Сервер приложений

Содержание

[Введение 5](#_Toc36206088)

[1 Технические характеристики 6](#_Toc36206089)

[1.1 Среда разработки 6](#_Toc36206090)

[2 Описание программы 7](#_Toc36206091)

[2.1 Сторонние библиотеки 7](#_Toc36206092)

[2.2 Структура RoleAccess.json 8](#_Toc36206093)

[2.3 Настроечный файл Config.toml 8](#_Toc36206094)

[2.4 Структуры 9](#_Toc36206095)

[2.4.1 Account 9](#_Toc36206096)

[2.4.2 AreaInfo 10](#_Toc36206097)

[2.4.3 BoxPoint 10](#_Toc36206098)

[2.4.4 BusyArm 11](#_Toc36206099)

[2.4.5 CacheData 11](#_Toc36206100)

[2.4.6 CheckData 11](#_Toc36206101)

[2.4.7 Config 11](#_Toc36206102)

[2.4.8 DBConfig 12](#_Toc36206103)

[2.4.9 DeviceLog 12](#_Toc36206104)

[2.4.10 DeviceLogInfo 12](#_Toc36206105)

[2.4.11 EditCrossInfo 13](#_Toc36206106)

[2.4.12 FillingInfo 13](#_Toc36206107)

[2.4.13 Locations 13](#_Toc36206108)

[2.4.14 PassChange 13](#_Toc36206109)

[2.4.15 Permission 14](#_Toc36206110)

[2.4.16 PngSettings 14](#_Toc36206111)

[2.4.17 Point 14](#_Toc36206112)

[2.4.18 Privilege 14](#_Toc36206113)

[2.4.19 RegionInfo 15](#_Toc36206114)

[2.4.20 Role 15](#_Toc36206115)

[2.4.21 RoleAccess 15](#_Toc36206116)

[2.4.22 RoleData 16](#_Toc36206117)

[2.4.23 RouteInfo 16](#_Toc36206118)

[2.4.24 SelectedData 16](#_Toc36206119)

[2.4.25 ServerLogInfo 16](#_Toc36206120)

[2.4.26 ShortAccount 17](#_Toc36206121)

[2.4.27 TLSostInfo 17](#_Toc36206122)

[2.4.28 Token 18](#_Toc36206123)

[2.4.29 TrafficLights 18](#_Toc36206124)

[2.4.30 LogFile 18](#_Toc36206125)

[2.4.31 StateResult 19](#_Toc36206126)

[2.4.32 ArmCommandMessage 19](#_Toc36206127)

[2.4.33 StateMessage 19](#_Toc36206128)

[2.4.34 TCPConfig 20](#_Toc36206129)

[2.5 Утилиты 20](#_Toc36206130)

[2.6 Middleware route 20](#_Toc36206131)

[2.7 Обработчики 21](#_Toc36206132)

[2.7.1 ActAddAccount 21](#_Toc36206133)

[2.7.2 ActChangePw 21](#_Toc36206134)

[2.7.3 ActDeleteAccount 21](#_Toc36206135)

[2.7.4 ActUpdateAccount 21](#_Toc36206136)

[2.7.5 BuildCross 21](#_Toc36206137)

[2.7.6 BuildMainPage 22](#_Toc36206138)

[2.7.7 CheckAllCross 22](#_Toc36206139)

[2.7.8 CheckSelectedDirCross 22](#_Toc36206140)

[2.7.9 ControlCheckButton 22](#_Toc36206141)

[2.7.10 ControlCloseCross 22](#_Toc36206142)

[2.7.11 ControlCreateButton 23](#_Toc36206143)

[2.7.12 ControlCross 23](#_Toc36206144)

[2.7.13 ControlDeleteButton 23](#_Toc36206145)

[2.7.14 ControlEditableCross 23](#_Toc36206146)

[2.7.15 ControlSendButton 23](#_Toc36206147)

[2.7.16 ControlTestState 24](#_Toc36206148)

[2.7.17 CrossEditFree 24](#_Toc36206149)

[2.7.18 CrossEditInfo 24](#_Toc36206150)

[2.7.19 DevCrossInfo 24](#_Toc36206151)

[2.7.20 DispatchControlButton 24](#_Toc36206152)

[2.7.21 DisplayAccInfo 25](#_Toc36206153)

[2.7.22 DisplayDeviceLogFile 25](#_Toc36206154)

[2.7.23 DisplayServerLogFile 25](#_Toc36206155)

[2.7.24 DisplayServerLogInfo 25](#_Toc36206156)

[2.7.25 LocationButtonMapPage 25](#_Toc36206157)

[2.7.26 LogDeviceInfo 26](#_Toc36206158)

[2.7.27 LoginAcc 26](#_Toc36206159)

[2.7.28 LoginAccOut 26](#_Toc36206160)

[2.7.29 MainCrossCreator 26](#_Toc36206161)

[2.7.30 MakeSelectedDirCross 26](#_Toc36206162)

[2.7.31 UpdateMapPage 27](#_Toc36206163)

# Введение

Сервер приложений (TLServer) осуществляет обработку запросов от пользователей.

Данные передаются посредством протокола HTTPS (HyperText Transfer Protocol Secure - протокол передачи [гипертекста](https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%BF%D0%B5%D1%80%D1%82%D0%B5%D0%BA%D1%81%D1%82) защищенный) (расширением протокола HTTP) в целях повышения безопасности. Данные в протоколе HTTPS передаются поверх криптографических протоколов TLS.

Хранение информации осуществляется посредством  [свободн](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%9F%D0%9E)ой [объектно-реляционная](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D1%80%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94) [системы управления базами данных](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) PostgreSQL, базируется на языке [SQL](https://ru.wikipedia.org/wiki/SQL) и поддерживает многие из возможностей стандарта SQL.

