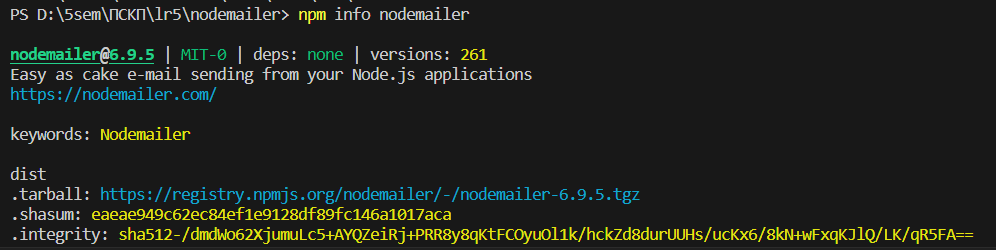
Лабораторная работа 05

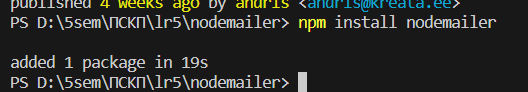
ПСКП

**Задание 01**

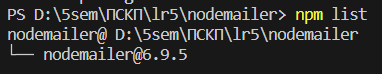
1. С помощью **npm** получите информацию о пакете **nodemailer**.

****

1. С помощью **npm** скачайте пакет **nodemailer** в локальный директорий.

****

1. С помощью **npm** убедитесь, что пакет установлен.



**Задание 02**

1. Разработайте приложение **06-02**, использующее пакет **nodemailer.**
2. Приложение **06-02**, должно отправлять на браузер HTML-страницу, позволяющую ввести почтовые ящики отправителя и получателя, а также пересылаемое сообщение.
3. Убедитесь в работоспособности приложении.

**Задание 03**

1. Разработайте модуль **m06\_XXX**, где XXX – ваши инициалы, который экспортирует одну функцию **send**.
2. Функция **send** принимает три параметра: почтовый адрес, на с которого будет отправлено сообщение, пароль и само сообщение.
3. Результатом выполнения функции **send**, является почтовое сообщение (email), содержащее строку, принятую в качестве параметра. Отправка сообщения осуществляется с помощью пакета **nodemailer**.
4. Сообщение, отправляемое функцией **send,** отправляется на указанный почтовый адрес.
5. Разработайте приложение **06-03,** применяющее модуль **m06\_XXX** и демонстрирующее работу функции **send** (примерный вид приложения представлен на сл. рис).



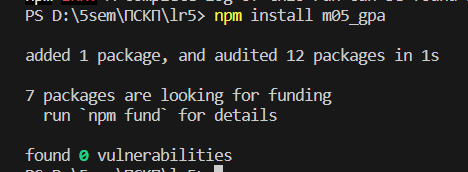
1. Убедитесь в работоспособности приложения.

**Задание 04**

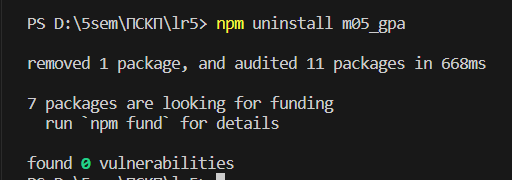
1. Опубликуйте пакет **m06\_XXX**, содержащий соответствующий модуль.
2. Зайдите на сайт <http://npmjs.com> и убедитесь, что пакет опубликован.



1. Скачайте пакет в локальный репозиторий и разработайте приложение **06-04,** демонстрирующее применение скачанного пакета.



1. Удалите пакет **m06\_XXX** из локального репозитория.



1. Установите пакет **m06\_XXX** в глобальный репозиторий и проверьте работоспособность приложения **06-04.**

**Задание 05**

1. Поясните назначение **npm**.

NPM – Node Package Manager – чухня, которая устанавливается вместе с нодой. Нужна для простой установки любого пакета, созданного другим разработчиком. Самих пакетов огромное количество и любой человек может его опубликовать.

1. Перечислите команды **npm**,с помощью которых можно просмотреть список установленных пакетов.

**npm list, npm ls, npm la, npm ll**

1. Поясните назначение параметра **–g** в командах **npm.**

Устанавливает пакет в глобальный репозиторий (обычно по пути установки npm Users/UserName/AppData/Roaming/npm).

Этот путь можно узнать с помощью команды **npm root -g**.

Как и с глобальными объектами, рекомендуется не загружать пакеты в глобальный репозиторий. Обычно пакеты загружаются глобально если я хочу с ними взаимодействовать через коммандную строку. Иначе, без параметра -g, пакеты устанавливаются локально, в рабочую директорию, и применяются в проектах через require или import.

1. Перечислите команды **npm**,с помощью которых можно скачать пакет.

**npm install, npm i** (allias).

После команды можно прописать необязательные параметры: путь установки, @scope, чтобы установить пакет в свою область видимости, --save чтобы также добавить запись о добавленной зависимости в package.json (после npm 5 эти записи добавляются автоматически), -g для установки в глобальный репозиторий, express@1.0.5 для установки конкретной версии, а также можно указать прямую ссылку на гитхаб пакета.

1. Поясните назначение файла **package.json**.

Основной файл кофигурации npm, нечто вроде файла манифеста. В простейшем случае, если вы просто устанавливаете пакеты, то в нем будут прописаны названия зависимостей локального репозитория npm. При создании же пакетов необходимо туда записать инфу о вашем пакете.

1. Перечислите последовательность действий, позволяющих опубликовать пакет с помощью **npm**.

**npm login** – сначала надо войти в свой аккаунт npm.

**npm init** – инциализируем пакет в текущей директории; создается файл package.json, который далее будет конфигурировать через командную строку.

**npm publish** – просто выгрузка пакета на ваш аккаунт, аналогия git push