Основи роботи з Node.js

Модулі

Node.js використовує модульну систему. Тобто вся вбудована функціональність розбита на окремі пакети чи модулі. Модуль є блоком коду, який може використовуватися повторно в інших модулях.

За потреби ми можемо підключати потрібні нам модулі. Які вбудовані модулі є в node.js і яку функціональність вони надають можна дізнатися з документації .

Для завантаження модулів використовується функція `require()`, до якої передається назва модуля. Наприклад, у першому додатку з попередньої теми для отримання та обробки запиту був необхідний модуль **http** :

```
1 const http = require("http");
```

Після отримання модуля ми зможемо використовувати весь певний у ньому функціонал, який знову ж таки можна подивитися в документації .

Подібним чином ми можемо завантажувати та використовувати інші вбудовані модулі. Наприклад, використовуємо модуль os , який надає інформацію про оточення та операційну систему:

```
1 const os = require("os");
2 // получим имя текущего пользователя
3 let userName = os.userInfo().username;
4
5 console.log(userName);
```

Ми не обмежені вбудованими модулями і за потреби можемо створити свої. Так, у минулій темі проект складався з файлу *app.js* , в якому створювався сервер, який обробляє запити. Додамо до того ж каталогу новий файл *greeting.js* і визначимо в ньому наступний код:

```
1 console.log("greeting module");
```

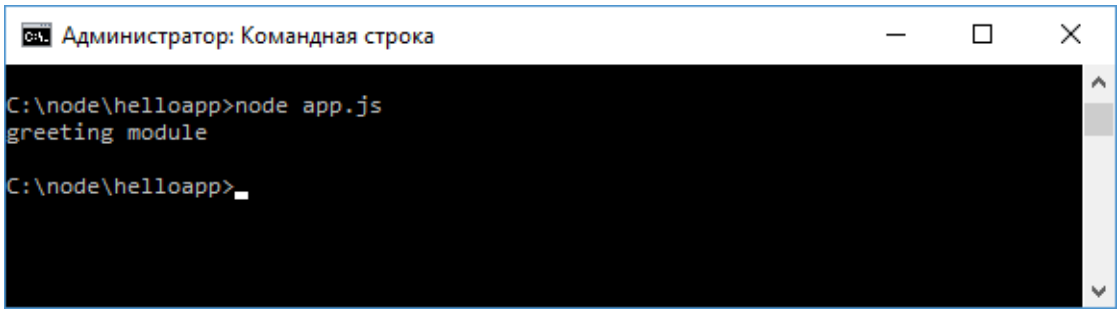
У файлі *app.js* підключимо наш модуль:

```
const greeting = require("./greeting");
```

В отличие от встроенных модулей для подключения своих модулей надо передать в функцию require относительный путь с именем файла (расширение файла необязательно):

```
const greeting = require("./greeting");
```

Запустим приложение:



На консоль выводится та строка, которая определена в файле greeting.js.

Теперь изменим файл *greeting.js*:

```
let currentDate = new Date();
module.exports.date = currentDate;

module.exports.getMessage = function(name){
    let hour = currentDate.getHours();
    if(hour > 16)
        return "Добрый вечер, " + name;
    else if(hour > 10)
        return "Добрый день, " + name;
    else
        return "Доброе утро, " + name;
}
```

Здесь определена переменная `currentDate`. Однако из вне она недоступна. Она доступна только в пределах данного модуля. Чтобы какие переменные или функции модуля были доступны, необходимо определить их в объекте **module.exports**. Объект `module.exports` - это то, что возвращает функция `require()` при получении модуля.

Вообще объект **module** представляет ссылку на текущий модуль, а его свойство **exports** определяет все свойства и методы модуля, которые могут быть экспортированы и использованы в других модулях. Подробнее определение загрузки модуля и все его функции можно посмотреть на странице https://github.com/nodejs/node/blob/master/lib/module.js.

