

# Инструкция по установке экземпляра ПО «LAB\_Front. Мультибанковская платформа дистанционного обслуживания физических лиц»

Требования к системе для запуска системы «LAB\_Front. Мультибанковская платформа дистанционного обслуживания физических лиц»:

- Предпочтительно развёртывание на ОС Linux CentOS 7 (2003), которая доступна для загрузки по адресу <https://centos.org/download/> и распространяется под лицензией GNU General Public License v2;
- 2-4 ядра CPU;
- 10-16 GB оперативной памяти;
- 10 GB свободного места на диске;
- Доступ в Internet для скачивания Docker образов системы «LAB\_Front. Мультибанковская платформа дистанционного обслуживания физических лиц»;
- Docker версии 19.03.\*, который доступен (с описанием по установке) для загрузки по адресу <https://docs.docker.com/engine/install/centos/> и распространяется под лицензией Apache License 2.0;
- Docker-compose 1.27.\*, который доступен (с описанием по установке) для загрузки по адресу <https://docs.docker.com/compose/install/#install-compose-on-linux-systems> и распространяется под лицензией Apache License 2.0.

Требования к клиентской системе:

- Браузер для использования системы LAB\_Front. Мультибанковская платформа дистанционного обслуживания физических лиц. Рекомендуется использовать Firefox. Минимальная версия Firefox 78.0, рекомендуемая 82.0.2. В представленной демонстрационной системе для упрощения взаимодействия между сервисами используется протокол HTTP без шифрования, поэтому в браузерах Chrome и Opera могут быть ошибки при открытии страниц.
- «Мультибанковская платформа дистанционного обслуживания «LAB Front» является интернет-сервисом, доступ к которому осуществляется с помощью любого интернет-браузера. Установка дополнительного программного обеспечения на рабочем месте конечного пользователя:
  - Для клиентов ПК не требуется;
  - Для клиентов мобильных устройств – скачивается с Google Play/ Apple Store.

**Ссылка для загрузки экземпляра ПО:**

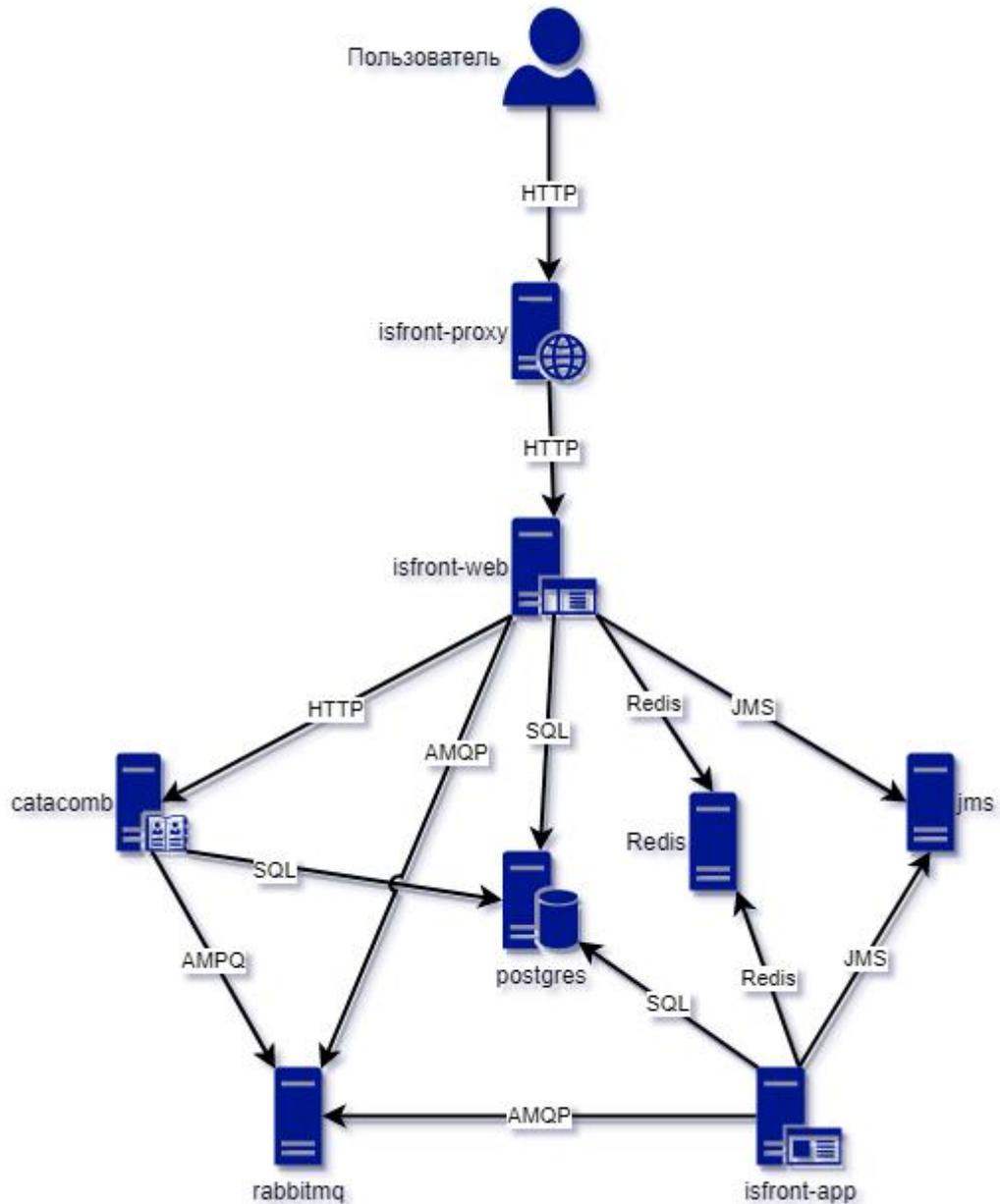
[https://registry.skblab.ru/labfront\\_raw/distribution-fl.tar.gz](https://registry.skblab.ru/labfront_raw/distribution-fl.tar.gz)

