
Документация

Выпуск 6.6.7

Руководство администратора XSQUARE - RAD

июл. 20, 2026

Содержание

1	Общие сведения	1
1.1	Для кого это руководство	1
2	Быстрая установка	2
2.1	Быстрая установка на DEB-based ОС	2
2.2	Быстрая установка на RPM-based ОС	4
3	Архитектура и системные требования	7
3.1	Архитектура	7
3.2	Системные требования	8
4	Установка и настройка	10
4.1	Установка XRAD	10
4.2	Настройка NGINX	11
4.3	Настройка Apache2 на DEB-based ОС	12
4.4	Настройка Apache2 на RPM-based ОС	12
4.5	Настройка XRAD	13
5	Конфигурационные файлы	14
5.1	Файл конфигурации config.json	14
5.2	Файл конфигурации схем аутентификации auth_config.json	17
6	Функциональные характеристики	21
6.1	Что такое XRAD?	21
6.2	Как работает XRAD?	22
7	Эксплуатация	23
7.1	Сервер приложений	23
7.2	База данных	23

Общие сведения

Данное руководство описывает как выполнять административные задачи для среды разработки XRAD.

1.1 Для кого это руководство

Руководство предназначено для администраторов систем и разработчиков приложений. Вы должны обладать базовым пониманием концепции работы баз данных и знать особенности операционной системы на которой выполняется приложение XRAD.

Быстрая установка

2.1 Быстрая установка на DEB-based ОС

Рассмотрим быструю установку XRAD + PGHS на примере ОС Debian. Все команды следует выполнять с правами суперпользователя (root)

1. Создаем каталог для дистрибутива

```
mkdir /root/xsquare
```

переходим в каталог

```
cd /root/xsquare
```

2. Скачиваем/получаем дистрибутив в созданный каталог

```
wget https://ftp.xsquare.ru/edu/files/...version.zip
```

3. Распаковываем дистрибутив

```
apt -y install unzip  
unzip xsquare.[APP_VERSION].zip
```

4. Переходим в каталог с файлами дистрибутива

```
cd xsquare.[APP_VERSION]
```

5. Настраиваем часовой пояс и локализацию ОС

```
echo "Europe/Moscow" > /etc/timezone && \  
dpkg-reconfigure -f noninteractive tzdata && \  
sed -i -e 's/# en_US.UTF-8 UTF-8/en_US.UTF-8 UTF-8/' /etc/locale.gen && \  
sed -i -e 's/# ru_RU.UTF-8 UTF-8/ru_RU.UTF-8 UTF-8/' /etc/locale.gen && \  
echo 'LANG="ru_RU.UTF-8"'>/etc/default/locale && \  
(продолжается на следующей странице)
```

(продолжение с предыдущей страницы)

```
dpkg-reconfigure --frontend=noninteractive locales && \
update-locale LANG=ru_RU.UTF-8
update-locale LC_ALL=ru_RU.UTF-8
```

6. Устанавливаем PostgreSQL

```
apt -y install postgresql
```

7. Подготавливаем PostgreSQL

переключаемся на пользователя postgres

```
su - postgres
```

создаем пользователей БД xrad_user и app_user

```
psql -c "create user xrad_user with encrypted password 'xrad_user';"
psql -c "create user app_user with encrypted password 'app_user';"
```

создаем базы appdb и xraddb

```
psql -c "CREATE DATABASE \"appdb\" WITH OWNER \"app_user\" ENCODING 'UTF8' LC_COLLATE_
↪= 'ru_RU.UTF-8' LC_CTYPE = 'ru_RU.UTF-8';"
psql -c "CREATE DATABASE \"xraddb\" WITH OWNER \"xrad_user\" ENCODING 'UTF8' LC_
↪COLLATE = 'ru_RU.UTF-8' LC_CTYPE = 'ru_RU.UTF-8';"
```

назначаем пользователям xrad_user и app_user максимальные привилегии

```
psql -c "ALTER USER xrad_user WITH SUPERUSER;"
psql -c "ALTER USER app_user WITH SUPERUSER;"
```

выходим из сеанса учетной записи postgres

```
exit
```

8. Импортируем базы данных

```
export PGPASSWORD='xrad_user';
psql -U xrad_user -h 127.0.0.1 xraddb < db/xraddb.xsquare.pgsql

export PGPASSWORD='app_user';
psql -U app_user -h 127.0.0.1 appdb < db/appdb.xsquare.pgsql
```

9. Устанавливаем nginx

```
apt -y install nginx
```

отключаем сайт по умолчанию

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

копируем из дистрибутива файлы веб-контроллера для PGHS и XRAD

```
cp -R ./var /
```

копируем из дистрибутива конфигурационные файлы nginx

```
cp -R ./etc/nginx /etc/
```

10. Перезапускаем nginx

```
systemctl restart nginx
```

проверяем его состояние

```
systemctl status nginx
systemctl enable nginx
```

11. Копируем исполняемые и конфигурационные файлы XRAD, PGHS

```
cp -R ./etc/systemd /etc/
cp -R ./usr /
```

12. Запускаем XRAD как службу и проверяем статус

```
systemctl start xsquare.xrad.service
systemctl enable xsquare.xrad.service
systemctl status xsquare.xrad.service
```

13. Запускаем PGHS как службу и проверяем статус

```
systemctl start xsquare.pghs.service
systemctl enable xsquare.pghs.service
systemctl status xsquare.pghs.service
```

14. Проверяем доступность дефолтного веб-приложения и конструктора XRAD в браузере

Примечание: в случае проблем с доступом по http необходимо проверить настройки nginx и разрешения в брандмауэре.

