

# Инструкция по установке экземпляра программного обеспечения Smarty

## Содержание

Введение .....	3
Получение демо-доступа к облачной Smarty и порталу .....	3
Установка Smarty на локальный сервер .....	4
Установка необходимых пакетов и библиотек.....	4
Установка СУБД для работы Smarty .....	5
Подготовка конфигурации Web-сервера nginx и сервера приложений uWSGI .....	5
Установка лицензионного ключа.....	6
Установка схемы БД.....	6
Запуск сервисов .....	6
Установка системных настроек и примера конфигурации .....	6
Настройка регулярных команд в crontab .....	7
Создание пользователя и начало работы .....	7
Установка портала для STB и Smart TV на локальный сервер .....	8
Установка Web-сервера nginx .....	8
Установка пакетов портала.....	8
Подготовка конфигурации Web-сервера nginx.....	9
Подключение портала к Smarty.....	9
Контакты .....	11

## Введение

Smarty состоит из двух компонентов:

- Smarty — сервер управления и API.
- Абонентский портал (Web-приложение для Linux и Android-приставок и Smart TV). Взаимодействует с сервером Smarty через API.

Установочные пакеты Smarty и портала можно скачать на нашем FTP-сервере, доступ предоставляется клиентам после заключения договора.

Далее приведена инструкция по получению демо-доступа к облачной Smarty и portalу, а также инструкция по локальной установке.

## Получение демо-доступа к облачной Smarty и portalу

Для ускорения процесса тестирования можно воспользоваться уже развернутыми в облаке Smarty и порталом.

Веб-панель администрирования:

<http://smarty.microimpuls.com/>

Логин и пароль указан в описании к данному документу.

Получить доступ к клиентскому приложению можно по инструкции ниже:

1. Установить приложение **Плюс ТВ** на любое доступное устройство по инструкции: <https://micro.im/docs/smarty-lite/how-to-install-plustv>
2. Выбрать в списке операторов на первой странице провайдера **Plus TV**
3. Авторизоваться учетной записью **1119:9991**

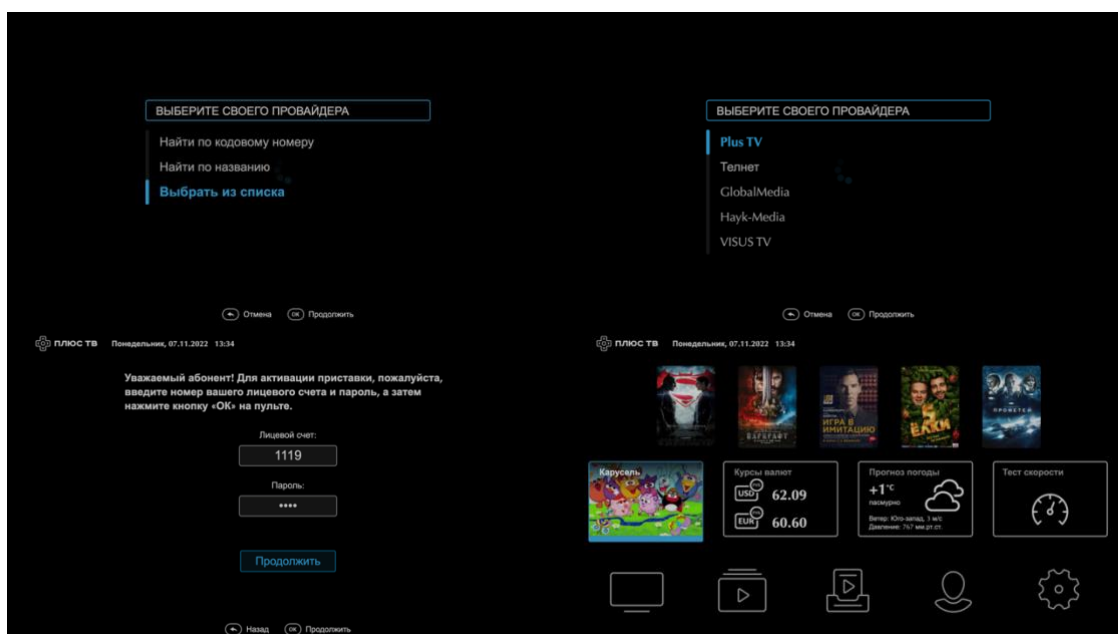


Рисунок 1 – Алгоритм входа в приложение Плюс ТВ

Данный способ тестирования позволит ознакомиться с приложением на всех доступных устройствах.