**Логин для загрузки экземпляра ПО:**

labfront\_r

**Пароль для загрузки экземпляра ПО:**

Общая схема работы компонентов системы:



Структура системы:

Контейнер	Назначение	Основн ое ПО	Лицензия	Репозиторий
isfront-web	Сервер приложений	Jetty 9.4.30	Apache License 2.0	<a href="https://github.com/bell-sw/Liberica/blob/master/docker/repos/liberica-openjdk-">https://github.com/bell-sw/Liberica/blob/master/docker/repos/liberica-openjdk-</a>

	для обработки клиентских запросов		Eclipse Public License 1.0	<a href="https://github.com/eclipse/jetty.project/releases/tag/jetty-9.4.30.v20200611">alpine/8/Dockerfile</a> <a href="https://github.com/eclipse/jetty.project/releases/tag/jetty-9.4.30.v20200611">https://github.com/eclipse/jetty.project/releases/tag/jetty-9.4.30.v20200611</a>
isfront-app	Сервер приложений для обмена с банковскими системами	Jetty 9.4.30	Apache License 2.0 Eclipse Public License 1.0	<a href="https://github.com/bell-sw/Liberica/blob/master/docker/repos/liberica-openjdk-alpine/8/Dockerfile">https://github.com/bell-sw/Liberica/blob/master/docker/repos/liberica-openjdk-alpine/8/Dockerfile</a> <a href="https://github.com/eclipse/jetty.project/releases/tag/jetty-9.4.30.v20200611">https://github.com/eclipse/jetty.project/releases/tag/jetty-9.4.30.v20200611</a>
catacomb	Сервер авторизации пользователей	Spring Boot v2.3.0 Spring v5.2.6	Apache License 2.0	<a href="https://github.com/AdoptOpenJDK/openjdk-docker/blob/master/11/jdk/alpine/Dockerfile.hotspot.releases.full">https://github.com/AdoptOpenJDK/openjdk-docker/blob/master/11/jdk/alpine/Dockerfile.hotspot.releases.full</a>
jms	Сервер очередей	OpenMQ 5.1.4	Eclipse Public License - v 2.0	<a href="https://github.com/bell-sw/Liberica/blob/master/docker/repos/liberica-openjdk-alpine/8/Dockerfile">https://github.com/bell-sw/Liberica/blob/master/docker/repos/liberica-openjdk-alpine/8/Dockerfile</a> <a href="https://github.com/eclipse-ee4j/openmq/tree/5.1.4-RELEASE">https://github.com/eclipse-ee4j/openmq/tree/5.1.4-RELEASE</a>
rabbitmq	Сервер очередей	RabbitMQ 3.8.9	Mozilla Public License Version 2.0	<a href="https://github.com/docker-library/rabbitmq/blob/888638927482f86af6e88bebb67423926cb1112f/3.8/ubuntu/management/Dockerfile">https://github.com/docker-library/rabbitmq/blob/888638927482f86af6e88bebb67423926cb1112f/3.8/ubuntu/management/Dockerfile</a> <a href="https://github.com/rabbitmq/rabbitmq-server">https://github.com/rabbitmq/rabbitmq-server</a> <a href="https://github.com/rabbitmq/rabbitmq-management">https://github.com/rabbitmq/rabbitmq-management</a>
postgres	Сервер баз данных	PostgreSQL 11.9	PostgreSQL Licence	<a href="https://github.com/docker-library/postgres/blob/bfc5d81c8f5647c690f452dc558e64fddb1802f6/11/Dockerfile">https://github.com/docker-library/postgres/blob/bfc5d81c8f5647c690f452dc558e64fddb1802f6/11/Dockerfile</a> <a href="https://github.com/postgres/postgres/tree/REL_11_9">https://github.com/postgres/postgres/tree/REL_11_9</a>
nginx	Web-сервер для отдачи статики и проксирования запросов к isfront-web	Nginx 1.19.2	2-clause BSD license	<a href="https://github.com/nginxinc/docker-nginx-unprivileged/blob/main/mainline/alpine/Dockerfile">https://github.com/nginxinc/docker-nginx-unprivileged/blob/main/mainline/alpine/Dockerfile</a> <a href="https://github.com/nginx/nginx">https://github.com/nginx/nginx</a>
redis-node-*	Кэш	Redis 6.0.8	three clause BSD license	<a href="https://github.com/bitnami/bitnami-docker-redis-cluster/tree/master/6.0/debian-10">https://github.com/bitnami/bitnami-docker-redis-cluster/tree/master/6.0/debian-10</a> <a href="https://github.com/redis/redis/tree/6.0.8">https://github.com/redis/redis/tree/6.0.8</a>

Запуск:

1. Загрузить установочный комплект Программного обеспечения в домашний каталог текущего пользователя.