2.2 Быстрая установка на RPM-based ОС

Рассмотрим быструю установку XRAD + PGHS на примере ОС Fedora. Все команды следует выполнять с правами суперпользователя (root)

1. Создаем каталог для дистрибутива

```
mkdir /root/xsquare
```

переходим в каталог

```
cd /root/xsquare
```

2. Скачиваем дистрибутив в созданный каталог

```
wget https://ftp.xsquare.ru/edu/files/...version.zip
```

3. Распаковываем дистрибутив

```
dnf install -y unzip
unzip xsquare.[APP_VERSION].zip
```

4. Переходим в каталог с файлами дистрибутива

```
cd xsquare.[APP_VERSION]
```

5. Настраиваем часовой пояс и локализацию ОС

```
timedatectl set-timezone Europe/Moscow
localectl set-locale LANG=ru_RU.UTF-8
export LANG=ru_RU.UTF-8
```

6. Устанавливаем и запускаем PostgreSQL

```
dnf install -y postgresql
postgresql-setup --initdb
systemctl start postgresql
systemctl enable postgresql
```

7. Подготавливаем PostgreSQL

переключаемся на пользователя postgres

```
su - postgres
```

создаем пользователей БД xrad_user и app_user

```
psql -c "create user xrad_user with encrypted password 'xrad_user';"
psql -c "create user app_user with encrypted password 'app_user';"
```

создаем базы appdb и xraddb

```
psql -c "CREATE DATABASE \"appdb\" WITH OWNER \"app_user\" ENCODING 'UTF8' LC_COLLATE_
↪= 'ru_RU.UTF-8' LC_CTYPE = 'ru_RU.UTF-8';"
psql -c "CREATE DATABASE \"xraddb\" WITH OWNER \"xrad_user\" ENCODING 'UTF8' LC_
↪COLLATE = 'ru_RU.UTF-8' LC_CTYPE = 'ru_RU.UTF-8';"
```

назначаем пользователям xrad_user и app_user максимальные привилегии

```
psql -c "ALTER USER xrad_user WITH SUPERUSER;"
psql -c "ALTER USER app_user WITH SUPERUSER;"
```

выходим из сеанса учетной записи postgres

```
exit
```

8. Импортируем базы данных

```
export PGPASSWORD='xrad_user';
psql -U xrad_user -h 127.0.0.1 xraddb < db/xraddb.xsquare.pgsql

export PGPASSWORD='app_user';
psql -U app_user -h 127.0.0.1 appdb < db/appdb.xsquare.pgsql
```

9. Устанавливаем nginx

```
dnf install -y nginx
```

отключаем сайт по умолчанию

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

копируем из дистрибутива файлы веб-контроллера для PGHS и XRAD

```
cp -R ./var /
```

копируем из дистрибутива конфигурационные файлы nginx

```
cp -R ./etc/nginx /etc/
```

10. Перезапускаем nginx

```
systemctl restart nginx
```

проверяем его состояние

```
systemctl --no-pager status nginx
```

11. Копируем исполняемые и конфигурационные файлы XRAD, PGHS

```
cp -R ./etc/systemd /etc/  
cp -R ./usr /
```

12. Запускаем XRAD как службу и проверяем статус

```
systemctl start xsquare.xrad.service  
systemctl enable xsquare.xrad.service  
systemctl --no-pager status xsquare.xrad.service
```

13. Запускаем PGHS как службу и проверяем статус

```
systemctl start xsquare.pghs.service  
systemctl enable xsquare.pghs.service  
systemctl --no-pager status xsquare.pghs.service
```

14. Проверяем доступность дефолтного веб-приложения и конструктора XRAD в браузере

Примечание: в случае проблем с доступом по http необходимо проверить настройки nginx и разрешения в брандмауэре.

