



Работа с http запросами



# Node.js

## Автор курса



Владимир Виноградов



MCID: 9210561

# Node.js

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](http://itvdn.com)

Доступ можно получить через руководство  
вашего учебного центра



Проверьте как Вы усвоили данный материал  
на [TestProvider.com](http://testprovider.com)

# Node.js

Тема

## Работа с http запросами

# Http

## Http протокол

**HTTP** – это протокол прикладного уровня для передачи данных от браузера к серверу и обратно. HTTP сообщение обычно передаются между сервером и браузером через порт 80 или 443 при использовании Secure HTTP (HTTPS).

# Http

## Request

При запросе страницы браузер отправляет текстовую команду на сервер.

```
GET/default.aspxHTTP/1.1  
Host:www.example.com
```

**GET** – HTTP глагол (метод или команда) описывающая действие, которое должен выполнить вебсервер.

**/default.aspx** – запрашиваемая на сервере страница.

**HTTP/1.1** – версия протокола.

**Host:www.example.com** – заголовок. Доменное имя сайта, к которому выполняется запрос. Полезно в том случае, если на сервере одновременно работает несколько веб приложений.

# Http

## Response

HTTP/1.1 200 OK

Content-Type: text/html

Content-Length: 36

<html> <body>Hello world</body> </html>

HTTP/1.1 – версия протокола

200 – status code

OK – описание статуса

**Content-Type: text/html** – заголовок с MIME типом ответа. Данное значение нужно для того, чтобы браузер правильно интерпретировал данные полученные от сервера.

**Content-Length: 36** – размер тела ответа в байтах.

**<html> <body>Helloworld</body> </html>** – тело ответа.

# Http

## Http глаголы

HTTP глагол	Описание
OPTIONS	Используется клиентским приложением для получения списка доступных глаголов
GET	Получение данных с сервера.
HEAD	Получение метаданных (заголовков) ресурса. При данном запросе ресурс не возвращается.
POST	Отправка данных на сервер для обработки. Обычно данные введенные пользователем в форму на странице.
PUT	Позволяет клиенту создать ресурс по указанному URL (создать файл на сервере).
DELETE	Удаление ресурса на сервере.
CONNECT	Команда для использования прокси серверами.



# Http

## Группы статус кодов

Группа	Описание
1xx	Информационные
2xx	Успешное завершение
3xx	Команды перенаправлений
4xx	Клиентские ошибки
5xx	Серверные ошибки

# Module http

## Модуль http

Для использования клиента и сервера HTTP необходимо подключить соответствующий модуль с помощью `require('http')`.

Для создания сервера используется метод `http.createServer([requestListener])`. Метод возвращает новый экземпляр объекта `http.Server`.

Метод `server.listen(port, [hostname], [callback])` начинает приём соединений на указанном порту и имени хоста. Если имя хоста не указано, сервер будет принимать соединения на любой IPv4-адрес машины.

```
var http = require('http');
var server = http.createServer(function(req, res) {
  res.end('Hello World!');
}).listen(3000);
```

# Query processing

## Обработка запросов

`http.Server` - это `EventEmitter` со следующими событиями: `request`, `connection`, `close` etc.

Функция, обрабатывающая запрос - обработчик события `request`:



Событие `request` генерируется каждый раз при получении запроса. В течении одного соединения может происходить несколько запросов (в случае `keep-alive` соединения).

Объект `request` — экземпляр `http.ServerRequest`, объект `response` - экземпляр `http.ServerResponse`.

# Query

## Создание запросов

Для создания HTTP запросов используется метод `http.request(options, callback)`

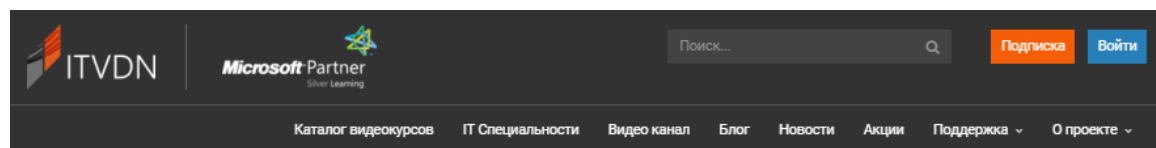
`http.request()` возвращает объект класса `http.ClientRequest`, являющийся потоком с возможностью записи.