Сервер приложений осуществляет связь посредством запросов с сервером устройств [Ag-Server](https://github.com/ruraomsk/ag-server).

1 Технические характеристики

* 1. Среда разработки

Языком программирования выбран GOlang - [компилируемый](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D0%B8%D1%80%D1%83%D0%B5%D0%BC%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) [многопоточный](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D0%B0%D0%BB%D0%BB%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B) [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), разработанный внутри компании [Google](https://ru.wikipedia.org/wiki/Google_(%D0%BA%D0%BE%D0%BC%D0%BF%D0%B0%D0%BD%D0%B8%D1%8F)).

Версия языка на 03.2020 является 1.14.1.

# 2 Описание программы

## 2.1 Сторонние библиотеки

1) github.com/BurntSushi/toml v0.3.1

Библиотека для работы с TOML (формат конфигурационных файлов), позволяется кодировать и раскодировать файлы данного формата.

2) github.com/dgrijalva/jwt-go v3.2.0+incompatible

Библиотека поддерживает синтаксический анализ и проверку, а также генерацию и подписание JWT (json web token). В настоящее время поддерживаются следующие алгоритмы подписи: HMAC SHA, RSA, RSA-PSS и ECDSA.

3) github.com/gorilla/handlers v1.4.2

Библиотека обработчика пакетов – содержит в себе набор инструментов, позволяющих писать промежуточные обработчики запроса.

4) github.com/gorilla/mux v1.7.4

Библиотека реализует маршрутизатор и диспетчер запросов для сопоставления входящих запросов с их соответствующим обработчиком.

5) github.com/jinzhu/gorm v1.9.12

Библиотека реализует интерфейсы для работы с БД.

6) github.com/lib/pq v1.3.0 // indirect

Драйвер для работы с postgresql.

7) github.com/pkg/errors v0.9.1

Библиотека, обеспечивающая простую обработку ошибок.

8) golang.org/x/crypto v0.0.0-20200323165209-0ec3e9974c59

Стандартная библиотека, предоставляющая возможности по шифрованию информации.

9) github.com/ruraomsk/ag-server v0.0.0-20200325075148-7bcc97b69ad3

Сервер управления АСУД.

## 2.2 Структура RoleAccess.json

RoleAccess.json содержит информацию для осуществления контроля доступа пользователей к ресурсам.

Файл содержит 3 ключа:

1. roles – информация о роли

ключи:

name – имя,

permissions - массив разрешений;

1. permissions – информация о разрешении

ключи:

id – уникальный номер,

description – описание,

visible - флаг видимости, необходим для работы с гибкими ролями;

1. routes – информация о маршруте

ключи:

id – уникальный номер,

description – описание,

permission – к какому разрешению относится маршрут,

path – маршрут ресурса.

## 2.3 Настроечный файл Config.toml

Config.toml содержит всю настроечную информацию для корректной работы сервера.

Структура toml файла:

logger\_path = "" //путь до каталога с логами сервера

views\_path = "" //путь до каталога views (содержит все ресурсы для отображения перекрестков)

resourcePath = "" //путь до каталога frontend

cache\_path = "" //путь до каталога cachefile

server\_ip = " //ip сервера / порт

ya\_key = "" //ключ авторизации для яндекса

token\_password = "" //ключ для шифрования токенов доступа

Настройка подключения к тсп серверу обемена командами  
[tcpServer]  
tcpServerAddress = "" //адресс сервера  
portState = " //порт для обмена Стате  
portArmCommand = " //порт для обмена арм командами  
  
Настройки для постгреса  
[database]  
db\_name = "" //имя бд  
db\_password = "" //пароль доступа к бд  
db\_user = "" //пользователь для обращения к бд  
db\_type = "" //тип бд  
db\_host = "" //ip сервера бд  
db\_port = "" //порт обращения к бд  
db\_SetMaxOpenConst = //максимальное количество пустых соединений с бд  
db\_SetMaxIdleConst = //максимальное количество соединений с бд  
cross\_table = "" //название таблицы cross  
account\_table = "" //название таблицы accout  
region\_table = "" //название таблицы region  
devices\_table = "" //название таблицы divices  
status\_table = "" //название таблицы status  
logDevice\_table = "" //название таблицы logDevice  
  
Настройки для работы с каталогами перекрестков  
[picture]  
png\_sizeX = //размер картинки по координате X

png\_sizeY = //размер картинки по координате Y  
png\_Z = //величина отдаления

## 2.4 Структуры

### 2.4.1 Account

Account структура аккаунта пользователя

type Account struct {

[gorm](http://localhost:6060/pkg/github.com/jinzhu/gorm/).[Model](http://localhost:6060/pkg/github.com/jinzhu/gorm/#Model)

Login [string](http://localhost:6060/pkg/builtin/#string) `json:"login",sql:"login"` //Имя пользователя

Password [string](http://localhost:6060/pkg/builtin/#string) `json:"password"` //Пароль

BoxPoint [BoxPoint](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#BoxPoint) `json:"boxPoint",sql:"-"` //Точки области отображения

WorkTime [time](http://localhost:6060/pkg/time/).[Duration](http://localhost:6060/pkg/time/#Duration) `json:"workTime",sql:"workTime"` //Время работы пользователя в часах

YaMapKey [string](http://localhost:6060/pkg/builtin/#string) `json:"ya\_key",sql:"-"` //Ключ доступа к яндекс карте

Token [string](http://localhost:6060/pkg/builtin/#string) `json:"token",sql:"-"` //Токен пользователя

}

Методы:

1. ChangePW изменение пароля пользователя

func (account \*[Account](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Account)) ChangePW() map[[string](http://localhost:6060/pkg/builtin/#string)]interface{}

1. Create создание аккаунта для пользователей

func (account \*[Account](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Account)) Create(privilege [Privilege](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Privilege)) map[[string](http://localhost:6060/pkg/builtin/#string)]interface{}

1. Delete удаление аккаунта из БД

func (account \*[Account](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Account)) Delete() map[[string](http://localhost:6060/pkg/builtin/#string)]interface{}

1. GetInfoForUser собор информацию для пользователя, который авторизировался

func (account \*[Account](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Account)) GetInfoForUser() map[[string](http://localhost:6060/pkg/builtin/#string)]interface{}

1. ParserBoxPointsUser заполняет BoxPoint

func (account \*[Account](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Account)) ParserBoxPointsUser() (err [error](http://localhost:6060/pkg/builtin/#error))

1. Update обновление данных аккаунта

func (account \*[Account](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Account)) Update(privilege [Privilege](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Privilege)) map[[string](http://localhost:6060/pkg/builtin/#string)]interface{}

1. Validate проверка аккаунда в бд

func (account \*[Account](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Account)) Validate() (map[[string](http://localhost:6060/pkg/builtin/#string)]interface{}, [bool](http://localhost:6060/pkg/builtin/#bool))

### 2.4.2 AreaInfo

AreaInfo информация о районе

type AreaInfo struct {

Num [string](http://localhost:6060/pkg/builtin/#string) `json:"num"` //уникальный номер района

NameArea [string](http://localhost:6060/pkg/builtin/#string) `json:"nameArea"` //расшифровка номера

}

Методы:

1. SetAreaInfo установить в структуру номер и имя района по номеру района и региона

func (area \*[AreaInfo](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#AreaInfo)) SetAreaInfo(numReg, numArea [string](http://localhost:6060/pkg/builtin/#string))

2.4.3 BoxPoint

BoxPoint координаты для отрисовки зоны работы пользователя

type BoxPoint struct {

Point0 [Point](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Point) `json:"point0"` //левая нижняя точка на карте

Point1 [Point](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Point) `json:"point1"` //правая верхняя точка на карте

}

2.4.4 BusyArm

BusyArm информация о занятом перекрестке

type BusyArm struct {

Region [string](http://localhost:6060/pkg/builtin/#string) `json:"region"` //регион

Area [string](http://localhost:6060/pkg/builtin/#string) `json:"area"` //район

ID [int](http://localhost:6060/pkg/builtin/#int) `json:"ID"` //ID

Description [string](http://localhost:6060/pkg/builtin/#string) `json:"description"` //описание

}

2.4.5 CacheData

CacheData Данные для обновления в определенный период

type CacheData struct {

mapRegion map[string]string //регионы  
mapArea map[string]map[string]string //районы  
mapTLSost map[int]string //светофоры

}

2.4.6 CheckData

CheckData структура проверки для перекрестков

type CheckData struct {

ID [string](http://localhost:6060/pkg/builtin/#string) `json:"ID"` //ID устройства

PngStatus [bool](http://localhost:6060/pkg/builtin/#bool) `json:"pngStatus"` //флаг наличия map.png

SvgStatus [bool](http://localhost:6060/pkg/builtin/#bool) `json:"svgStatus"` //флаг наличия cross.svg

}

2.4.7 Config

Config структура с объявлением всех переменных config.toml файла

type Config struct {

LoggerPath [string](http://localhost:6060/pkg/builtin/#string) `toml:"logger\_path"` //путь до каталога с логами сервера

ViewsPath [string](http://localhost:6060/pkg/builtin/#string) `toml:"views\_path"` //путь до каталога views (содержит все ресурсы для отображения перекрестков)

CachePath [string](http://localhost:6060/pkg/builtin/#string) `toml:"cache\_path"` //путь до каталога cachefile

ResourcePath [string](http://localhost:6060/pkg/builtin/#string) `toml:"resourcePath"` //путь до каталога frontend

ServerIP [string](http://localhost:6060/pkg/builtin/#string) `toml:"server\_ip"` //ip сервера / порт

YaKey [string](http://localhost:6060/pkg/builtin/#string) `toml:"ya\_key"` //ключ авторизации для яндекса

TokenPassword [string](http://localhost:6060/pkg/builtin/#string) `toml:"token\_password"` //ключ для шифрования токенов доступа

TCPConfig [tcpConnect](http://localhost:6060/pkg/github.com/JanFant/TLServer/tcpConnect/).[TCPConfig](http://localhost:6060/pkg/github.com/JanFant/TLServer/tcpConnect/#TCPConfig) `toml:"tcpServer"` //информация о tcp соединении с сервером устройств

DBConfig [DBConfig](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#DBConfig) `toml:"database"` //информация о соединении с базой данных

PngSettings [PngSettings](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#PngSettings) `toml:"picture"` //инфомарция о настройках подложки

}

2.4.8 DBConfig

DBConfig настройки для работы с базой данных

type DBConfig struct {

Name [string](http://localhost:6060/pkg/builtin/#string) `toml:"db\_name"` //имя БД

Password [string](http://localhost:6060/pkg/builtin/#string) `toml:"db\_password"` //пароль доступа к БД

User [string](http://localhost:6060/pkg/builtin/#string) `toml:"db\_user"` //пользователя для обращения к бд

Type [string](http://localhost:6060/pkg/builtin/#string) `toml:"db\_type"` //тип бд

Host [string](http://localhost:6060/pkg/builtin/#string) `toml:"db\_host"` //ip сервера бд

