



Что такое npm

Менеджер пакетов, входящий в состав Node.js

npm используется для подключения дополнительных модулей в проект

Посмотреть существующие пакеты можно на сайте <https://www.npmjs.com/>

Как работать с prn

prn запускается в терминале. Команда для запуска - prn

Команды npm

Установка модулей:

```
npm install <имя модуля>
```

Есть короткая запись “npm i”

```
npm install express
```

express - модуль для работы с сервером, улучшенная версия http

Простой пример работы с express

```
let express = require("express");
```

```
let app = express();
```

```
app.get("/", (req, res)=>{  
  res.end("EXPRESS!");  
});
```

```
app.listen(3000);
```

Установка пакетов

После установки в папке с проектом окажется папка `node_modules`, которая и будет содержать все модули

Дополнительные команды install

`npm install -g <имя пакета>` - установка пакета глобально

`npm uninstall <имя пакета>`

Удаление пакета

package.json

После установки пакета вам становится доступен package.json

Его можно создать и без установки пакетов.

package.json - это заготовка для вашего проекта, там хранятся вся информация о проекте.

Структура package.json

Создайте пустую папку, откройте папку с помощью Visual Studio Code, создайте там файл package.json.

Поместите туда следующий код

```
{  
  "name": "testapp",  
  "version": "1.0.0",  
  "dependencies": {  
    "express": "^4.17.1"  
  }  
}
```

Запустите команду `npm install`

Все необходимые файлы установятся в папку с проектом

Дополнительные возможности package.json

Добавьте файл index.js с кодом из предыдущего примера. Дополните файл package.json следующим кодом

```
"scripts": {  
  "start": "node index.js"  
}
```

Вызовите команду npm start

Раздел scripts

Раздел scripts позволяет выполнять различные команды, не разбираясь в структуре проекта.

Предустановленные команды выполняются следующим образом:

```
npm <имя команды: start, test и др>
```

Команды, придуманные нами выполняются таким образом:

```
npm run <имя команды>
```

Пример собственной команды

```
"scripts": {  
  "start": "node index.js",  
  "say": "npm -v"  
}
```

npm run say

Избранные поля package.json

name	Имя пакета
version	Версия пакета
description	Описание пакета
keywords	Ключевые поля пакета (массив)
homepage	Домашняя страница пакета
dependencies	Зависимости пакета
devDependencies	Зависимости на этапе разработки

package.json и package-lock.json

package.json указывает зависимости от версии: там можно указать вариации версий.

Файл package-lock.json хранит в неизменном виде сведения о версии каждого установленного пакета и npm будет использовать именно эти версии пакетов при выполнении команды `npm install`.

Создание пакета автоматически

Команда `npm init` начнёт создание `package.json` файла

Данную команду можно использовать вместе с инициализаторами, помогающими установить пакет с нужными зависимостями, файлами и т.д.

Установите полезный модуль nodemon

Он позволяет перезагружать сервер автоматически при сохранении изменений.

Использование:

```
nodemon <имя файла>
```


Примечание. Установку произведите глобально

Публикация собственных пакетов

Только теория!!!

Не хочется забивать своими бесполезными пакетами мощную структуру.

Действие 1. Регистрация



Sign Up

Full Name

Public Email

Username

Password

Действие 2. Вход в npm

`npm login`

После вам нужно будет ввести логин, пароль и почту

Шаг 3. Создание пакета

```
npm init
```

Шаг 4. Публикация

`npm publish`

Задание

Создать сервер, генерирующий новости.

Новости хранить в отдельном JSON файле (использовать fs).

Новость состоит из заголовка, картинки и текста.

При запросе выдаётся только одна новость. При следующем запросе выдаётся следующая новость.

При достижении конца либо вывести текст «Новостей больше нет», либо начать сначала.

Frontend должен работать на ajax (можете подключить jquery, но сделайте это через npm).

Дополнительно – реализовать добавление новостей.