## Установка Smarty на локальный сервер

Ниже приведена инструкция по установке Smarty на сервер с ОС Debian Buster. Информацию о настройке тех или иных функций или сервисов платформы Smarty Middleware, которую вы не найдёте здесь, вам следует искать на странице помощи полной версии Smarty: <https://micro.im/docs/smarty/about>

Для работы потребуется SSH доступ к серверу с правами root. Установка будет производиться в консольном режиме через команды терминала. Если для получения привилегий root требуется выполнить команду sudo, то перед началом установки необходимо выполнить команду:

```
sudo su
```

### Установка необходимых пакетов и библиотек

Устанавливаем зависимости с помощью apt:

```
apt update
```

```
apt install git python-dev libtiff-dev libjpeg-dev zlib1g-dev libfreetype6-dev liblcms2-dev libwebp-dev python-tk uwsgi uwsgi-plugin-python redis-server build-essential wget pwgen libmariadb-dev nginx
```

Далее устанавливаем утилиту pip, она потребуется для установки используемых python-библиотек:

```
wget https://bootstrap.pypa.io/pip/2.7/get-pip.py && python get-pip.py && rm get-pip.py
```

Далее необходимо скачать установочные deb-пакеты smarty-lite и python2.7-jsonrpctcp и установить их:

```
dpkg -i smarty-lite*.deb
```

```
dpkg -i python2.7-jsonrpctcp_*.deb
```

Будьте внимательны — необходимо использовать установочный пакет Smarty именно для версии Debian Buster, он располагается в директории buster. В противном случае при выполнении дальнейших команд может возникнуть ошибка undefined symbol: PyFPE\_jbuf. При возникновении ошибки

```
dpkg: warning: 'ldconfig' not found in PATH or not executable
```

```
dpkg: warning: 'start-stop-daemon' not found in PATH or not executable
```

выполните следующую команду:

```
export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

После этого попробуйте установить пакеты еще раз.

После установки основных пакетов необходимо выполнить установку python-библиотек, используемых в Smarty. Для этого сначала нужно выполнить (только для Debian Buster):  
cp /usr/share/microimpuls/smarty-lite/requirements.txt ./temp.txt && cat ./temp.txt | grep -vE '^[Mm]y[Ss][Qq][Ll]-python=.\*' > /usr/share/microimpuls/smarty-lite/requirements.txt && rm ./temp.txt

```
pip install mysql-connector
```

```
pip install mysql-connector-python
```

```
pip install mysqlclient
```

Затем установить python-библиотеки:

```
pip install -r /usr/share/microimpuls/smarty-lite/requirements.txt
```

## Установка СУБД для работы Smarty

Smarty поддерживает различные СУБД, в базовом варианте рекомендуется использовать MySQL или MariaDB, для Debian Buster устанавливаем MariaDB:

```
apt install mariadb-server
```

Создаём пользователя для подключения Smarty в БД и указываем пароль в настройках Smarty:

```
export DBPASSWORD=`pwgen -s 16`
```

```
echo "CREATE DATABASE smarty CHARACTER SET = utf8;" | mysql -u root
```

```
echo "SET default_storage_engine=InnoDB;" | mysql -u root
```

```
echo "CREATE USER 'smarty'@'%' IDENTIFIED BY '$DBPASSWORD';" | mysql -u root
```

```
echo "GRANT ALL PRIVILEGES ON smarty.* TO 'smarty'@'%';" | mysql -u root
```

```
echo "FLUSH PRIVILEGES;" | mysql -u root
```

```
sed -i -e 's/PUT DB PASSWORD HERE/"$DBPASSWORD"/g' /etc/microimpuls/smarty-lite/smarty-lite.py
```

Последняя команда заменяет строку «PUT DB PASSWORD HERE» на сгенерированный автоматически пароль в файле `/etc/microimpuls/smarty-lite/smarty-lite.py` — файл настроек Smarty.

В файле настроек Smarty по умолчанию указаны настройки подключения к локально установленной СУБД MySQL (или совместимой с ней MariaDB), однако при необходимости можно установить СУБД на другую машину.

## Подготовка конфигурации Web-сервера nginx и сервера приложений uWSGI

Прописываем секретный ключ для работы внутренних механизмов защиты Smarty:

```
export SECRETKEY=`pwgen -s 32`
```

```
sed -i -e 's/PUT RANDOM SECRET KEY HERE/"$SECRETKEY"/g' /etc/microimpuls/smarty-lite/smarty-lite.py
```

Активируем конфигурацию Smarty для nginx и uwsgi:

```
ln -s /etc/nginx/sites-available/smarty-lite /etc/nginx/sites-enabled/smarty-lite
```

```
ln -s /etc/uwsgi/apps-available/smarty-lite.ini /etc/uwsgi/apps-enabled/smarty-lite.ini
```

Создаем директорию для кеша и логов nginx:

```
mkdir /var/cache/nginx
```

```
mkdir -p /var/log/nginx/microimpuls/smarty-lite
```

Изменяем лимит открытых сокетов для оптимальной работы uwsgi:

```
echo "net.core.somaxconn=4096" >> /etc/sysctl.conf
```

```
sysctl -p
```

Установим правильные права на директории для логов и загрузки файлов в Smarty:

```
chown -R www-data:www-data /var/log/microimpuls
```

```
chown -R www-data:www-data /usr/share/microimpuls
```

## Установка лицензионного ключа

Дальше необходимо прописать лицензионный ключ.

Для получения лицензионного ключа вам необходимо использовать HW key — отпечаток машины, на котором установлена Smarty. Для его получения необходимо запустить uwsgi: `/etc/init.d/uwsgi restart`

А затем в файле `/var/log/uwsgi/app/smarty-lite.log` найти значение HW key, или воспользоваться командой:

```
cat /var/log/uwsgi/app/smarty-lite.log | grep "Exception: Key incorrect, HW key"
```

Например, полученный HW key будет иметь следующий вид:

```
9eb4a0c2b360c93e32cf2471780cda494705f0a9d026c69fe033d8c8
```

Это значение необходимо вставить в соответствующее поле в окне генерации лицензии в личном кабинете оператора, после чего скопировать получившийся ключ со всеми параметрами лицензии, которые необходимо прописать в файл

конфигурации `/etc/microimpuls/smarty-lite/smarty-lite.py`.

Генерация лицензии вызывается по кнопке «Сгенерировать новый лицензионный ключ» в соответствующем продукте в ЛК.

Копирование лицензии и всех параметров производится с помощью иконки «Скопировать» в строке с лицензионным ключом.

## Установка схемы БД

Выполним установку схемы базы данных Smarty с помощью встроенного механизма миграций данных:

```
python /usr/share/microimpuls/smarty-lite/manage.py migrate --settings=settings.smarty-lite
```

## Запуск сервисов

Перезапустим nginx и uwsgi с актуальными настройками:

```
/etc/init.d/nginx restart
```

```
/etc/init.d/uwsgi restart
```

## Установка системных настроек и примера конфигурации

Для быстрого старта использования Smarty мы подготовили команду, которая установит системные настройки и создаст примеры данных — наборы телеканалов, EPG, тарифы, стриминг-сервисы и другие объекты, необходимые для работы сервиса IPTV & OTT. Для этого выполните:

```
python /usr/share/microimpuls/smarty-lite/manage.py setup_initial_data --settings=settings.s
smarty-lite
```

Затем вы сможете отредактировать их и добавить свои данные.

## Настройка регулярных команд в crontab

Для корректной работы Smarty и актуализации данных абонентов требуется настроить в crontab выполнение команд по расписанию, для этого в конце файла `/etc/crontab` добавьте следующие строки:

```
*/1 * * * * root python /usr/share/microimpuls/smarty-lite/manage.py cache_channel_list --se
ttings=settings.smarty-lite
```

```
0 5,9,13 * * * root python /usr/share/microimpuls/smarty-lite/manage.py epg_import --setting
s=settings.smarty-lite
```

```
0 3 * * * root python /usr/share/microimpuls/smarty-lite/manage.py clean_old_messages --day
s_count 3 --settings=settings.smarty-lite
```

Первая команда обновляет кеш телеканалов. Вторая команда импортирует EPG (телегид) каждые сутки 3 раза — в 5:00, 9:00 и 13:00 по времени сервера. Третья команда очищает старые недоставленные до абонентов текстовые сообщения и команды (с датой создания старше трех дней).

## Создание пользователя и начало работы

Управление сервисом IPTV & OTT в Smarty осуществляется через Web-панель управления.

Далее создадим пользователя с правами суперадминистратора для доступа в панель управления:

```
export ADMINPASSWORD=`pwgen -s 16`
```

```
python /usr/share/microimpuls/smarty-lite/manage.py create_user --settings=settings.smarty-l
ite --reset_password=False --username=admin --password=`echo $ADMINPASSWORD` --is_admin=True --client_id=1 --is_superuser=True
```

```
echo $ADMINPASSWORD
```

Последняя команда выведет сгенерированный пароль — сохраните его.

Затем открываем в браузере панель управления Smarty `http://X.X.X.X:8180` и авторизуемся созданным пользователем (в нашем примере это `admin` и пароль, сгенерированный выше):

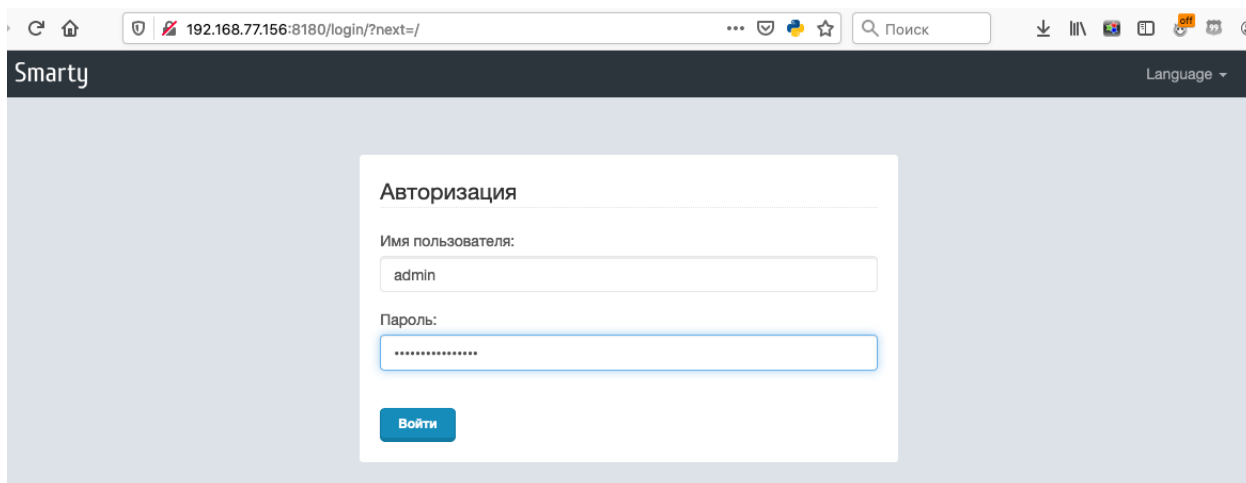


Рисунок 2 - Экран авторизации в панели управления Smarty

Далее необходимо установить и настроить абонентский портал, соединив его с API Smarty. Через портал абонент получает доступ к сервисам IPTV & OTT.

## Установка портала для STB и Smart TV на локальный сервер

Ниже приведена инструкция по установке портала для STB и Smart TV на сервер с ОС Debian, а также по настройке его взаимодействия с сервером Smarty. Информацию о настройке тех или иных возможностей и об изменении поведения портала вы можете найти здесь: <https://micro.im/docs/smarty/portal-and-apps-settings>

Для работы потребуется SSH доступ к серверу с правами root. Установка будет производиться в консольном режиме через команды терминала. Если для получения привилегий root требуется выполнить команду `sudo`, то перед началом установки необходимо выполнить команду:

```
sudo su
```

Предполагается, что портал устанавливается на тот же сервер, на котором уже установлен сервер Smarty, иначе на шаге настройки конфигурации `nginx` будут возникать ошибки из-за отсутствия настроек Smarty.

## Установка Web-сервера nginx

Портал это статический одностраничный сайт, состоящий из файлов `html/css/javascript`. Для хостинга портала можно использовать Web-сервер `nginx` с базовой конфигурацией для раздачи статики.

Устанавливаем зависимости с помощью `apt`:

```
apt update
```

```
apt install nginx-common nginx nginx-extras
```

## Установка пакетов портала

Портал состоит из двух видов установочных пакетов:

- `mw-engine-*.deb` — основной код портала и механизмы взаимодействия с устройством.



- `mw-template-*.deb` — шаблон интерфейса.
- Для работы портала необходимо обязательно установить и `mw-engine` и какой-либо `mw-template-*`, при этом можно установить несколько различных шаблонов интерфейса `mw-template-*`.

Про особенности и возможности различных интерфейсов портала можно узнать на этой странице: <https://micro.im/docs/smarty/portal-and-apps-settings>  
Установочные пакеты портала можно скачать в вашем личном кабинете оператора также, как и пакеты Smarty.

Далее установим пакеты портала, для примера будем настраивать шаблон `futuristic` и `impuls`:

```
dpkg -i mw-engine_*.deb
dpkg -i mw-template-*.deb
```

При возникновении ошибки

```
dpkg: warning: 'ldconfig' not found in PATH or not executable
dpkg: warning: 'start-stop-daemon' not found in PATH or not executable
```

выполните следующую команду:

```
export PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

После этого попробуйте установить пакеты еще раз.

## Подготовка конфигурации Web-сервера nginx

Активируем конфигурацию портала для nginx и отключим default-конфигурацию:

```
ln -s /etc/nginx/sites-available/portal /etc/nginx/sites-enabled/portal
```

```
rm /etc/nginx/sites-enabled/default
```

По умолчанию портал подключается к полной версии Smarty, поэтому в стандартной конфигурации используется имя `smarty`. Для Smarty нужно выполнить замену на `smarty-lite` с помощью следующей команды:

```
sed -i -e 's/smarty/smarty-lite/g' /etc/nginx/sites-available/portal
```

Создаем директорию для кеша и логов nginx:

```
mkdir -p /var/cache/nginx
```

```
mkdir -p /var/log/nginx/microimpuls/portal
```

Установим правильные права на директории для логов и загрузки файлов в Smarty:

```
chown -R www-data:www-data /usr/share/nginx/html/microimpuls
```

Перезапустим nginx с актуальными настройками:

```
/etc/init.d/nginx restart
```

## Подключение портала к Smarty

Портал взаимодействует с сервером Smarty с помощью API. Данные для подключения задаются в файле конфигурации портала `/etc/microimpuls/portal/client.js`.

После завершения установки Smarty и запуска команды `setup_initial_data` в Smarty создается первый оператор с идентификатором Client ID = 1 и генерируется случайное

значение API key. Это значение можно посмотреть в панели управления Smarty в разделе «Общие настройки» → «Общие настройки Client» в поле «TVMW API key»:

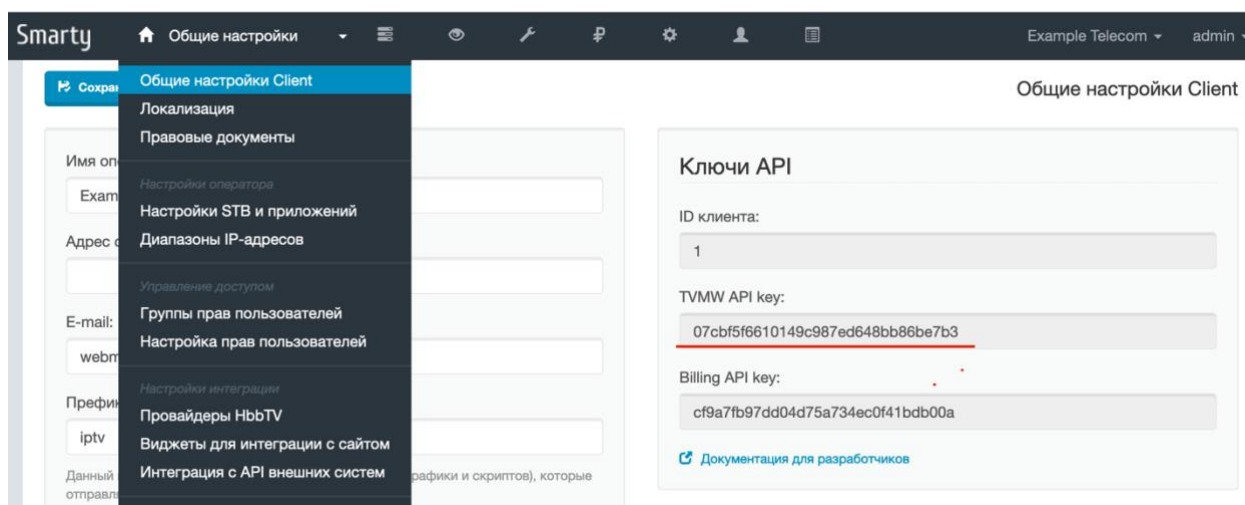


Рисунок 3 – Страница настроек в веб-панели администрирования Smarty

Скопируйте данное значение (в этом примере: **07cbf5f6610149c987ed648bb86be7b3**) и вставьте его в поле `api_key` в файле конфигурации портала