Port [string](http://localhost:6060/pkg/builtin/#string) `toml:"db\_port"` //порт для обращения к бд

SetMaxOpenConst [int](http://localhost:6060/pkg/builtin/#int) `toml:"db\_SetMaxOpenConst"` //максимальное количество пустых соединений с бд

SetMaxIdleConst [int](http://localhost:6060/pkg/builtin/#int) `toml:"db\_SetMaxIdleConst"` //максимальное количество соединенияй с бд

CrossTable [string](http://localhost:6060/pkg/builtin/#string) `toml:"cross\_table"` //название таблицы cross

RegionTable [string](http://localhost:6060/pkg/builtin/#string) `toml:"region\_table"` //название таблицы region

AccountTable [string](http://localhost:6060/pkg/builtin/#string) `toml:"account\_table"` //название таблицы account

DevicesTable [string](http://localhost:6060/pkg/builtin/#string) `toml:"devices\_table"` //название таблицы devices

StatusTable [string](http://localhost:6060/pkg/builtin/#string) `toml:"status\_table"` //название таблицы status

LogDeviceTable [string](http://localhost:6060/pkg/builtin/#string) `toml:"logDevice\_table"` //название таблицы logDevice

}

2.4.9 DeviceLog

DeviceLog описание таблицы, хранящей лог от устройств

type DeviceLog struct {

Time [time](http://localhost:6060/pkg/time/).[Time](http://localhost:6060/pkg/time/#Time) `json:"time"` //время записи

ID [int](http://localhost:6060/pkg/builtin/#int) `json:"id"` //id устройства которое прислало информацию

Text [string](http://localhost:6060/pkg/builtin/#string) `json:"text"` //информация о событие

Devices [BusyArm](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#BusyArm) `json:"devices"` //информация о девайсе

}

2.4.10 DeviceLogInfo

DeviceLogInfo структура запроса пользователя за данными в бд

type DeviceLogInfo struct {

Devices [][BusyArm](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#BusyArm) `json:"devices"` //информация о девайсах

TimeStart [time](http://localhost:6060/pkg/time/).[Time](http://localhost:6060/pkg/time/#Time) `json:"timeStart"` //время начала отсчета

TimeEnd [time](http://localhost:6060/pkg/time/).[Time](http://localhost:6060/pkg/time/#Time) `json:"timeEnd"` //время конца отсчета

}

2.4.11 EditCrossInfo

EditCrossInfo информация о пользователе, занявшем перекресток на изменение

type EditCrossInfo struct {

Login [string](http://localhost:6060/pkg/builtin/#string) `json:"login"` //логин пользователя

EditFlag [bool](http://localhost:6060/pkg/builtin/#bool) `json:"editFlag"` //флаг разрешения на редактирование перекрестка

Kick [bool](http://localhost:6060/pkg/builtin/#bool) `json:"kick"` //флаг закрытия арма у данного пользователя

// contains filtered or unexported fields

}

2.4.12 FillingInfo

FillingInfo информация о запросе

type FillingInfo struct {

User [string](http://localhost:6060/pkg/builtin/#string) //пользователь запросивший заполнение таблицы

Status [bool](http://localhost:6060/pkg/builtin/#bool) //статус выполнения запроса

}

2.4.13 Locations

Locations информация о запрашиваемом регионе и районе карты

type Locations struct {

Region [string](http://localhost:6060/pkg/builtin/#string) `json:"region"` //регион

Area [][string](http://localhost:6060/pkg/builtin/#string) `json:"area"` //районы

}

Методы:

1. MakeBoxPoint расчет координат для перемещения по карте

func (location \*[Locations](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Locations)) MakeBoxPoint() (box [BoxPoint](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#BoxPoint), err [error](http://localhost:6060/pkg/builtin/#error))

2.4.14 PassChange

PassChange структура для изменения пароля

type PassChange struct {

OldPW [string](http://localhost:6060/pkg/builtin/#string) `json:"oldPW"` //старый пароль

NewPW [string](http://localhost:6060/pkg/builtin/#string) `json:"newPW"` //новый пароль

}

Методы:

1. ValidOldNewPW проверка данных полученных от пользователя для изменения своего пароля

func (passChange \*[PassChange](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#PassChange)) ValidOldNewPW(login [string](http://localhost:6060/pkg/builtin/#string)) (account \*[Account](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Account), err [error](http://localhost:6060/pkg/builtin/#error))

### 2.4.15 Permission

Permission структура полномочий содержит ID, команду и описание команды

type Permission struct {

ID [int](http://localhost:6060/pkg/builtin/#int) `json:"id"` //ID порядковый номер

Visible [bool](http://localhost:6060/pkg/builtin/#bool) `json:"visible"` //флаг отображения пользователю

Description [string](http://localhost:6060/pkg/builtin/#string) `json:"description"` //описание команды

}

### 2.4.16 PngSettings

PngSettings настройки размеров создаваемой map.png

type PngSettings struct {

SizeX [int](http://localhost:6060/pkg/builtin/#int) `json:"sizeX",toml:"png\_sizeX"` //размер по координате X

SizeY [int](http://localhost:6060/pkg/builtin/#int) `json:"sizeY",toml:"png\_sizeY"` //размер по координате Y

Z [int](http://localhost:6060/pkg/builtin/#int) `json:"z",toml:"png\_Z"` //величина отдаление

}

### 2.4.17 Point

Point координаты точки

type Point struct {

Y, X [float64](http://localhost:6060/pkg/builtin/#float64) //Координата Х и Y

}

Методы:

1. GetPoint возврашает значение координаты

func (points \*[Point](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Point)) GetPoint() (y, x [float64](http://localhost:6060/pkg/builtin/#float64))