Ссылка: [https://registry.skblab.ru/labfront\\_raw/distribution-fl.tar.gz](https://registry.skblab.ru/labfront_raw/distribution-fl.tar.gz)

**Логин для загрузки экземпляра ПО:**

labfront\_r

**Пароль для загрузки экземпляра ПО:**

azBs7TZgR53dC9W7qzTc

Пример скачивания на сервер в домашний каталог текущего пользователя:

```
curl -u "username:password"
```

```
https://registry.skblab.ru/labfront_raw/distribution.tar.gz -o ~/distribution.tar.gz
```

В качестве username и password надо использовать отдельно предоставленные учётные данные для скачивания Docker образов.

2. Создать рабочий каталог, в котором будет производиться запуск «LAB\_Front. Мультибанковская платформа дистанционного обслуживания физических лиц». Распаковать скачанный ранее скачанный distribution.tar.gz

```
mkdir ~/lab_front/
```

```
cd ~/lab_front/
```

```
tar xf ~/distribution.tar.gz
```

```
root@testnode5:~# mkdir ~/lab_front/
root@testnode5:~# cd ~/lab_front/
root@testnode5:~/lab_front# tar xf ~/distribution.tar.gz
root@testnode5:~/lab_front# ls
app_log      core.tar     docker-compose.yaml  nginx-unprivileged.tar  rabbitmq.tar  web.tar
catacomb.tar db.tar      jetty-log           openmq.tar              redis-cluster.tar  web_log
```

3. Загрузить полученные образы в Docker  

```
for image in $(ls *.tar); do docker load -i ${image}; done
```
4. Убедиться, что каталоги для логов доступны на запись:  

```
chmod -R 777 *_log
```
5. Запустить систему  

```
docker-compose up -d
```
6. Прописать в hosts на рабочей станции, с которой будет осуществляться работа с системой «LAB\_Front. Мультибанковская платформа дистанционного обслуживания физических лиц» в браузере:

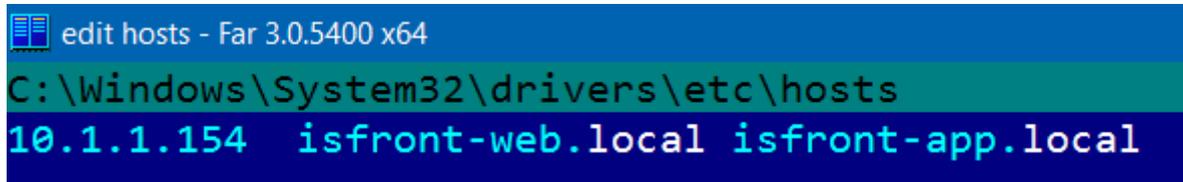
```
127.0.0.1      isfront-app.local isfront-web.local
```

Вместо 127.0.0.1 может быть IP адрес хоста, на котором производится запуск «LAB\_Front.

Мультибанковская платформа дистанционного обслуживания физических лиц».

Это необходимо для обеспечения правильной работы web-интерфейса системы «LAB\_Front.

Мультибанковская платформа дистанционного обслуживания физических лиц».



```
edit hosts - Far 3.0.5400 x64
C:\Windows\System32\drivers\etc\hosts
10.1.1.154 isfront-web.local isfront-app.local
```

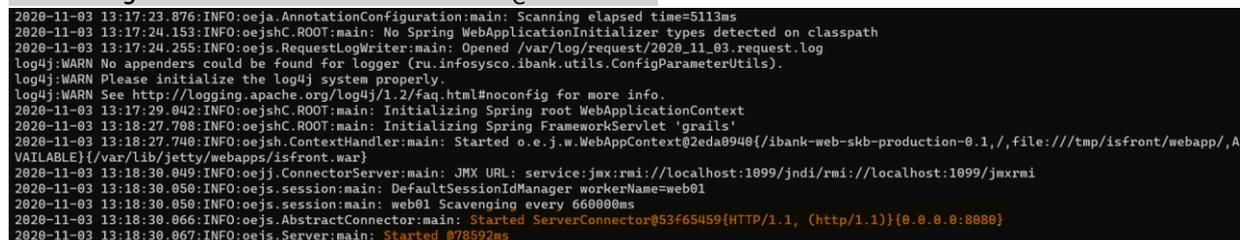
7. Дождаться старта систем:

#### 7.1. web:

Необходимо наблюдать за текстовым логом `~/lab_front/web_log/*.jetty.log`

При успешном запуске должна появиться строка вида:

```
INFO:oejs.Server:main: Started @148407ms
```



```
2020-11-03 13:17:23.876:INFO:oeja.AnnotationConfiguration:main: Scanning elapsed time=5113ms
2020-11-03 13:17:24.153:INFO:oejshC.ROOT:main: No Spring WebApplicationInitializer types detected on classpath
2020-11-03 13:17:24.255:INFO:oejs.RequestLogWriter:main: Opened /var/log/request/2020_11_03.request.log
log4j:WARN No appenders could be found for logger (ru.infosysco.ibank.util.ConfigParameterUtils).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
2020-11-03 13:17:29.042:INFO:oejshC.ROOT:main: Initializing Spring root WebApplicationContext
2020-11-03 13:18:27.798:INFO:oejshC.ROOT:main: Initializing Spring FrameworkServlet 'grails'
2020-11-03 13:18:27.740:INFO:oejsh.ContextHandler:main: Started o.e.j.w.WebAppContext@2eda8940{/ibank-web-skb-production-0.1/,file:///tmp/isfront/webapp/,AVAILABLE}{/var/lib/jetty/webapps/isfront.war}
2020-11-03 13:18:30.049:INFO:oejj.ConnectorServer:main: JMX URL: service:jmx:rmi://localhost:1099/jndi/rmi://localhost:1099/jmxrmi
2020-11-03 13:18:30.050:INFO:oejs.session:main: DefaultSessionIdManager workerName=web01
2020-11-03 13:18:30.050:INFO:oejs.session:main: web01 Scavenging every 660000ms
2020-11-03 13:18:30.066:INFO:oejs.AbstractConnector:main: Started ServerConnector@53f65459{HTTP/1.1, (http/1.1)}{0.0.0.0:8080}
2020-11-03 13:18:30.067:INFO:oejs.Server:main: Started @78592ms
```

#### 7.2. app:

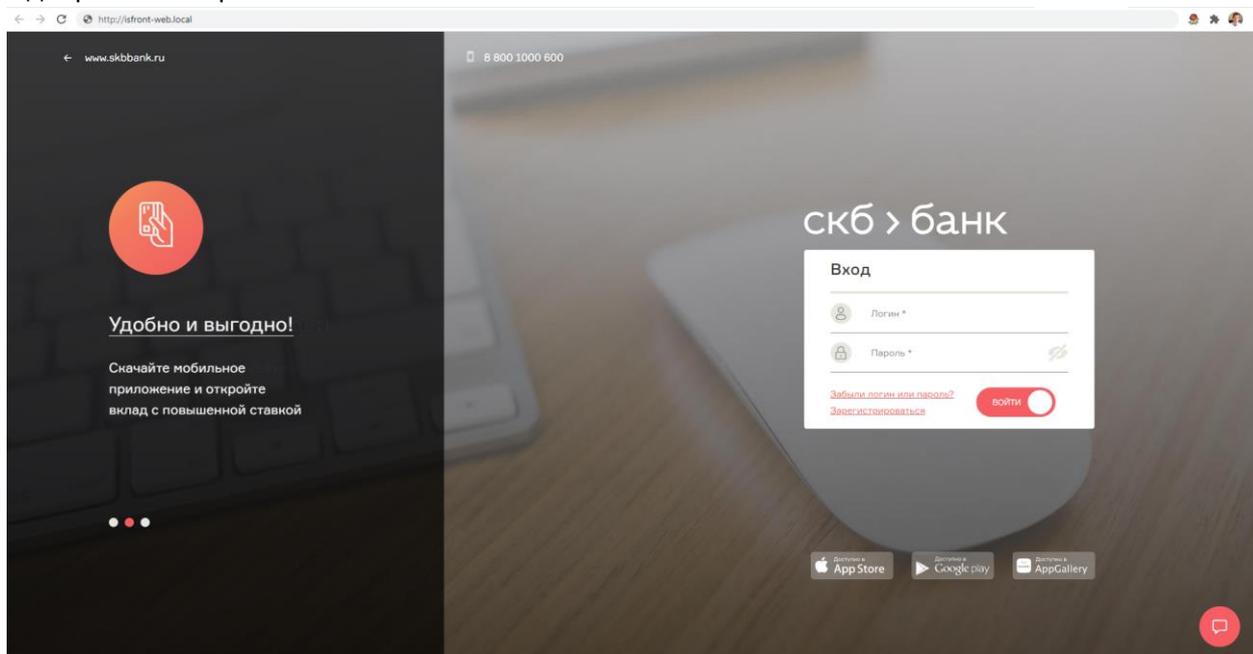
Необходимо наблюдать за текстовым логом `~/lab_front/app_log/*.jetty.log`

При успешном запуске должна появиться строка:

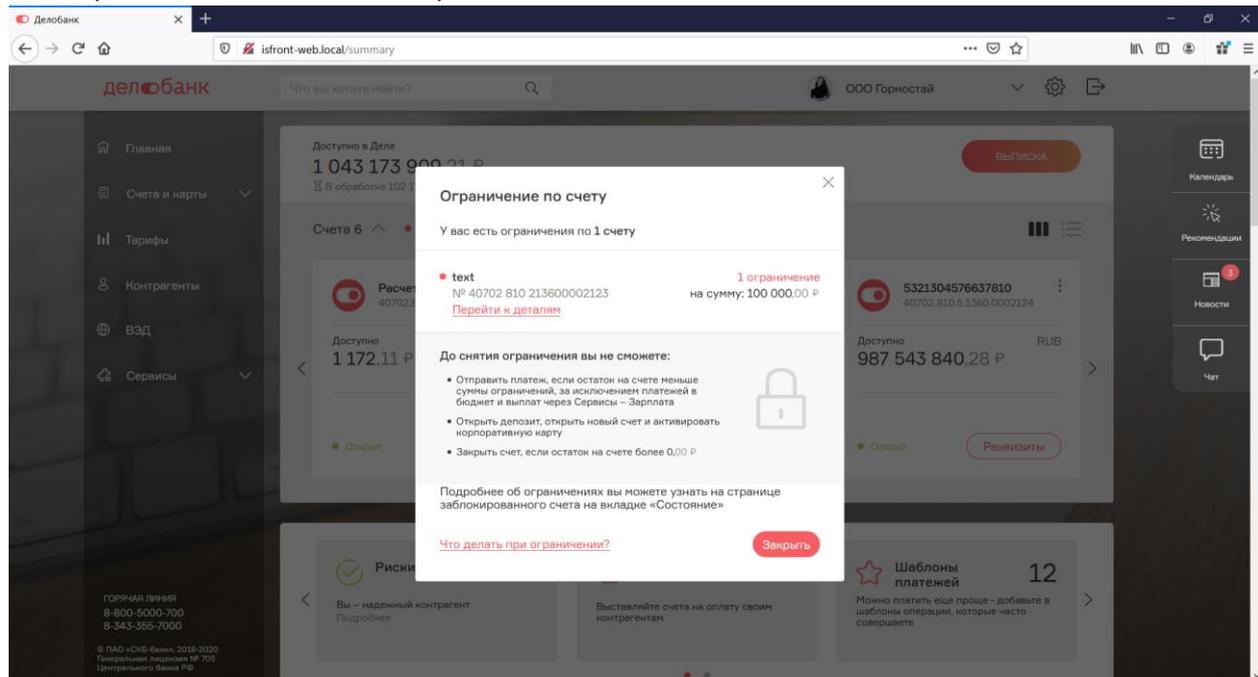
```
INFO:oejs.Server:main: Started @139655ms
```

```
root@testnode5:~/lab_front$tail -f ~/lab_front/app_log/2020_11_03_jetty.log
2020-11-03 13:06:00.408:WARN:oeja.AnnotationParser:qtp312116338-41: oracle.sql.converter.JdbcCharacterConverters scanned from multiple locations: jar:file:///tmp/isfront/webapp/WEB-INF/lib/jdbc-11.2.0.3.0.jar!/oracle/sql/converter/JdbcCharacterConverters.class, jar:file:///tmp/isfront/webapp/WEB-INF/lib/ojdbc6.jar!/oracle/sql/converter/JdbcCharacterConverters.class
2020-11-03 13:06:00.408:WARN:oeja.AnnotationParser:qtp312116338-41: oracle.sql.converter.Oracle18nCharacterSetMetaData scanned from multiple locations: jar:file:///tmp/isfront/webapp/WEB-INF/lib/ojdbc6.jar!/oracle/sql/converter/Oracle18nCharacterSetMetaData.class, jar:file:///tmp/isfront/webapp/WEB-INF/lib/ojdbc6.jar!/oracle/sql/converter/Oracle18nCharacterSetMetaData.class
2020-11-03 13:06:00.408:WARN:oeja.AnnotationParser:qtp312116338-41: oracle.sql.utilpack scanned from multiple locations: jar:file:///tmp/isfront/webapp/WEB-INF/lib/jdbc-11.2.0.3.0.jar!/oracle/sql/utilpack.class, jar:file:///tmp/isfront/webapp/WEB-INF/lib/ojdbc6.jar!/oracle/sql/utilpack.class
2020-11-03 13:06:02.038:INFO:oeja.AnnotationConfiguration:main: Scanning elapsed time=4864ms
2020-11-03 13:06:02.439:INFO:oejs.C.ROOT:main: No Spring WebApplicationInitializer types detected on classpath
2020-11-03 13:06:02.546:INFO:oejs.RequestLogWriter:main: Opened /var/log/request/2020_11_03_request.log
log4j:WARN No appenders could be found for logger (ru.infosysco.ibank.utils.ConfigParameterUtils).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
2020-11-03 13:06:06.936:INFO:oejs.C.ROOT:main: Initializing Spring root WebApplicationContext
2020-11-03 13:07:01.703:INFO:oejs.C.ROOT:main: Initializing Spring FrameworkServlet 'grails'
2020-11-03 13:07:01.758:INFO:oejs.ContextHandler:main: Started o.e.j.w.WebAppContext@2eda0940{/ibank-core-skb-production-0.1.1/,file:///tmp/isfront/webapp/,AVAILABLE}{/var/lib/jetty/webapps/isfront.war}
2020-11-03 13:07:01.855:INFO:oejj.ConnectorServer:main: JMX URL: service:jmx:rmi://localhost:1099/jndi/rmi://localhost:1099/jmxrmi
2020-11-03 13:07:01.857:INFO:oejs.session.main: DefaultSessionIdManager workerName=app01
2020-11-03 13:07:01.859:INFO:oejs.session.main: app01 Scavenging every 660000ms
2020-11-03 13:07:01.881:INFO:oejs.AbstractConnector:main: Started ServerConnector@53f65459{HTTP/1.1, (http/1.1)}{0.0.0.0:8080}
2020-11-03 13:07:01.882:INFO:oejs.Server:main: Started @72825ms
```

