tr>
  <tr>
    <td>Breed</td>
    <td>Jack Russell</td>
    <td>Poodle</td>
  </tr>
  <tr>
    <td>Age</td>
    <td>16</td>
    <td>9</td>
  </tr>
</table>
```

КРАТКОЕ ОПИСАНИЕ XML.

- ◆ С помощью инструментария XML можно передать любой массив данных вместе с его описанием.
- ◆ Например: «химический» язык разметки:

```
<molecule>  
  <weight>234.5</weight>  
  <figures>...</figures>  
</molecule>
```

КРАТКОЕ ОПИСАНИЕ JSON.

- ◆ Очень похож на XML, но значительно компактнее и в случае сложных структур документов будет значительно эффективнее.
- ◆ Применяется в веб-приложениях как для обмена данными между браузером и сервером (AJAX), так и между серверами.

КРАТКОЕ ОПИСАНИЕ JSON. ПРИМЕР ДОКУМЕНТА.

```
{
  "added_at": "2015-01-25T07:51:45Z",
  "added_by": {
    "external_urls": {
      "spotify":
        "http://open.spotify.com/user/exampleuser"
    },
    "href": "https://api.spotify.com/v1/users/exampleuser",
    "id": "exampleuser",
    "type": "user",
    "uri": "spotify:user:exampleuser"
  },
  "is_local": true,
  "track": {
    [Spotify Track Object]
  }
}
```

ИНСТРУМЕНТЫ ВИЗУАЛИЗАЦИИ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА: HTML ФОРМА.

- ◆ Один из элементов HTML документа (доступных тэгов языка разметки).
- ◆ Ключевые компоненты формы:
 - ◇ ACTION – указание на URI сервера данных или приложений
 - ◇ METHOD – выбор HTTP GET или POST
 - ◇ NAME – название формы, нужно для скриптов на стороне клиента



ВОПРОС.

А как получить в мою HTML форму данные из реляционной или постреляционной БД?

SOAP ИЛИ REST API В JS.

- ◆ Обычно встраивается прямо в HTML документ с тэгами `<SCRIPT>...</SCRIPT>`.
- ◆ Основная цель – добавить функциональности статическим html документам и формам, в том числе и в деле передачи и размещения данных XML или JSON документов в формах.

ПРИМЕР REST API В СКРИПТЕ JS. ПОДКЛЮЧЕНИЕ К БД.

```
var sqlConfig = {  
  user: 'Shmuser',  
  password: 'Korol'',  
  server: 'localhost',  
  database: 'DBName'  
}
```

ПРИМЕР REST API В СКРИПТЕ JS. ЗАПРОС К БД ТИПА GET.

```
app.get('/sales', function (req, res) {  
  sql.connect(sqlConfig, function() {  
    var request = new sql.Request();  
    request.query('select * from sales', function(err,  
resp) {  
      if(err) console.log(err);  
      res.json(resp.recordset);  
      sql.close();  
    });  
  });  
});
```



ВОПРОС.

Как мне упорядочить и стандартизовать большое количество веб форм и документов моего приложения?

ТАБЛИЦЫ СТИЛЕЙ ВЕБ ПРИЛОЖЕНИЯ.

- ◆ Идея: разное воспроизведение контента, в зависимости от его содержимого и адаптация выдачи информации под разные форматы.
- ◆ Языки таблиц стилей:
 - ◆ Cascading style sheets (CSS) – в первую очередь для HTML документов
 - ◆ Extensible stylesheet language (XSL) – для документов XML

CASCADE STYLESHEETS.

- ◆ Определяет, каким образом выводить на экран документы HTML.
- ◆ CSS может управлять множеством HTML документов.
- ◆ Каждая строка CSS состоит из трех частей: selector {property: value}
 - ◇ selector: тэг с определенным форматом
 - ◇ property: атрибут тэга с устанавливаемым значением
 - ◇ value: значение атрибута

CASCADE STYLESHEETS. ПРИМЕР.

```
body {  
    width: 600px;  
    margin: 0 auto;  
    background-color: #ff9500;  
    padding: 0 20px 20px 20px;  
    border: 5px solid black;  
}
```

САМОСТОЯТЕЛЬНОЕ ЗАДАНИЕ

- ◆ CS50 2020 - Lecture 8 - HTML, CSS, JavaScript (pre-release):

<https://www.youtube.com/watch?v=xvkHFSNfzyQ>

СПАСИБО!

ВАШИ ВОПРОСЫ,
ПОЖАЛУЙСТА?