1. SetPoint задать значение координаты

func (points \*[Point](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Point)) SetPoint(y, x [float64](http://localhost:6060/pkg/builtin/#float64))

1. StrToFloat преобразует строку, полученную из БД в структуру Point

func (points \*[Point](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Point)) StrToFloat(str [string](http://localhost:6060/pkg/builtin/#string))

### 2.4.18 Privilege

Privilege структура для запросов к БД

type Privilege struct {

Role [Role](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Role) `json:"role"` //информация о роли пользователя

Region [string](http://localhost:6060/pkg/builtin/#string) `json:"region"` //регион пользователя

Area [][string](http://localhost:6060/pkg/builtin/#string) `json:"area"` //массив районов пользователя

PrivilegeStr [string](http://localhost:6060/pkg/builtin/#string) `json:"-"` //строка для декодирования

}

Методы:

1. ConvertToJson из строки в структуру

func (privilege \*[Privilege](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Privilege)) ConvertToJson() (err [error](http://localhost:6060/pkg/builtin/#error))

1. DisplayInfoForAdmin отображение информации о пользователях для администраторов

func (privilege \*[Privilege](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Privilege)) DisplayInfoForAdmin(mapContx map[[string](http://localhost:6060/pkg/builtin/#string)][string](http://localhost:6060/pkg/builtin/#string)) map[[string](http://localhost:6060/pkg/builtin/#string)]interface{}

1. ReadFromBD прочитать данные из бд и разобрать

func (privilege \*[Privilege](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Privilege)) ReadFromBD(login [string](http://localhost:6060/pkg/builtin/#string)) [error](http://localhost:6060/pkg/builtin/#error)

1. ToSqlStrUpdate запись привилегий в базу

func (privilege \*[Privilege](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Privilege)) WriteRoleInBD(login [string](http://localhost:6060/pkg/builtin/#string)) (err [error](http://localhost:6060/pkg/builtin/#error))

### 2.4.19 RegionInfo

RegionInfo расшифровка региона

type RegionInfo struct {

Num [string](http://localhost:6060/pkg/builtin/#string) `json:"num"` //уникальный номер региона

NameRegion [string](http://localhost:6060/pkg/builtin/#string) `json:"nameRegion"` //расшифровка номера

}

Методы:

1. SetRegionInfo установить в структуру номер и имя региона по номеру

func (region \*[RegionInfo](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#RegionInfo)) SetRegionInfo(num [string](http://localhost:6060/pkg/builtin/#string))

### 2.4.20 Role

Role информация о роли

type Role struct {

Name [string](http://localhost:6060/pkg/builtin/#string) `json:"name"` //название роли

Perm [][int](http://localhost:6060/pkg/builtin/#int) `json:"permissions"` //массив полномочий

}

### 2.4.21 RoleAccess

RoleAccess информация наборах ролей и полномочий

type RoleAccess struct {

Roles [][Role](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Role) `json:"roles"` //массив ролей

Permission [][Permission](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Permission) `json:"permissions"` //массив разрешений

Routes [][RouteInfo](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#RouteInfo) `json:"routes"` //массив маршрутов

}

Методы:

1. ReadRoleAccessFile чтение RoleAccess файла

func (roleAccess \*[RoleAccess](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#RoleAccess)) ReadRoleAccessFile() (err [error](http://localhost:6060/pkg/builtin/#error))

### 2.4.22 RoleData

RoleData структура, включающая всю информацию о ролях, привелегиях, и маршрутах

type RoleData struct {

MapRoles map[[string](http://localhost:6060/pkg/builtin/#string)][][int](http://localhost:6060/pkg/builtin/#int) //роли

MapPermisson map[[int](http://localhost:6060/pkg/builtin/#int)][Permission](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Permission) //привилегии

MapRoutes map[[string](http://localhost:6060/pkg/builtin/#string)][RouteInfo](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#RouteInfo) //маршруты

}

### 2.4.23 RouteInfo

RouteInfo информация о всех расписанных маршрутах

type RouteInfo struct {

ID [int](http://localhost:6060/pkg/builtin/#int) `json:"id"` //уникальный номер маршрута

Permission [int](http://localhost:6060/pkg/builtin/#int) `json:"permission"` //номер разрешения к которому относится этот маршрут

Path [string](http://localhost:6060/pkg/builtin/#string) `json:"path"` //путь (url) обращения к ресурсу

Description [string](http://localhost:6060/pkg/builtin/#string) `json:"description"` //описание маршрута

}

### 2.4.24 SelectedData

SelectedData общая структура обмена

type SelectedData struct {

SelectedData map[[string](http://localhost:6060/pkg/builtin/#string)]map[[string](http://localhost:6060/pkg/builtin/#string)][][CheckData](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#CheckData) `json:"selected"` //хранилище перекрестков которые были выбраны

PngSettings [PngSettings](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#PngSettings) `json:"pngSettings"` //настройки для создания map.png

}

### 2.4.25 ServerLogInfo

ServerLogInfo данные для хранения информации лог файлов

type ServerLogInfo struct {

Type [string](http://localhost:6060/pkg/builtin/#string) `json:"type"` //тип лог сообщения

Time [string](http://localhost:6060/pkg/builtin/#string) `json:"time"` //врямя когда произошло событие

IP [string](http://localhost:6060/pkg/builtin/#string) `json:"IP"` //IP с которого делали запрос

Login [string](http://localhost:6060/pkg/builtin/#string) `json:"login"` //логин пользователя который делал запрос

Resource [string](http://localhost:6060/pkg/builtin/#string) `json:"resource"` //путь к ресурсу на котором произошло событие