- 7.3. Старт системы обычно занимает от 2 до 5 минут, в зависимости от мощности железа.
- 8. После старта web и app приложение доступно:
  - 8.1. Web-интерфейс пользователя <http://isfront-web.local> (лучше пользоваться EDGE или FireFox)  
Логин: ZaharovAA  
Пароль: 1234  
Одноразовый пароль: 1234



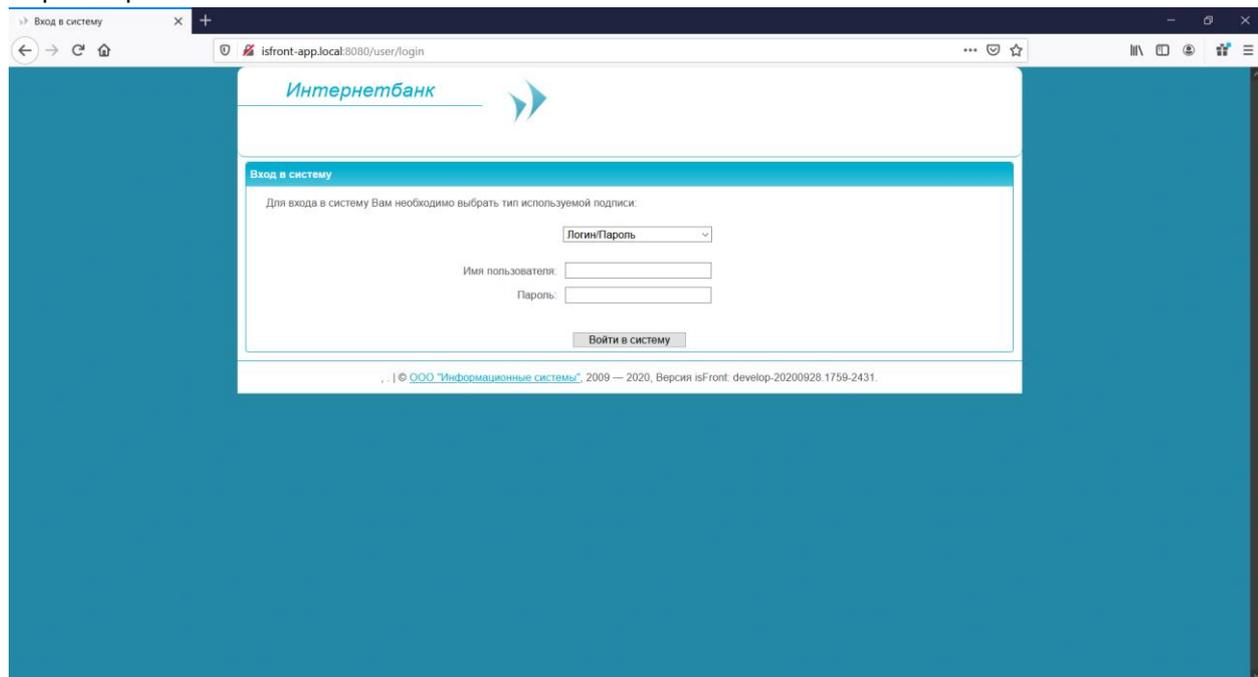
После успешного логина в систему:



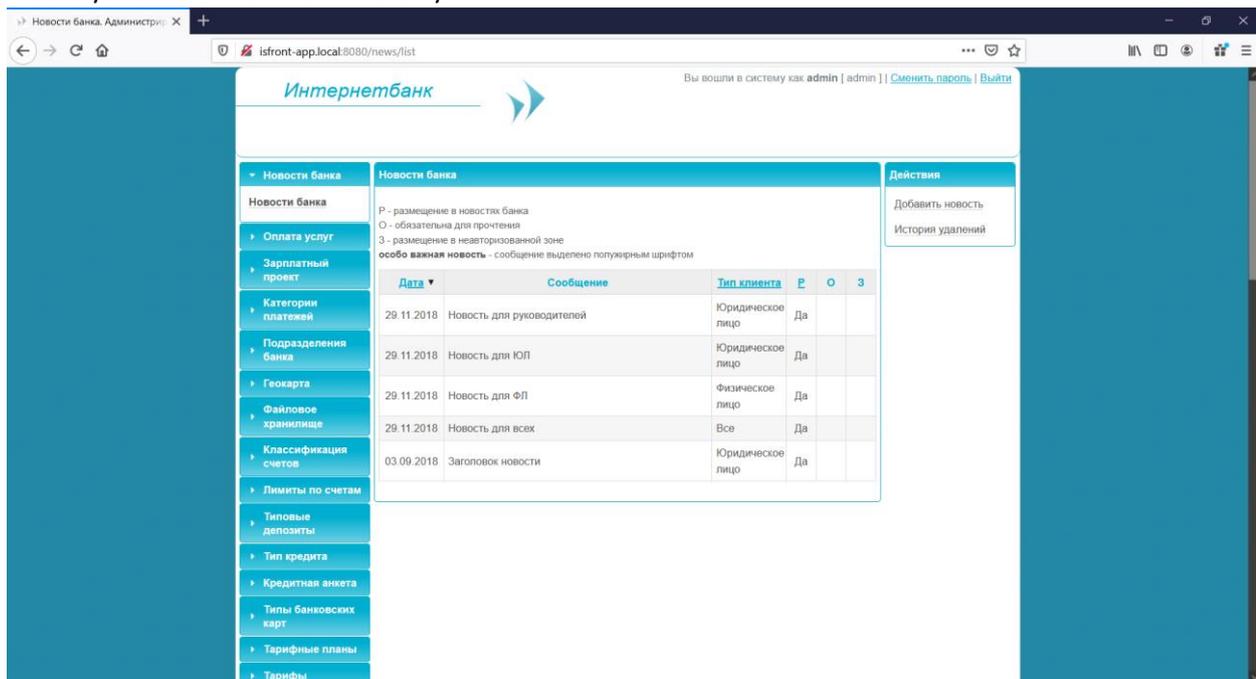
8.2. Интерфейс администратора <http://isfront-app.local:8080> (лучше пользоваться EDGE или FireFox)

