

Framework Express

№ урока: 5 Курс: Node.js

Средства обучения: Персональный компьютер с установленной Visual Studio Code, и Node.js.

Обзор, цель и назначение урока

На этом уроке слушатель узнает, что такое Фреймворк express.js и для чего он нужен, научится устанавливать данный Фреймворк и создавать приложения. На уроке также будет рассмотрено работа с объектами запроса и ответа в express, работа с маршрутами, конфигурация маршрутов с помощью класса Route. На уроке будет рассмотрено написание cookie пасера и логера. Будет рассмотрена работа со статическими файлам и обработка ошибок.

Изучив материал данного занятия, учащийся сможет:

- Установить платформу Node.js
- Писать скрипты, которые будут выполняться на платформе Node.js
- Понимать, что такое модуль
- Уметь создавать модули и подключать их к другим модулям

Содержание урока

1. Знакомство с фреймворком express.
2. Установка фреймворка express.
3. Создание первого приложения.
4. Рассмотрение работы конвейера обработки запросов.
5. Маршрутизация в express.
6. Обработка ошибок.

Резюме

- **Express** – это минималистичный и гибкий фреймворк для создания приложений Node.js, построенный на базе фреймворка *connect*.
- Основное предназначение фреймворка – маршрутизация и промежуточная обработка минимальной собственной функциональностью.
- Приложение Express представляет собой серию вызовов функций промежуточной обработки, который в терминологии express называются *middleware*.
- Функции промежуточной обработки (*middleware*) – это функции, имеющие доступ к объекту запроса, объекту ответа и следующей функции промежуточной обработки в цикле «запрос-ответ».
- Для установки фреймворка необходимо его установить командой: ***npm install express***.
- Для создания приложения, необходимо подключить установленный модуль и вызвать функцию, экспортируемую из подключаемого модуля.
- Маршрутизация определяет, как приложение отвечает на клиентский запрос к конкретному адресу, и определенному методу http запроса.
- Модуль express содержит конструктор Router, который позволяет создавать модульные, монтируемые обработчики маршрутов. Экземпляр этого класса представляет комплексную систему промежуточных обработчиков и маршрутизации.
- Модуль **body-parser** – предоставляет различные функции-фабрики для создания промежуточных обработчиков.

Закрепление материала

- Что такое express?

- Как создать приложение используя фреймворк express?
- Для чего нужен конструктор Router?
- Для чего нужен модуль body-parser?

Дополнительное задание

Задание

Скачайте и подключите модуль express. Создайте приложение используя фреймворк express. Запустите приложение.

Самостоятельная деятельность учащегося

Задание 1

Выучите основные понятия, рассмотренные на уроке.

Задание №2

Создайте несколько html страниц и фалов со стилями, которые подключены к страницам. Создайте приложение используя фреймворк express, которое будет пользователю возвращать данные страницы при запросе.

Задание №3

Создайте несколько html страниц. Добавьте в разметку главное меню, которое будет состоять из пунктов – «Домашняя», «Новости», «О нас». В правом верхнем углу страницы создайте две ссылки – «Вход» и «Регистрация». При переходе по ссылке «Вход», должна возвращаться форма с полями ввода «Логин» и «Пароль» и кнопкой «Войти». При переходе по ссылке «Регистрация», должна возвращаться форма с полями ввода «Логин», «Пароль», «Подтверждение пароля» и «Email», а также кнопка «Зарегистрироваться». При нажатии на кнопку «Вход» на стороне сервера данные о попытке входа должны записываться в файл. При нажатии на кнопку «Зарегистрироваться» данные на стороне сервера должны записываться в файл.

Рекомендуемые ресурсы

Express:

<http://expressjs.com/en/4x/api.html>