Message [string](http://localhost:6060/pkg/builtin/#string) `json:"message"` //расшифровка действия пользователя

}

### 2.4.26 ShortAccount

ShortAccount удобная структура аккаунта для обмена с пользователем

type ShortAccount struct {

Login [string](http://localhost:6060/pkg/builtin/#string) `json:"login"` //логин пользователя

WorkTime [int](http://localhost:6060/pkg/builtin/#int) `json:"workTime"` //время сеанса пользователя

Password [string](http://localhost:6060/pkg/builtin/#string) `json:"password"` //пароль пользователя

Role [Role](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Role) `json:"role"` //роль пользователя

Privilege [string](http://localhost:6060/pkg/builtin/#string) `json:"-"` //привелегии (не уходят на верх)

Region [RegionInfo](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#RegionInfo) `json:"region"` //регион работы пользователя

Area [][AreaInfo](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#AreaInfo) `json:"area"` //районы работы пользователя

}

Методы:

1. ConvertShortToAcc преобразование информации об аккаунте

func (shortAcc \*[ShortAccount](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#ShortAccount)) ConvertShortToAcc() (account [Account](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Account), privilege [Privilege](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Privilege))

1. DecodeRequest расшифровываем json данные полученные от пользователя

func (shortAcc \*[ShortAccount](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#ShortAccount)) DecodeRequest(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) [error](http://localhost:6060/pkg/builtin/#error)

1. ValidChangePW проверка данных полученных от админа для смены паролей пользователя

func (shortAcc \*[ShortAccount](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#ShortAccount)) ValidChangePW(role [string](http://localhost:6060/pkg/builtin/#string), region [string](http://localhost:6060/pkg/builtin/#string)) (account \*[Account](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Account), err [error](http://localhost:6060/pkg/builtin/#error))

1. ValidCreate проверка данных полученных от пользователя на создание нового пользователя

func (shortAcc \*[ShortAccount](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#ShortAccount)) ValidCreate(role [string](http://localhost:6060/pkg/builtin/#string), region [string](http://localhost:6060/pkg/builtin/#string)) (err [error](http://localhost:6060/pkg/builtin/#error))

1. ValidDelete проверка данных полученных от пользователя на удаление аккаунта

func (shortAcc \*[ShortAccount](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#ShortAccount)) ValidDelete(role [string](http://localhost:6060/pkg/builtin/#string), region [string](http://localhost:6060/pkg/builtin/#string)) (account \*[Account](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Account), err [error](http://localhost:6060/pkg/builtin/#error))

### 2.4.27 TLSostInfo

TLSostInfo информация о состояния светофоров

type TLSostInfo struct {

Num [int](http://localhost:6060/pkg/builtin/#int) `json:"num"` //номер состояния

Description [string](http://localhost:6060/pkg/builtin/#string) `json:"description"` //описание состояния

}

### 2.4.28 Token

Token (JWT) структура токена доступа

type Token struct {

UserID [uint](http://localhost:6060/pkg/builtin/#uint) //Уникальный ID пользователя

Login [string](http://localhost:6060/pkg/builtin/#string) //Уникальный логин пользователя

IP [string](http://localhost:6060/pkg/builtin/#string) //IP пользователя

Role [string](http://localhost:6060/pkg/builtin/#string) //Роль

Permission [][int](http://localhost:6060/pkg/builtin/#int) //Привелегии

Region [string](http://localhost:6060/pkg/builtin/#string) //Регион пользователя

jwt.[StandardClaims](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#StandardClaims)

}

### 2.4.29 TrafficLights

TrafficLights информация о светофоре

type TrafficLights struct {

ID [int](http://localhost:6060/pkg/builtin/#int) `json:"ID"` //Уникальный ID светофора

Region [RegionInfo](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#RegionInfo) `json:"region"` //Регион

Area [AreaInfo](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#AreaInfo) `json:"area"` //Район

Subarea [int](http://localhost:6060/pkg/builtin/#int) `json:"subarea"` //ПодРайон

Idevice [int](http://localhost:6060/pkg/builtin/#int) `json:"idevice"` //Реальный номер устройства

Sost [TLSostInfo](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#TLSostInfo) `json:"tlsost"` //Состояние светофора

Description [string](http://localhost:6060/pkg/builtin/#string) `json:"description"` //Описание светофора

Points [Point](http://localhost:6060/pkg/github.com/JanFant/TLServer/data/#Point) `json:"points"` //Координата где находится светофор

}

### 2.4.30 LogFile

LogFile информация о лог файле

type LogFile struct {

flog \*[os](http://localhost:6060/pkg/os/).[File](http://localhost:6060/pkg/os/#File) //дескриптор открытого файла

path [string](http://localhost:6060/pkg/builtin/#string) //путь да текущего файла лога

date [string](http://localhost:6060/pkg/builtin/#string) //текущая дата

}

Методы:

1. Close закрытие лог файла

func (l \*[LogFile](http://localhost:6060/pkg/github.com/JanFant/TLServer/logger/#LogFile)) Close() [error](http://localhost:6060/pkg/builtin/#error)

1. Read чтение лог файла

func (l \*[LogFile](http://localhost:6060/pkg/github.com/JanFant/TLServer/logger/#LogFile)) Read(p [][byte](http://localhost:6060/pkg/builtin/#byte)) (n [int](http://localhost:6060/pkg/builtin/#int), err [error](http://localhost:6060/pkg/builtin/#error))

1. Write записать лог файла

func (l \*[LogFile](http://localhost:6060/pkg/github.com/JanFant/TLServer/logger/#LogFile)) Write(p [][byte](http://localhost:6060/pkg/builtin/#byte)) (n [int](http://localhost:6060/pkg/builtin/#int), err [error](http://localhost:6060/pkg/builtin/#error))

