



Медицинская информационная система «LynxClinic»

Руководство по установке



Оглавление

Общая информация.....	3
1 Настройка сервера приложений.....	3
2 Настройка СУБД.....	6
3 Наполнение справочников Системы.....	8

Общая информация

Медицинская информационная система LynxClinic (далее - Система) распространяется в виде дистрибутива в формате WAR-файла (Web ARchive - формат файла для упаковки Web-приложения). Подготовка дистрибутива осуществляется индивидуально для каждого клиента. При этом дистрибутив именуется в следующем формате:

`<client>.war`

где `<client>` - наименование организации клиента латиницей без использования специальных символов и цифр. Пример: `myclinic.war`

Для работы дистрибутива необходимо выполнение подготовительных работ: настройка сервера приложений, настройка СУБД и наполнение справочников системы.

Минимальные рекомендуемые технические требования к серверу:

- Процессор: 2 ядра;
- Оперативная память: 4 Гб;
- Дисковая подсистема: 200 Гб;
- Операционная система: Linux

Для корректной работы в среде Linux необходимо наличие следующего программного обеспечения, которое не входит в состав дистрибутива:

- Apache Tomcat (версия 8.0)
- Oracle Java SE 8
- СУБД PostgreSQL 9.2
- JDBC-драйвер PostgreSQL
-

1 Настройка сервера приложений

Для выполнения дистрибутива необходима установка сервера приложений Apache Tomcat (версия 8.0). Сервер приложений Apache Tomcat (далее - Tomcat) распространяется по открытой лицензии Apache License 2.0 и доступен для скачивания по адресу <https://archive.apache.org/dist/tomcat/tomcat-8/>. Во время установки Tomcat все параметры можно оставить по умолчанию. Для

обеспечения возможности подключения приложения к СУБД необходимо скачать с сайта <https://jdbc.postgresql.org/> JDBC-драйвер для подключения Java-приложений к СУБД PostgreSQL. Драйвер распространяется в виде файла в формате JAR. Скачанный файл необходимо разместить в директории /lib/ сервера приложений Tomcat.

Доступ к Системе осуществляется по адресу:

`http://<server_address>:<http_port>/<client>/`

где:

- <server_address> - адрес сервера, доступный для подключения с рабочих мест пользователей системы
- <http_port> - порт сервера приложений Tomcat. Значение по умолчанию - 8080
- <client> - наименование организации клиента латиницей без использования спец.символов и цифр

Пример адреса Системы - `http://192.168.0.1:8080/myclinic/`

Опционально имеется возможность использования сервера приложений совместно с HTTP-сервером Apache или Nginx. Для настройки достаточно выполнить проксирование запросов на порт 8080 сервера приложений Tomcat. В этом случае можно избежать необходимости указывать порт сервера приложения в строке адреса Системы. Дистрибутив Системы в виде WAR-файла необходимо разместить в директории /webapps/ сервера приложений Tomcat или загрузить через Панель управления Tomcat, доступную по адресу:

`http://<server_address>:<http_port>/manager/`

(например, <http://192.168.0.1:8080/manager/>). Дистрибутив Системы будет автоматически загружен и доступен для использования. Также дистрибутив автоматически загружается при последующих запусках Tomcat.

Для обновления Системы достаточно заменить WAR-файл в директории /webapps/ сервера приложений Tomcat или загрузить новый WAR-файл через Панель управления Tomcat.

Настройка параметров Системы осуществляется через конфигурационный файл `ssnpp.properties`, который необходимо разместить в директории /conf/ сервера приложений Tomcat. Конфигурационный файл должен содержать следующие параметры:

```
client.name=<client_name>
# Database connection configuration
jdbc.driverClassName=org.postgresql.Driver
jdbc.url=jdbc:postgresql://<postgres_address>:<postgres_port>/mis_<client>
jdbc.username=<client>_tomcat
jdbc.password=<client_tomcat_password>
# FileAttachments path
medicalRecord.attachmentsPath=<attachments_path>
```

где:

- <client_name> - наименование организации клиента, допускаются любые символы
- <postgres_address> - адрес, по которому доступна СУБД PostgreSQL. Значение по умолчанию - 127.0.0.1
- <postgres_port> - порт СУБД PostgreSQL. Значение по умолчанию - 5432
- <client> - наименование организации клиента латиницей без использования спец.символов и цифр
- <client_tomcat_password> - пароль пользователя БД для Tomcat
- <attachments_path> - путь сервера к директории для хранения файлов, прикрепляемых пользователями Системы при ее эксплуатации. Пример - /opt/tomcat/misAttachments

2 Настройка СУБД

Для работы Системы требуется наличие установленной СУБД PostgreSQL 9.2. Во время установки PostgreSQL все параметры можно оставить по умолчанию. Пароль для пользователя postgres назначается любой, по усмотрению пользователя. После установки PostgreSQL рекомендуется включить логирование. Для этого в конфигурационном файле postgresql.conf необходимо параметру logging_collector задать значение "on":

```
logging_collector=on
```

Для запуска Системы необходимо создание:

- пользователя-владельца базы данных;
Пользователь имеет полные права на работу с базой данных, владельцем которой является. От имени данного пользователя выполняется создание структуры базы данных, необходимой для работы Системы. Создание пользователя выполняется средствами PostgreSQL путем выполнения SQL-запроса:

```
CREATE ROLE <client> NOINHERIT LOGIN PASSWORD  
'<client_password>';
```

где:

- <client> - наименование организации клиента латиницей без использования спец.символов и цифр
 - <client_password> - пароль пользователя-владельца
- пользователя для Tomcat

Пользователь имеет права на изменение данных, но не может менять структуру БД и создавать объекты.

Создание пользователя выполняется средствами PostgreSQL путем выполнения SQL-запроса:

```
CREATE ROLE <client>_tomcat NOINHERIT LOGIN PASSWORD  
'<client_tomcat_password>';
```

где:

- <client> - наименование организации клиента латиницей без использования спец.символов и цифр
- <client_tomcat_password> - пароль пользователя БД для Tomcat

- базы данных

Владельцем базы данных является ранее созданный пользователь.
Используемая кодировка - UTF8.

Создание пользователя выполняется средствами PostgreSQL путем выполнения SQL-запроса:

```
CREATE DATABASE mis_<client>
  WITH OWNER = <client>
  ENCODING = 'UTF8';
```

где <client> - наименование организации клиента латиницей без использования спец.символов и цифр

После создания пользователя и базы данных необходимо проставить некоторые привилегии путем выполнения SQL-запроса:

```
GRANT CREATE
  ON SCHEMA public TO <client>;
GRANT USAGE
  ON SCHEMA public TO <client>;
GRANT USAGE
  ON SCHEMA public TO <client>_tomcat;
GRANT CONNECT
  ON DATABASE mis_<client> TO <client>;
GRANT CREATE
  ON DATABASE mis_<client> TO <client>;
GRANT TEMP
  ON DATABASE mis_<client> TO <client>;
GRANT CONNECT
  ON DATABASE mis_<client> TO <client>_tomcat;
```

где <client> - наименование организации клиента латиницей без использования спец.символов и цифр

Структура БД, необходимая для работы Системы, создается автоматически при первом запуске приложения.

3 Наполнение справочников Системы

Для использования Системы необходимо выполнить первоначальное наполнение следующих ее справочников:

- пользователь-администратор Системы
Логин и пароль по умолчанию - admin / admin
- организация-владелец Системы
- филиал организации-владельца Системы
- отделение организации-владельца Системы
- принадлежность пользователя-администратора к организации-владельцу Системы
- роли пользователя в организации-владельце Системы
- страховая организация
- касса

