**ПО «Интерсвязь. Цифровая мультисервисная платформа - суперприложение для жителей. Модуль “Личный кабинет”»**

Инструкция по установке

Экземпляр ПО доступен по ссылке:

https://domofon.io/platform\_superapp/lk\_docs

По данной ссылки можно найти:

· Документацию по эксплуатации ПО;

· Инструкцию по развертыванию ПО;

· Жизненный цикл СП;

· Архив с содержанием серверного ПО;

Для проверки работы модуля “Личный кабинет”, являющегося модулем ПО «Интерсвязь. Цифровая мультисервисная платформа - суперприложение для жителей”, следует перейти по ссылке: <https://domofon.io/platform_superapp/lk_docs> и скачать архив с файлами под названием: core\_module\_lk.zip

# 1 Установка системы

## 1.1 Системные требования для работы серверного ПО:

Современная операционная система Linux, имеющая интерпретатор командной строки, рекомендуется «Debian 10».

* Docker 20.10.12+
* Docker-compose 2.2.2+
* Golang 1.17.5
* Аппаратные требования:

1. Оперативная память: 8 ГБ
2. Свободное дисковое пространство: не менее 100 ГБ (зависит от объема данных)
3. Количество логических ядер процессора: 4
4. Частота процессора: 2.10 GHz

## 1.2 Используемые компоненты и языки

| Название | Лицензия | Назначение |
| --- | --- | --- |
| Docker | Apache 2.0 | Программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации, контейнеризатор приложений. |
| Golang | BSD | Для написания серверных модулей, которые получают, обрабатывают и отправляют информацию. |
| Redis | BSD | Используется как быстрое хранилище данных типа «ключ‑значение». |
| PostgreSQL | BSD | База данных для хранения долгоживущей информации. |
| Javascript | LGPL v2.1 | Используется как встраиваемый язык для программного доступа к объектам приложений. |
| Css | MIT | Формальный язык описания внешнего вида документа. |

## 1.3 Установка

### 1.3.1 Установка серверного ПО

Для установки программного обеспечения на собственные сервера требуется произвести следующие действия:

Загрузить архив “core\_module\_lk.zip” на сервер (в желаемую папку на усмотрение пользователя/администратора, осуществляющего установку экземпляра ПО), на котором предполагается развернуть модуль “личный кабинет”, после чего разархивировать архив перейти в новый каталог.

Перейти в директорию с загруженными архивами и выполнить следующие команды:

1. unzip core\_module\_lk.zip
2. cd core\_module\_lk
3. cd core-mb
4. docker-compose up -d --force-recreate
5. cd ../core-adapter-mini-billing/
6. go run mini\_biling/cmd/adapter

После чего модуль будет установлен и запущен.

## **2 Руководство пользователя**

Система представлена в виде модуля “личный кабинет”, являющимся модулем платформы Интерсвязь.Цифровая мультисервисная платформа – суперприложение для жителей супер приложение для жителей и взаимодействует с ней посредством api. Для демонстрации работы системы ниже представлена инструкция по взаимодействию с ней.

Для удобства тестирования в архиве “core\_module\_lk.zip” есть файл “core mb simple.postman\_collection.json” с коллекцией запросов postman.

### Создание пользователя

Для начала работы с системой необходимо создать в ней пользователей. Создание пользователей осуществляется с помощью POST запроса **http://<host>:8093/api/v1/mb/user/extend**. В теле запроса передается:

* user - информация о пользователе
  + ext\_id - внешний уникальный номер пользователя
  + login - логин
  + password - пароль
  + full\_name - ФИО
* phones - список номеров телефонов
  + confirmed - статус телефона
  + number - номер телефона

Пример тела запроса:

{

"user" : {"ext\_id" : "198708", "login":"test\_new\_user", "password":"1234", "full\_name" :"Иван Иванов"},

"phones" :[{"confirmed": **true**, "number":"9000000000"}]

}

### Создание сервиса

Для интеграции данных пользователя с сторонней биллинг системой в модуле “Личный кабинет) необходимо создать структуру связей между сервисом, тарифами и договорами.

Для создания сервиса необходимо воспользоваться POST запросом **http://<host>:8093/api/v1/mb/tariff/service/add**. В теле запроса передается:

* Name - название сервиса
* ext\_id - внешний уникальный номер сервиса

Пример тела запроса:

{

"Name" : "Домофония",

"ext\_id" :"123"

}

### Создание тарифа

POST запрос для создания тарифа **http://<host>:8093/api/v1/mb/tariff/add**. В теле запроса передается:

* name - название тарифа
* description - описание тарифа
* ext\_id - внешний уникальный номер тарифа
* base\_amount - сумма тарифа
* connect\_url - url на описание тарифа
* services - список сервисов

Пример тела запроса:

{

"name" : "Улыбка",

"description": "Описание тарифа",

"ext\_id" :"1234",

"base\_amount": 4000,

"connect\_url": "ya.ru",

"services" : [45, 44]

}

### **Создание договора**

Договор является центральной сущностью для управления связями с внешней биллинг системой. POST запрос для создания договора **http://<host>:8093/api/v1/mb/tariff/contract/add**. В теле запроса передается:

* account - лицевой счет договора
* contract - номер договора
* balance - баланс

Пример тела запроса:

{

"account" :"new c" ,

"contract" :"contract" ,

"balance" : 12000

}

### Связка договора и тарифа

Для корректного отображения информации в модуле (личный кабинет) необходимо связать договоры с тарифами. Для установления связи используется POST запрос **http://<host>:8093/api/v1/mb/contract/extend/<uuid договора>/tariffs**. В теле запроса передается:

* id - уникальный номер тарифа

Пример тела запроса:

{

"id" : 503

}

### Связка договора и пользователя

Далее нужно назначить договор определенному пользователю. Таким образом будет установлена целостность всех данных и зафиксированы связи между ними. Такая структура позволяет гибко управлять информацией о пользователе.

POST запрос для создания связки между договором и пользователем **http://<host>:8093/api/v1/mb/contract/relations**.

В теле запроса передается:

* user\_uuid - внутренний уникальный идентификатор пользователя
* contract\_uuid - внутренний уникальный идентификатор договора
* role\_id - номер роли

Пример тела запроса:

{

"user\_uuid": "d3afa023-a588-4bfe-ab68-9eb426daa5f4",

"contract\_uuid": "6942fe21-276b-4d0c-8ff7-018b0c7276a8",

"role\_id": 1

}

### Получить информацию о всех пользователях организации

Для просмотра всей информации по пользователям GET запроса **http://<host>:8093/api/v2/mb/user/organization**.

### Получить информации пользователя

Для просмотра всей информации о пользователе нужно воспользоваться GET запроса **http://<host>:8093/api/v1/mb/user/<uuid пользователя>**.

### Авторизация

Все созданные пользователи могут проходить авторизацию по логину и паролю. Для прохождения авторизации необходимо воспользоваться POST запросом **http://<host>:8093/api/v1/mb/auth/login**.

В теле запроса передается:

* login - логин
* password - пароль
* org\_id - уникальный номер организации

Пример тела запроса:

{

"login" :"test\_new\_user",

"password" :"1234",

"org\_id" :55

}