### 2.4.31 StateResult

StateResult информация о прорке данных

type StateResult struct {

SumResult [][string](http://localhost:6060/pkg/builtin/#string) //накоплении информации о ходе проверки

Err [error](http://localhost:6060/pkg/builtin/#error) //признак ошибки при проверке

}

Методы:

1. CtrlVerified проверка недельных карт

func CtrlVerified(cross \*[agS\_pudge](http://localhost:6060/pkg/github.com/ruraomsk/ag-server/pudge/).[Cross](http://localhost:6060/pkg/github.com/ruraomsk/ag-server/pudge/#Cross)) (result [StateResult](http://localhost:6060/pkg/github.com/JanFant/TLServer/stateVerified/#StateResult))

1. DaySetsVerified проверка суточных карт

func DaySetsVerified(cross \*[agS\_pudge](http://localhost:6060/pkg/github.com/ruraomsk/ag-server/pudge/).[Cross](http://localhost:6060/pkg/github.com/ruraomsk/ag-server/pudge/#Cross)) (result [StateResult](http://localhost:6060/pkg/github.com/JanFant/TLServer/stateVerified/#StateResult))

1. MouthSetsVerified проверка месячных(годовых) карт

func MouthSetsVerified(cross \*[agS\_pudge](http://localhost:6060/pkg/github.com/ruraomsk/ag-server/pudge/).[Cross](http://localhost:6060/pkg/github.com/ruraomsk/ag-server/pudge/#Cross), empty [IsEmpty](http://localhost:6060/pkg/github.com/JanFant/TLServer/stateVerified/#IsEmpty)) (result [StateResult](http://localhost:6060/pkg/github.com/JanFant/TLServer/stateVerified/#StateResult))

1. TimeUseVerified проверка недельных карт

func TimeUseVerified(cross \*[agS\_pudge](http://localhost:6060/pkg/github.com/ruraomsk/ag-server/pudge/).[Cross](http://localhost:6060/pkg/github.com/ruraomsk/ag-server/pudge/#Cross)) (result [StateResult](http://localhost:6060/pkg/github.com/JanFant/TLServer/stateVerified/#StateResult))

### 2.4.32 ArmCommandMessage

ArmCommandMessage ARM информация для отправки на сервер

type ArmCommandMessage struct {

User [string](http://localhost:6060/pkg/builtin/#string) //пользователь отправляющий данные (логин)

CommandStr [string](http://localhost:6060/pkg/builtin/#string) //данные подготовленные к отправке

Message [string](http://localhost:6060/pkg/builtin/#string) //информация о результате передачи данных

}

### 2.4.33 StateMessage

StateMessage state информация для отправки на сервер

type StateMessage struct {

User [string](http://localhost:6060/pkg/builtin/#string) //пользователь отправляющий данные (логин)

Info [string](http://localhost:6060/pkg/builtin/#string) //короткая информация о state

StateStr [string](http://localhost:6060/pkg/builtin/#string) //данные подготовленные к отправке

Message [string](http://localhost:6060/pkg/builtin/#string) //информация о результате передачи данных

}

### 2.4.34 TCPConfig

TCPConfig настройки для тсп соединения

type TCPConfig struct {

ServerAddr [string](http://localhost:6060/pkg/builtin/#string) `toml:"tcpServerAddress"` //адресс сервера

PortState [string](http://localhost:6060/pkg/builtin/#string) `toml:"portState"` //порт для обмена Стате

PortArmComm [string](http://localhost:6060/pkg/builtin/#string) `toml:"portArmCommand"` //порт для обмена арм командами

}

## 2.5 Утилиты

1) Message создает map для ответа пользователю

func Message(status [bool](http://localhost:6060/pkg/builtin/#bool), message [string](http://localhost:6060/pkg/builtin/#string)) map[[string](http://localhost:6060/pkg/builtin/#string)]interface{}

1. ParserInterface разбирает рефлексией интерфейс в map[string]string

func ParserInterface(in interface{}) (contx map[[string](http://localhost:6060/pkg/builtin/#string)][string](http://localhost:6060/pkg/builtin/#string))

3) Respond формирует ответ пользователю записывает необходимые хедеры и сворачивает json

func Respond(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request), data map[[string](http://localhost:6060/pkg/builtin/#string)]interface{})

4) WriteLogMessage обработчик u.message преобразует сообщение для записи в лог файл

func WriteLogMessage(ip [string](http://localhost:6060/pkg/builtin/#string), url [string](http://localhost:6060/pkg/builtin/#string), data map[[string](http://localhost:6060/pkg/builtin/#string)]interface{}, info interface{})

## 2.6 Middleware route

Middleware route – промежуточные фильтры, которые работают перед обработчиком запроса.

1) AccessControl проверка разрешен ли пользователя доступ к запрашиваемому ресурсу

var AccessControl = func(next [http](http://localhost:6060/pkg/net/http/).[Handler](http://localhost:6060/pkg/net/http/#Handler)) [http](http://localhost:6060/pkg/net/http/).[Handler](http://localhost:6060/pkg/net/http/#Handler){}

2) JwtAuth контроль токена для всех прошедших регистрацию и обрашающихся к ресурсу

var JwtAuth = func(next [http](http://localhost:6060/pkg/net/http/).[Handler](http://localhost:6060/pkg/net/http/#Handler)) [http](http://localhost:6060/pkg/net/http/).[Handler](http://localhost:6060/pkg/net/http/#Handler) {}