Первоначальное наполнение справочников выполняется средствами PostgreSQL путем выполнения SQL-запроса:

```
BEGIN;
```

```
-- 1 Удалить констрэйнты, ограничивающие вставку  
ALTER TABLE basic.users  
  DROP CONSTRAINT users_owner_organization_fk RESTRICT;
```

```
ALTER TABLE basic.users  
  DROP CONSTRAINT users_user_created_id_fk RESTRICT;
```

```
COMMIT;
```

```
-- 2 Вставить данные  
BEGIN;
```

```
INSERT INTO  
  basic.users  
(  
  login,  
  short_name,  
  full_name,  
  password_hash,  
  user_changed_id,  
  user_created_id,  
  owner_organization_id  
)
```

```

VALUES (
  'admin',
  'Администратор',
  'Администратор',

  '8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a81f6f2ab448
a918',
  1,
  1,
  1
);

INSERT INTO
  basic.owner_organizations
(
  owner_organization_name,
  legal_address,
  ogrn,
  inn,
  kpp,
  bank,
  bik,
  ks,
  rs,
  date_created,
  license,
  license_start_date,
  phone_number_reception,
  license_period,
  license_register,
  owner_organization_short_name,
  user_changed_id,
  user_created_id
)
VALUES (
  'LynxClinic',
  'г.Барнаул, ул.Ленина, 111, офис 1',
  '00000000000000',
  '000000000000',
  '0000000',
  'Банк России',
  '1111',
  '111111111',
  '2222222',
  now(),
  '№ 123',
  '2010-03-01',

```

```

    '22-22-22',
    '',
    '',
    'LynxSoft',
    1,1
);

INSERT INTO
  basic.filials
(
  owner_organization_id,
  short_name,
  actual_address,
  head_doctor,
  date_created,
  phone_number_registry,
  head_doctors_post,
  attorney_number,
  attorney_date,
  user_changed_id,
  user_created_id
)
VALUES (
  1,
  'Филиал 1',
  'г.Барнаул, ул.Ленина, 111, офис 1',
  'Иванов А.С.',
  now(),
  '23-23-23',
  'главный врач',
  '123',
  '2010-06-03',
  1,1
);

```

```

INSERT INTO
  basic.organizations
(
  short_name,
  code_name,
  email,
  full_name,
  organization_location,
  ogrn,
  okato,
  phone_information,
  phone_registry,

```

```

        user_changed_id,
        user_created_id
    )
VALUES (
    'ЛПУ',
    'ЛПУ',
    'help@ccnpp.ru',
    'ЛПУ',
    '',
    '0000000000000000',
    '000000000000',
    '',
    '',
    1,
    1
);

INSERT INTO
    basic.departments
(
    short_name,
    code_name,
    full_name,
    kind_department,
    department_location,
    order_rank,
    phone_information,
    filial_id,
    user_changed_id,
    user_created_id
)
VALUES (
    'Техническое',
    'Тех',
    'Техническое отделение',
    'TECHNICAL',
    '',
    NULL,
    '',
    1,
    1,
    1
);

INSERT INTO
    basic.user_in_organizations
(

```

```

    user_id,
    owner_organization_id,
    date_created,
    user_created_id,
    user_changed_id
)
VALUES (
    1,
    1,
    now(),
    1,
    1
);

INSERT INTO
    basic.user_organization_roles
(
    user_organization_id,
    role_name
)
VALUES (
    1,
    'SUPER_ADMINISTRATOR'
);

INSERT INTO
    cashbox.cashboxes
(
    filial_id,
    short_name,
    description,
    date_created,
    user_changed_id,
    user_created_id
)
VALUES (
    1,
    'Kacca №1 ',
    'Kacca №1 ',
    now(),
    1,
    1
);

COMMIT;

BEGIN;
```

```
-- 3 Вернуть констрэйнты, ограничивающие вставку
ALTER TABLE basic.users
  ADD CONSTRAINT users_owner_organization_fk FOREIGN KEY
  (owner_organization_id)
  REFERENCES basic.owner_organizations(id)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
  NOT DEFERRABLE;

ALTER TABLE basic.users
  ADD CONSTRAINT users_user_created_id_fk FOREIGN KEY
  (user_created_id)
  REFERENCES basic.users(id)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
  NOT DEFERRABLE;

COMMIT;
```

После загрузки данных система готова к работе: можно перейти на страницу авторизации, авторизоваться от имени пользователя-администратора и начать эксплуатацию Системы.

Также имеется возможность импортировать дополнительные справочники, которые могут потребоваться при эксплуатации Системы: справочник МКБ-10, справочник страховых компаний, классификатор адресов ФИАС.