Используя `http.request()` вы должны вызвать `req.end()`, иначе всё что вы делали до этого с объектом запроса не произойдёт и данные не будут отправлены

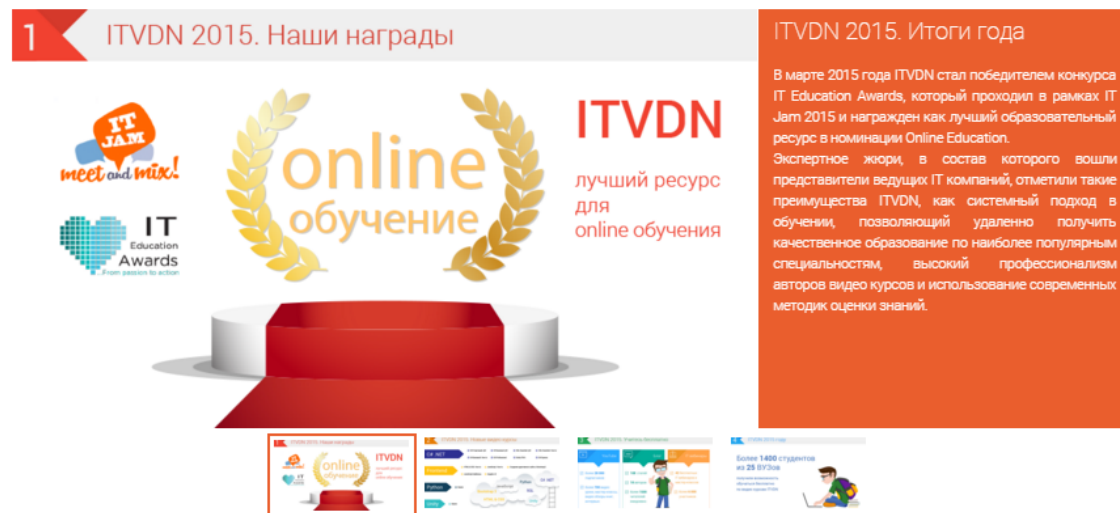
.

# Смотрите наши уроки в видео формате

ITVDN.com



Посмотрите этот урок в видео формате на образовательном портале [ITVDN.com](http://itvdn.com) для закрепления пройденного материала.



Все курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics

## Новые видео

Исключения	0
Итераторы и генераторы	0

## Популярные видео курсы

Видео курс C# Стартовый (для начинающих)	9 уроков (16 ч. 3 мин.)
Видео курс по шаблонам проектирования	29 уроков (16 ч. 7 мин.)

## Теги

.NET Developer
Frontend Developer



# Проверка знаний

## TestProvider.com

TestProvider

Мы помогаем людям оценить себя

Регистрация Войти

Главная Каталог Сертификация Microsoft Поддержка О нас

### Тестирование

Языки программирования и информационные технологии

**Microsoft**

C# ASP.NET MVC JavaScript Patterns Of Design SQL Architecture Guide WCF HTML&CSS XML SEO WPF HTML5&CSS3 JQuery XNA SharePoint GUI for Android Windows Azure Platform Microsoft Patterns&Practices TFS SCRUM ReSharper TDD WWF LINQ Entity Framework Windows Forms Refactoring Microsoft Expression Blend 4 Windows Phone 8 Windows 8 AppStore Visual Studio Tips&Tricks MSF MEF SilverLight AJAX MEF Service Oriented Architecture

Пройти тест

Наши партнеры

Microsoft Partner CyberBionic ITVDN PROMETRIC TEST CENTER PEARSON VUE Authorized Test Center Windows Azure Cloud Partner EBA

Дополнительные ресурсы:

Очное обучение On-line обучение Видео обучение

TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на [TestProvider.com](http://TestProvider.com)

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



## Q&A

# Информационный видеосервис для разработчиков программного обеспечения

