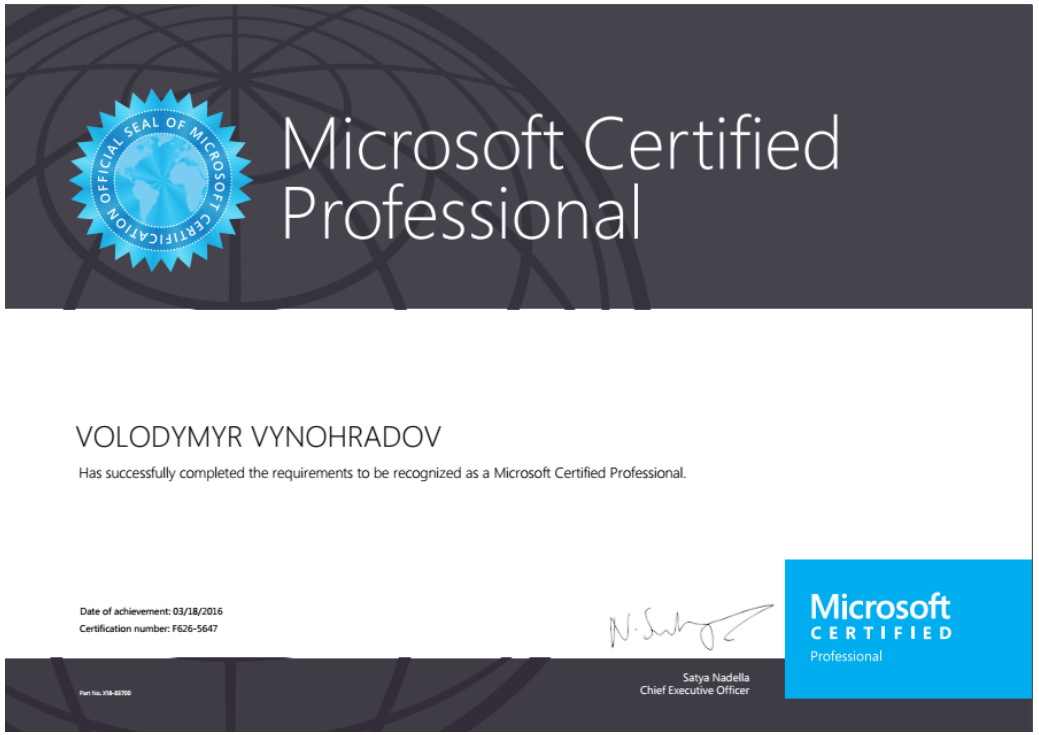
Фреймворк Express

Автор курса

Владимир Виноградов



MCID: 9210561

После урока обязательно

Повторите этот урок в видео формате на ITVDN.com

Доступ можно получить через руководство вашего учебного центра

Проверьте как Вы усвоили данный материал на TestProvider.com

Тема

Фреймворк Express

Введение в Express

Express - это минималистичный и гибкий веб-фреймворк для приложений Node.js, построенный на базе фреймворка connect.

Основное предназначение Express - маршрутизация и промежуточная обработка с минимальной собственной функциональностью: приложение Express, по сути, представляет собой серию вызовов функций промежуточной обработки (middleware).

Для использования фреймворка, необходимо подключить модуль express, и создать приложение.

```
var express = require('express');
var app = express();
```

Маршрутизация

Маршрутизация определяет, как приложение отвечает на клиентский запрос к конкретному адресу, и определенному методу HTTP запроса.

Каждый маршрут может иметь несколько обработчиков, которые выполняются при сопоставлении маршрута.

Для определения маршрута используют следующую структуру:

app.METHOD(path, handler)

app – экземпляр express приложения
METHOD – метод HTTP запроса
path – путь на сервере
Handler – функция обработчик на указанный путь

```
app.get('/', function (req, res) {
  res.send('Hello World!');
});
```

```
app.post('/', function (req, res) {
  res.send('Got a POST request');
});
```

```
app.put('/user', function (req, res) {
  res.send('Got a PUT request at /user');
});
```

```
app.delete('/user', function (req, res) {
  res.send('Got a DELETE request at /user');
});
```

Routing

Используя класс express.Rouer можно создавать модульные, монтируемые обработчики маршрутов. Экземпляр этого класса представляет комплексную систему промежуточных обработчиков и маршрутизации.

```
var birds = require('./birds');
...
app.use('/birds', birds);
```

```
var express = require('express');
var router = express.Router();

// middleware that is specific to this router
router.use(function timeLog(req, res, next) {
  console.log('Time: ', Date.now());
  next();
});
// define the home page route
router.get('/', function(req, res) {
  res.send('Birds home page');
});
// define the about route
router.get('/about', function(req, res) {
  res.send('About birds');
});

module.exports = router;
```

Методы ответа

Методы в объекте ответа (response), могут передавать клиенту ответ и завершать цикл «запрос-ответ». Если ни один из методов не вызвать, клиентский запрос зависнет.

Метод	Описание
response.download()	Запрос на загрузку файла
response.end()	Завершить процесс ответа
response.json()	Отправки ответа в вид JSON
response.redirect()	Перенаправление ответа
response.render()	Вывод шаблона представления
response.send()	Отправка ответа различных типов
response.sendFile()	Отправка файла
response.sendStatus()	Установка кода состояния ответа и отправка представления в виде строки в качестве тела ответа.

Middleware-функция

Функции промежуточной обработки (middleware) - это функции, имеющие доступ к объекту запроса (**req**), объекту ответа (**res**) и к следующей функции промежуточной обработки в цикле "запрос-ответ" приложения (**next**).

Функции промежуточной обработки могут выполнять следующие задачи:

- Выполнение любого кода.
- Внесение изменений в объекты запросов и ответов.
- Завершение цикла "запрос-ответ".
- Вызов следующего промежуточного обработчика из стека.

```
app.use(function(req, res, next) {
        res.send('OK');
});
```

Body-parser

Объект body-parser предоставляет различные функции-фабрики для создания промежуточных обработчиков. Все посредники заполняют свойство req.body обработанной информацией, если заголовок запроса Content-Type соответствует типу.

Модуль обеспечивает следующие парсеры:

- JSON body parser
- Raw body parser
- Text body parser
- URL-encoded from body parser

Смотрите наши уроки в видео формате

ITVDN.com

Посмотрите этот урок в видео формате на образовательном портале ITVDN.com для закрепления пройденного материала.

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

ITVDN
IT VIDEO DEVELOPERS NETWORK

Проверка знаний

TestProvider.com

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

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.

TestProvider

Q&A

Информационный видеосервис для разработчиков программного обеспечения

ITVDN

IT VIDEO DEVELOPERS NETWORK

TestProvider PROMETRIC TEST CENTER CyberBionic systematics Microsoft Partner Silver Learning Windows Azure Cloud Partner EBA European Business Association