`/etc/microimpuls/portal/client.js:`

```
sed -i -e 's/PUT API KEY HERE/'07cbf5f6610149c987ed648bb86be7b3'/g' /etc/microimpuls/portal/client.js
```

В конфигурации `nginx` по умолчанию задан специальный `location /api`, который направляет запросы к API в `uwsgi`-сокете Smarty, поэтому стандартное значение параметра `api_url` можно не менять.

Более подробно об опциях подключения портала к Smarty и других настройках вы можете узнать на этой странице: <https://micro.im/docs/smarty/portal-and-apps-settings>  
После этого откройте портал в браузере для проверки правильности настройки, введя IP-адрес сервера (порт 80 по умолчанию). Если все сделано верно, то вы должны увидеть экран авторизации:

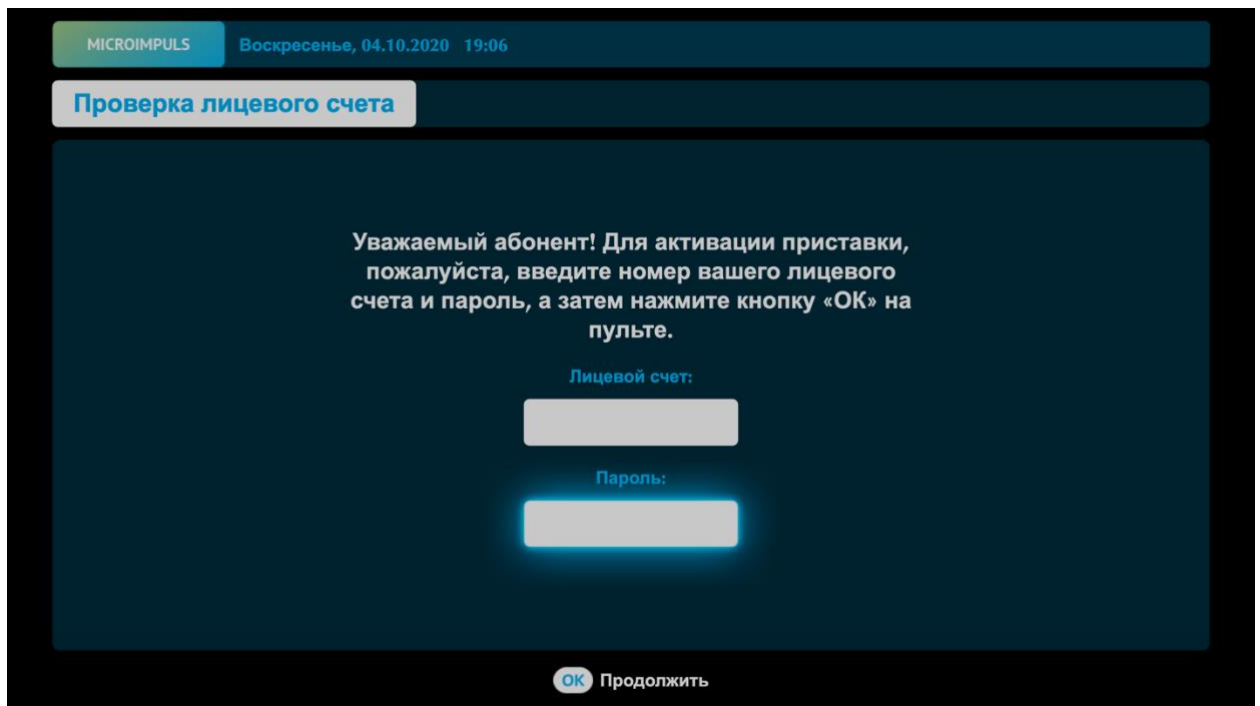


Рисунок 4 - Экран авторизации в клиентском портале (интерфейс Futuristic)

Для входа используйте логин и пароль демо-аккаунта 123 : 123.

Для открытия портала на ТВ-приставке укажите в настройках устройства тот же адрес портала, что и в браузере.

## Контакты

Круглосуточная техническая поддержка осуществляется по адресу [support@microimpuls.com](mailto:support@microimpuls.com)