



SHIFT
Training center

JavaScript. Уровень 2. Расширенные возможности

Модуль 8. Введение в модули и Node.js

- Что такое модули
- Импорт/экспорт
- Что такое Node.js
- Запуск скриптов
- Пакетный менеджер npm

Что такое Node.js?

- Платформа для разработки веб-приложений
- Сочетание
 - асинхронного ввода/вывода
 - использования JavaScript
 - однопоточной событийно-ориентированной архитектуры
- Автономная виртуальная машина JavaScript
- Основана на JavaScript Chrome V8

Установка Node

- Установщик (msi, pkg, deb)
- Пакетный менеджер (apt, brew, snap)
- Вручную (zip)
- Сборка (из исходного кода)
- Образ Docker

<https://nodejs.org/ru/download/>

Выполнение и тестирование команд

- Инструменты командной строки
- Встроенные асинхронные функции со стрелками
- Парадигма Promise
- Запуск сервера

Модули Node

- Модули и пакеты – строительные блоки приложения
- CommonJS и ES2018/ES6
- `require`, `import` и `import()`
- Использование модулей JSON
- Базовые модули
- Использование каталога
- Репозиторий пакетов `npm`, `yarn`

Обратные вызовы и события в Node

- Передача функции в качестве параметра
- Первым аргументом в колбеке должна быть ошибка
- Отсутствует возвращаемое значение

События и EventEmitter

- `on()` / `addListener()`
- `off()` или `removeListener()`
- `emit()`
- `setImmediate()`

Создание сервера REST

- Отправка и получение событий с использованием шаблона EventEmitter
- Понимание приложения HTTP-сервера
- Фреймворки веб-приложений
- Использование Express framework
- Обработка вычислений в Express-приложении и цикле событий Node.js
- Выполнение запросов HTTP-клиента
- Создание простого сервиса REST с помощью Express

Express.js

- Минималистичный
- Гибкий
- Идеальный для сервера приложений
- Быстрый
- Некатегоричный

Асинхронность и роутинг в Express

- Синхронные ошибки перехватываются Express
- Об асинхронных ошибках необходимо сообщать, вызывая `next(err)`
- Успешное выполнение `middleware`-функции заканчивается вызовом `next()`
- Функция маршрутизатора, которая возвращает результат HTTP-запроса, не вызывает `next()`

Promise и обработка ошибок в Express

- Последний шаг в цепочке Promise использует `res.render`, `res.send`, `res.json`
- Последний `catch` сообщает о любых ошибках с помощью `next(err)`

Парадигма MVC (Model View Controller)

- Каталог views – представления
- Каталог routes – контроллеры
- Каталог models – модели

Модель CRUD

- Create
- Read
- Update
- Delete/Destroy

Методы в моделях