1. JwtFile упрошенных контроль токена для получения данных из файлового хранилища

var JwtFile = func(next [http](http://localhost:6060/pkg/net/http/).[Handler](http://localhost:6060/pkg/net/http/#Handler)) [http](http://localhost:6060/pkg/net/http/).[Handler](http://localhost:6060/pkg/net/http/#Handler) {}

2.7 Обработчики

### 2.7.1 ActAddAccount

ActAddAccount обработчик запроса добавления (работа с пользователями).

var ActAddAccount = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /manage/add

### 2.7.2 ActChangePw

ActChangePw обработчик запроса смены пароля для пользователя

var ActChangePw = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /manage/changepw

### 2.7.3 ActDeleteAccount

ActDeleteAccount обработчик запроса удаления (работа с пользователями)

var ActDeleteAccount = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /manage/delete

### 2.7.4 ActUpdateAccount

ActUpdateAccount обработчик запроса обновления (работа с пользователями)

var ActUpdateAccount = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /manage/update

### 2.7.5 BuildCross

BuildCross обработчик собора данных для отображения перекрёстка

var BuildCross = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /cross

### 2.7.6 BuildMapPage

BuildMapPage собираем данные для авторизованного пользователя

var BuildMapPage = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /map

### 2.7.7 CheckAllCross

CheckAllCross обработчик проверки всех перекрестков из БД

var CheckAllCross = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /manage/crossCreator/checkAllCross

### 2.7.8 CheckSelectedDirCross

CheckSelectedDirCross обработчик проверки регионов, районов и перекрестков, выбранных пользователем

var CheckSelectedDirCross = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)){}

Маршрут: /manage/crossCreator/checkSelected

### 2.7.9 ControlCheckButton

ControlCheckButton обработчик данных для их проверки

var ControlCheckButton = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /cross/control/checkButton

### 2.7.10 ControlCloseCross

ControlCloseCross обработчик закрытия перекрестка

var ControlCloseCross = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /cross/control/close

### 2.7.11 ControlCreateButton

ControlCreateButton обработчик данных для создания перекрестка и отправка на устройство(сервер)

var ControlCreateButton = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /cross/control/createButton

### 2.7.12 ControlCross

ControlCross обработчик данных для заполнения таблиц управления

var ControlCross = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /cross/control

### 2.7.13 ControlDeleteButton

ControlDeleteButton обработчик данных для удаления перекрестка

var ControlDeleteButton = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /cross/control/deleteButton

### 2.7.14 ControlEditableCross

ControlEditableCross обработчик проверки редактирования перекрестка

var ControlEditableCross = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /cross/control/editable

### 2.7.15 ControlSendButton

ControlSendButton обработчик данных для отправки на устройство(сервер)

var ControlSendButton = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /cross/control/sendButton

### 2.7.16 ControlTestState

ControlTestState обработчик проверки State

var ControlTestState = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /manage/stateTest

### 2.7.17 CrossEditFree

CrossEditFree освобождение перекрестков

var CrossEditFree = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /manage/crossEditControl/free

### 2.7.18 CrossEditInfo

CrossEditInfo сбор информации о занятых перекрестках

var CrossEditInfo = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /manage/crossEditControl

### 2.7.19 DevCrossInfo

DevCrossInfo обработчик собора данных для отображения перекрёстка (idevice информация)

var DevCrossInfo = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /cross/dev

### 2.7.20 DispatchControlButtons

DispatchControlButtons обработчик кнопок диспетчерского управления

var DispatchControlButtons = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)){}

Маршрут: /cross/DispatchControlButtons

### 2.7.21 DisplayAccInfo

DisplayAccInfo отображение информации об аккаунтах для администрирования

var DisplayAccInfo = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /manage

### 2.7.22 DisplayDeviceLogFile

DisplayDeviceLogFile обработчик отображения файлов лога устройства

var DisplayDeviceLogFile = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)){}

Маршрут: /map/deviceLog

### 2.7.23 DisplayServerLogFile

DisplayServerLogFile обработчик отображения файлов лога сервера

var DisplayServerLogFile = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)){}

Маршрут: /manage/serverLog

### 2.7.24 DisplayServerLogInfo

DisplayServerLogInfo обработчик выгрузки содержимого лог файла сервера

var DisplayServerLogInfo = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)){}

Маршрут: /manage/serverLog/info

### 2.7.25 LocationButtonMapPage

LocationButtonMapPage обработка запроса на получение новых координат рабочей области

var LocationButtonMapPage = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /map/locationButton

### 2.7.26 LogDeviceInfo

LogDeviceInfo обработчик запроса на выгрузку информации логов устройства за определенный период

var LogDeviceInfo = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /map/deviceLog/info

### 2.7.27 LoginAcc

LoginAcc обработчик входа в систему

var LoginAcc = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /login

### 2.7.28 LoginAccOut

LoginAccOut обработчик выхода из системы

var LoginAccOut = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /map/logOut

### 2.7.29 MainCrossCreator

MainCrossCreator сборка информации для странички создания каталогов перекрестков

var MainCrossCreator = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /manage/crossCreator

### 2.7.30 MakeSelectedDirCross

MakeSelectedDirCross обработчик проверки регионов, районов и перекрестков, выбранных пользователем

var MakeSelectedDirCross = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)){}

Маршрут: /manage/crossCreator/makeSelected

### 2.7.31 UpdateMapPage

UpdateMapPage обновление информации о попавших в область светофорах

var UpdateMapPage = func(w [http](http://localhost:6060/pkg/net/http/).[ResponseWriter](http://localhost:6060/pkg/net/http/#ResponseWriter), r \*[http](http://localhost:6060/pkg/net/http/).[Request](http://localhost:6060/pkg/net/http/#Request)) {}

Маршрут: /map/update