Логин: admin

Пароль: qaz123



## После успешного логина в систему:



## Логирование:

1. RabbitMQ, Postgres, Catacomb, Redis - логирование производится в stdout, просмотр логов возможен средствами Docker.

Например

```
docker logs catacomb
```

```
root@testnode5:~/lab_front$ docker logs catacomb | tail -n 30
2020-11-03 13:17:33.493 DEBUG 1 --- [main] r.s.catacomb.configs.SecurityConfig : Configuring HttpSecurity
2020-11-03 13:17:33.568 INFO 1 --- [main] o.s.s.web.DefaultSecurityFilterChain : Creating filter chain: any request, [org.springframework.se
curity.web.context.request.async.WebAsyncManagerIntegrationFilter@c568f91, org.springframework.security.web.context.SecurityContextPersistenceFilter@39c87b4
2, org.springframework.security.web.header.HeaderWriterFilter@29bbc63c, org.springframework.security.web.authentication.logout.LogoutFilter@3cb6e0ee, org.sp
ringframework.security.web.savedrequest.RequestCacheAwareFilter@236c098, org.springframework.security.web.servletapi.SecurityContextHolderAwareRequestFilter
@6ad6443, org.springframework.security.web.authentication.AnonymousAuthenticationFilter@5fd43e58, org.springframework.security.web.session.SessionManagement
Filter@234cd86c, org.springframework.security.web.access.ExceptionTranslationFilter@6c4ce583, org.springframework.security.web.access.intercept.FilterSecurity
Interceptor@14d8a132]
2020-11-03 13:17:33.589 DEBUG 1 --- [main] ru.skblab.catacomb.configs.WebConfig : Configuring MappingJackson2XmlHttpMessageConverter
2020-11-03 13:17:35.118 INFO 1 --- [main] o.s.b.a.e.web.EndpointLinksResolver : Exposing 3 endpoint(s) beneath base path '/actuator'
2020-11-03 13:17:35.298 INFO 1 --- [main] o.s.i.endpoint.EventDrivenConsumer : Adding [logging-channel-adapter:_org.springframework.integr
ation.errorLogger] as a subscriber to the 'errorChannel' channel
2020-11-03 13:17:35.299 INFO 1 --- [main] o.s.i.channel.PublishSubscribeChannel : Channel 'application.errorChannel' has 1 subscriber(s).
2020-11-03 13:17:35.299 INFO 1 --- [main] o.s.i.endpoint.EventDrivenConsumer : started bean '_org.springframework.integration.errorLogger'
2020-11-03 13:17:35.984 INFO 1 --- [main] DeferredRepositoryInitializationListener : Triggering deferred initialization of Spring Data repositor
ies_
2020-11-03 13:17:36.136 INFO 1 --- [main] DeferredRepositoryInitializationListener : Spring Data repositories initialized!
2020-11-03 13:17:36.269 INFO 1 --- [main] o.s.a.r.c.CachingConnectionFactory : Attempting to connect to: [rabbitmq:5672]
2020-11-03 13:17:36.486 INFO 1 --- [main] o.s.a.r.c.CachingConnectionFactory : Created new connection: rabbitConnectionFactory#51d8f2f2:0/
SimpleConnection@1179fc8c [delegate=amqp://isfront@100.64.15.2:5672/test, localPort= 47002]
2020-11-03 13:17:36.598 INFO 1 --- [main] o.s.c.s.m.DirectWithAttributesChannel : Channel 'application.fastgate_output' has 1 subscriber(s).
2020-11-03 13:17:36.609 INFO 1 --- [main] o.s.c.s.m.DirectWithAttributesChannel : Channel 'application.events' has 1 subscriber(s).
2020-11-03 13:17:36.659 INFO 1 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path '/'
2020-11-03 13:17:36.661 INFO 1 --- [main] DeferredRepositoryInitializationListener : Triggering deferred initialization of Spring Data repositor
ies_
2020-11-03 13:17:36.661 INFO 1 --- [main] DeferredRepositoryInitializationListener : Spring Data repositories initialized!
2020-11-03 13:17:36.678 DEBUG 1 --- [ scheduling-1] r.s.c.c.e.p.EventsProcessingScheduler : Started processing pending events.
2020-11-03 13:17:36.686 INFO 1 --- [main] ru.skblab.catacomb.CatacombApplication : Started CatacombApplication in 22.932 seconds (JVM running
for 24.528)
2020-11-03 13:17:36.850 DEBUG 1 --- [ scheduling-1] r.s.c.c.e.p.EventsProcessingScheduler : Pending events: []
2020-11-03 13:20:36.852 DEBUG 1 --- [ scheduling-1] r.s.c.c.e.p.EventsProcessingScheduler : Started processing pending events.
2020-11-03 13:20:36.868 DEBUG 1 --- [ scheduling-1] r.s.c.c.e.p.EventsProcessingScheduler : Pending events: []
2020-11-03 13:23:36.871 DEBUG 1 --- [ scheduling-1] r.s.c.c.e.p.EventsProcessingScheduler : Started processing pending events.
2020-11-03 13:23:36.885 DEBUG 1 --- [ scheduling-1] r.s.c.c.e.p.EventsProcessingScheduler : Pending events: []
2020-11-03 13:26:36.886 DEBUG 1 --- [ scheduling-1] r.s.c.c.e.p.EventsProcessingScheduler : Started processing pending events.
2020-11-03 13:26:36.894 DEBUG 1 --- [ scheduling-1] r.s.c.c.e.p.EventsProcessingScheduler : Pending events: []
2020-11-03 13:29:36.895 DEBUG 1 --- [ scheduling-1] r.s.c.c.e.p.EventsProcessingScheduler : Started processing pending events.
2020-11-03 13:29:36.906 DEBUG 1 --- [ scheduling-1] r.s.c.c.e.p.EventsProcessingScheduler : Pending events: []
```

