

События

№ урока: 2 Курс: Node.js

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

Обзор, цель и назначение урока

На этом уроке, мы узнаем, что такое событийный цикл в Node.js. Поймем, что такое асинхронная обработка функций и как платформа Node.js обрабатывает функции не блокируя поток выполнения. И научимся работать с событиями, устанавливать на события обработчики, генерировать события и удалять обработчики с событий.

Изучив материал данного занятия, учащийся сможет:

- Понимать устройство событийного цикла на платформе Node.js
- Работать с системными событиями
- Создавать собственные классы, которые поддерживают возможность работы с событиями
- Уметь добавлять обработчики на события
- Уметь удалять обработчики с событий
- Обрабатывать ошибки используя события

Содержание урока

- Понятие событийного цикла в Node.js
- Модуль для работы с событиями
- Основные методы для работы с событиями
- Добавление и удаление обработчиков события
- Передача данных при помощи событий
- Использование событий для обработки ошибок

Резюме

- **Событийный цикл** – это сущность, которая перехватывает и обрабатывает внешние события и конвертирует их в функции обратного вызова.
- На основе событийного цикла в Node.js реализована асинхронность.
- **Событие** – это сообщение, которое возникает в различных точках исполняемого кода при выполнении определенных условий.
- Для работы с событиями в Node.js необходимо подключить модуль **events**.
- Основным классом в модуле *events* является класс **EventEmitter**.
- **EventEmitter** – это основной класс, который реализует работу обработчиков событий и содержит методы для работы с событиями.
- Все объекты, которые должны генерировать события, должны являться экземплярами класса *EventEmitter*.
- Когда происходит генерация события, все функции-обработчики, прикрепленные к данному событию, вызываются асинхронно.
- Для подписки на событие необходимо вызвать один из методов: **on()**, **addListener()**, **once()**.
- Что бы отписать обработчик (удалить) от события нужно воспользоваться методом **removeListener()**.
- Дополнительные методы объекта **EventEmitter**:
 - **listenerCount** – возвращает количество обработчиков, установленных на событие, имя которого передается в параметр методу;
 - **eventName** – возвращает массив с именами событий, которые есть у объекта;
 - **getMaxListeners** – возвращает максимально допустимое количество слушателей

- listeners – возвращает копию массива с функциями обработчиками для указанного события;
- prependListener – добавляет обработчик в начало массива обработчиков;
- setMaxListeners – метод, позволяющий установить максимально допустимое количество обработчиков на события.

Закрепление материала

- Что такое событийный цикл в Node.js?
- Какой модуль используется для работы с событиями?
- Что нужно сделать, чтобы в пользовательском классе (функции конструкторе) можно было производить подписку и отписку с\на события?
- С помощью каких методов мы можем добавить обработчик на событие?
- С помощью каких методов мы можем удалить обработчик с события?
- Как мы можем передать данные с событием?

Дополнительное задание

Задание

Создайте объект EventEmitter, установите несколько обработчиков на события, а затем сгенерируйте их.

Самостоятельная деятельность учащегося

Задание №1

Выучите основные понятия, рассмотренные на уроке

Задание №2

Создайте класс FileStream, который наследуется от класса EventEmitter. Реализуйте в нем методы для чтения и записи файлов. Методы должны выводить в консоль сообщения «файл прочитан» и «файл записан». Реализуйте возможность установки обработчиков на события чтения и записи. И при вызове соответствующих методов генерацию этих событий.

Рекомендуемые ресурсы

Событийный цикл:

<https://nodejs.org/en/docs/guides/event-loop-timers-and-nexttick/>

События:

<https://nodejs.org/dist/latest-v6.x/docs/api/events.html>