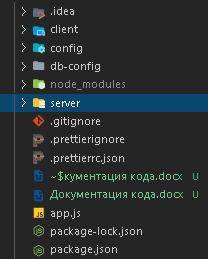
# Файловая структура

В данном разделе описывается общая файловая структура проекта более подробная информация косательно каждой папки будет описана в подразделах.



Корневой каталог веб системы содержит в себе такие папки/файлы:

* **client** – данная папка сдержит в себе полностью веб клиент приложения(frontend), так-же там содержаться модули необходимые для работы для работы приложения.
* **config –** может содержать в себе любые необходимые конфигурационные файлы которые могут использоваться для дальнейшего использования стороной сервера
* **db-config –** хранит в себе файлы необходимые для установки соединения с базой данных .
* **server –**  является представлением сервера содержит все необходимое для работы сервера.
* **.gitingore –** список папок и файлов которые игнорируются при загрузки проекта на GitHub.
* **app.js –** файл для запуска сервера веб приложения при условии что клиентская часть (client) собрана и готова к запуску.

## client

## app.js

## config



Содержит два файла

* default.json
* production.json

Каждый из файлов содержит в себе адрес порта который будет использоваться для запуска веб приложения в виде изображенном ниже



## db-config



Содержит два файла:

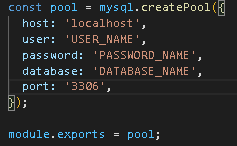
* mysql-config-example.js
* mysql-config.js

Данные файлы подключают модуль ‘**mysql’** который используется для подключения к базе данных, подключение происходит с использованием строки



Что позволяет настроить и подключится к базе данных используемой проектом.

Для настройки подключения используется структура



Таким образом мы создаем **pool** подключение и делаем его экспорт для того чтобы многоразово его использовать в ходе работы приложения.

Описание полей(все поля представляют собой строку):

**host**: адрес для подключения к базе данных

**user**: имя пользователя который подключается к базе(определяется базой или сервером на котором размещено)

**password**: пароль для доступа к базе

**database:** имя базы данных к которой создается подключение

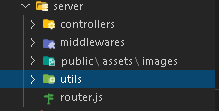
**port**: порт подключения, свойство может отсутствовать, в таком случае используется стандартное значение **3306**

Различия между двумя файлами только одно, адрес подключения, mysql-config.js содержит в себе путь к серверу на котором размещена база данных, данный файл используется приложением как основной для подключения к базе данных и взаимодействия с ней.

За более подробной информацией стоит обратится по ссылке:

* <https://www.npmjs.com/package/mysql>

## server



Папки:

**controllers** – содержит в себе файлы с функциями которые использует сервер при

получении запросов, именно здесь размещаются обработчики запросов.

**middleware –** определяет middlewares необходимые для определения заголовков запроса.

**public \assets\images**– данный файл содержит в себе картинки которые загружаются на стороне клиента в случае необходимости, в основном тут хранятся картинки которые используются для визуального отображения POI маркеров.

**utils** – содержит в себе вспомогательный функционал который может использоваться как постоянно так и единожды, например: загрузки данных о загрязнении с какого-то сайта.

Файл:

**router.js –** представляет собой роутер предоставляемый модулем **express**, данный роутер слушает запросы по адресам и вызывает свойственные им оброботчики расположеные в файле **controllers**

### **router.js**



Сначала подключаются модули, в данном случае это:

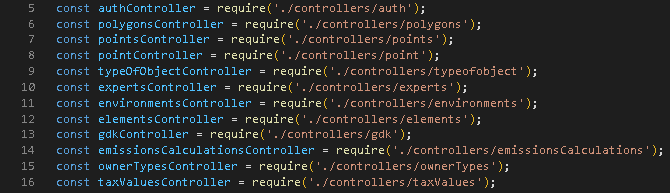
**path**- модуль представляющий собой копию модуля **path** из NodeJs

**express**- фреймор NodeJs который и использовался для создания сервера, необходим для рабботы с роутером.

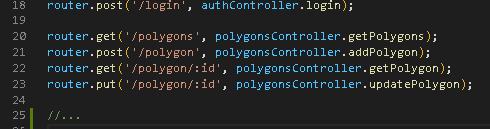
Подробнее: <http://expressjs.com/>

**router**- модуль express необходимый для создания роутера.

Далее идёт подключение оброботчиков из файла **controllers**;



После подключения модулей происходит распределение путей и указание обробтчиков для каждого из путей.



Где **router** модуль определяющий роутер, после чего идёт **.put/.get/.post…** определяющий тип запроса , **(‘/login’)** определяет путь по которому которому обрабатывается запрос, после чего через запятую ставится функция-оброботчик **authController.login** которая обрабатывает запрос после его получения.

## .gitigonre

# Сборка и запуск