В частности, здесь определяется свойство `date` и метод `getMessage`, который принимает некоторый параметр.

Далее изменим файл *app.js*:

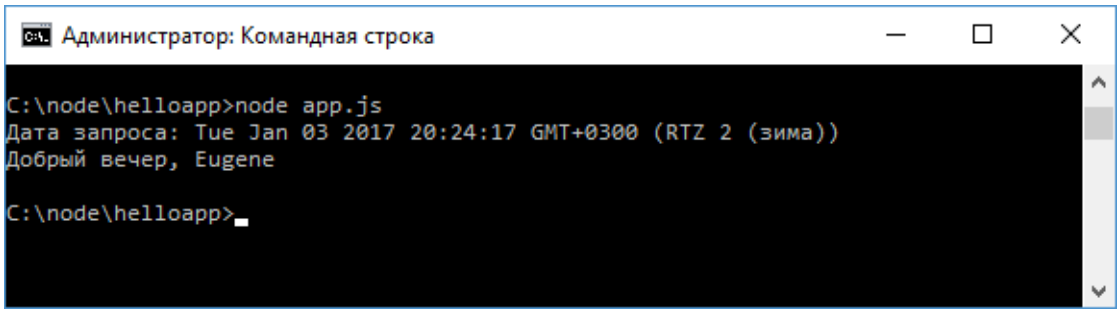
```
const os = require("os");
const greeting = require("./greeting");

// получим имя текущего пользователя
let userName = os.userInfo().username;


console.log(`Дата запроса: ${greeting.date}`);
console.log(greeting.getMessage(userName));
```

Все экспортированные методы и свойства модуля доступны по имени: `greeting.date` и `greeting.getMessage()`.

Перезапустим приложение:



Определение конструкторов и объектов в модуле

Крім визначення найпростіших функцій або властивостей у модулі можуть визначатися складні об'єкти або функції конструкторів, які використовуються для створення об'єктів. Так, додамо до папки проекту новий файл *user.js* :

```
function User(name, age){

    this.name = name;
    this.age = age;
    this.displayInfo = function(){

        console.log(`Имя: ${this.name}  Возраст: ${this.age}`);
    }
}
User.prototype.sayHi = function() {
    console.log(`Привет, меня зовут ${this.name}`);
};

module.exports = User;
```

Тут визначено стандартну функцію конструктора User, яка приймає два параметри. При цьому весь модуль тепер вказує на цю функцію конструктора:

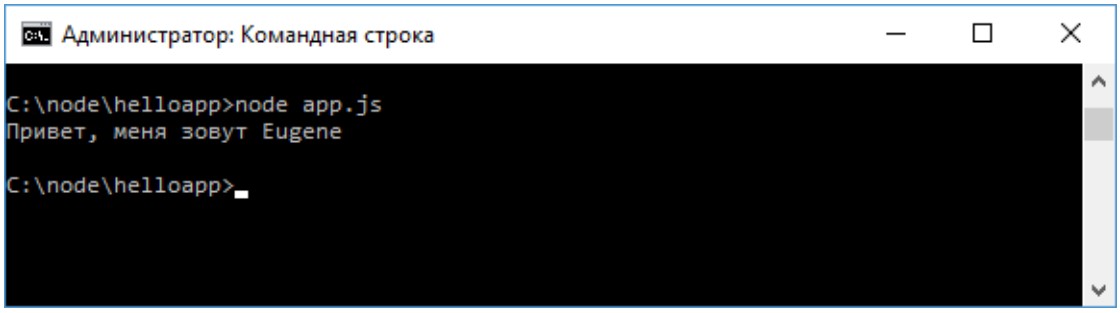
```
module.exports = User;
```

Підключимо та використовуємо цей модуль у файлі **app.js** :

```
const User = require("./user.js");

let eugene = new User("Eugene", 32);
eugene.sayHi();
```

Запустимо додаток:



Назад Зміст Вперед