2. Логирование jms производится в файл jms\_log/log.txt

Для просмотра потребуются права суперпользователя, поскольку по умолчанию файл создаётся с правами 600

3. Приложения isfront-web, isfront-app логируют в несколько логов. Структура логово одинаковая, на примере isfront-web:

```
web_log/
```

```
|— 2020_10_21.jetty.log # Лог работы сервлет-контейнера Jetty.
```

```
|— application # Каталог для прикладных логов.  
|   |— 2020_10_21_10.isfront-web.isf-web01.log.gz # Прикладной лог за  
2020-10-21 с 10:00:00 по 10:59:59 в заархивированном виде  
|   |— 2020_10_21_11.isfront-web.isf-web01.log.gz # Прикладной лог  
за 2020-10-21 за 11 час в заархивированном виде  
|   └─ 2020_10_21_12.isfront-web.isf-web01.log # Текущий прикладной  
лог  
|— jettygc.log.0.current # Лог Java Garbage Collector  
└─ request # Каталог для логирования HTTP запросов  
   └─ 2020_10_21.request.log # Лог HTTP запросов за 2020-10-21
```

Возможные неисправности:

1. Web-интерфейс не стартует
  - 1.1. Убедиться, что хватает места на диске, на котором работает Docker.  
В случае необходимости можно остановить систему «LAB\_Front. Мультибанковская платформа дистанционного обслуживания физических лиц» и очистить все логи.  

```
docker-compose stop  
find "*_log" -t f -exec rm -f {} \;  
docker-compose start
```
  - 1.2. Не начинается запись в файлы логов.
    - 1.2.1. Убедиться, что сервисы стартовали  

```
docker-compose ps
```
    - 1.2.2. Убедиться, что сервисам достаточно прав для создания логов в каталогах для логов.  
В случае необходимости выдать права на каталоги  

```
docker-compose stop  
find . -t d -exec chmod 777 {} \;  
docker-compose start
```
2. Не получается залогиниться в «LAB\_Front. Мультибанковская платформа дистанционного обслуживания физических лиц»
  - 2.1. Просмотреть текущий прикладной лог в web\_log/application/  
Надо искать там ошибки соединения с catacomb.  
Можно искать так:  

```
grep ERROR web_log/application/*.log  
grep catacomb web_log/application/*.log
```
  - 2.2. Убедиться, что сервер аутентификации catacomb доступен с web.  
Для этого надо зайти в контейнер с web и выполнить запрос к catacomb  

```
docker exec -it isfront-web sh  
wget -q -O - http://catacomb:8080/actuator/health
```

  
Если всё хорошо - от catacomb будет ответ {"status": "UP"}  
Если ответ отличается - надо смотреть лог catacomb и искать там ошибки  

```
docker logs catacomb | less +G
```
  - 2.3. Аналогичным способом проверить доступность сервера базы данных postgres.
  - 2.4. Проверить доступность сервера jms.
3. Не проходит аутентификация, после ввода одноразового пароля просит снова авторизоваться.

- 3.1. Проверить, что используется браузер Firefox/Edge/Opera. Представленный демостенд работает по HTTP без шифрования, поэтому есть особенности при работе разных браузеров. В промышленной эксплуатации всегда используется https.

В случае возникновения вопросов или проблем по запуску – всегда можно

1. Написать письмо с запросом на техническую поддержку на адрес [info@skblab.ru](mailto:info@skblab.ru)
2. Позвонить в техническую поддержку по телефону +7 (343) 355 73 72