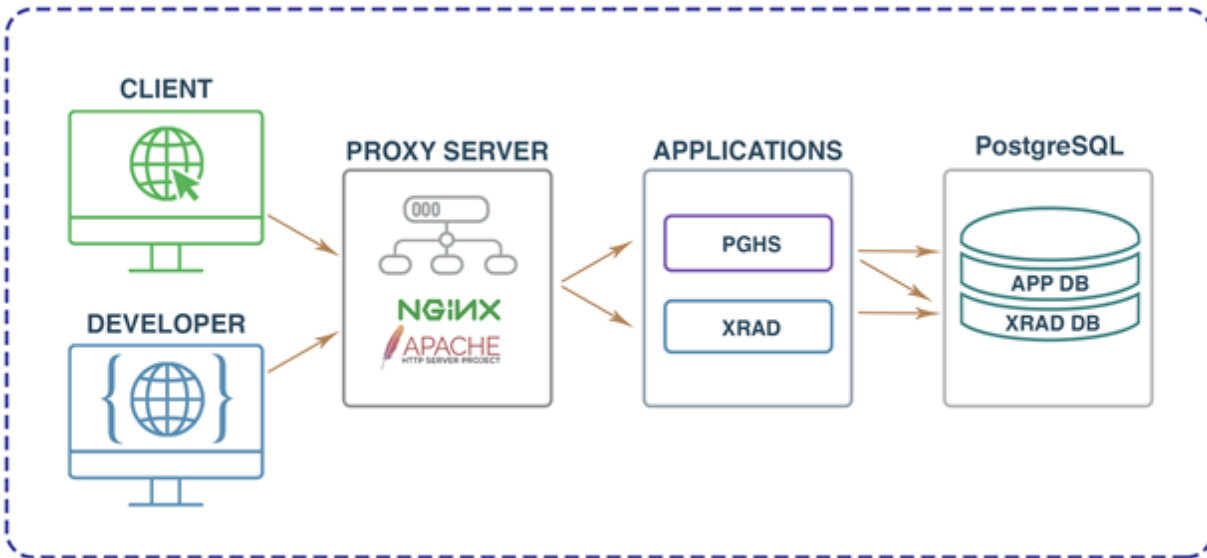
Архитектура и системные требования

3.1 Архитектура

Базовая архитектура XRAD состоит из 4-х компонентов:

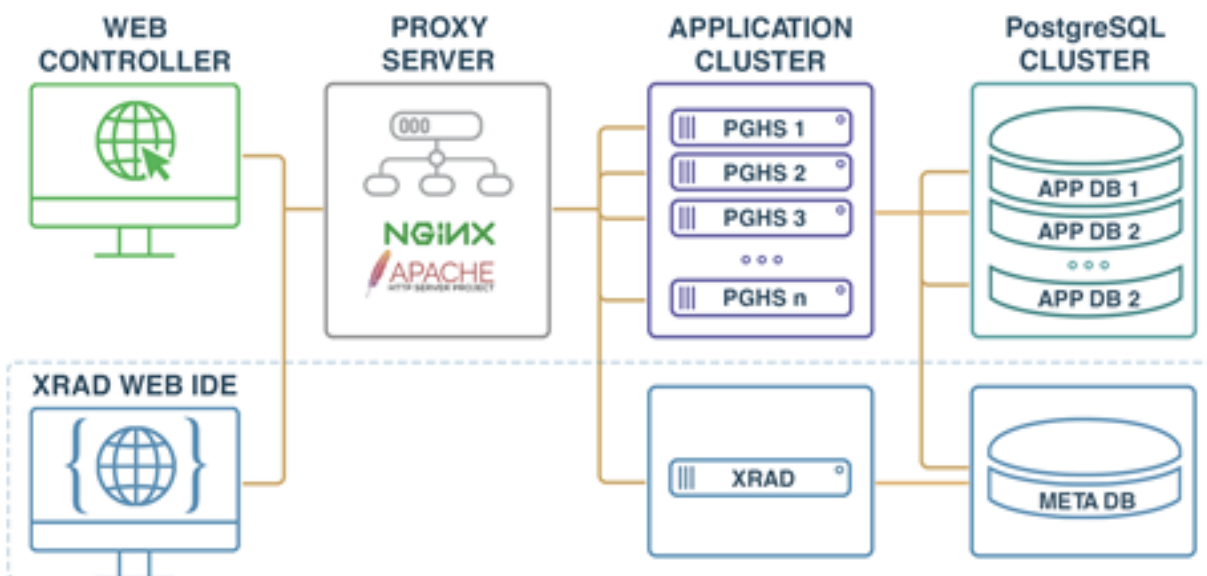
- База данных XRAD – база данных с метаданными приложения. В ней хранится информация о компонентах запрашиваемой станицы, процессах которые должны произойти во время обработки страницы.
- Сервер приложений XRAD, который отвечает за обработку действий разработчика и подготовку структуры страницы для рендеринга.
- Веб-контроллер XRAD, которое обрабатывает полученную от веб-сервера структуру страницы и отображает её.
- HTTP Proxy Server, который связывает веб-сервер и веб-контроллер XRAD.

Для отображения разрабатываемого веб-приложения необходим сервер приложений PGHS, поэтому XRAD и PGHS работают в связке и имеют одинаковую 4-х компонентную архитектуру. Независимо от того, запускаете ли вы среду разработки XRAD или приложение, созданное с использованием XRAD - процесс функционирования один и тот же. Ваш браузер отправляет запрос, который преобразуется в соответствующий вызов кода на стороне базы данных. После того, как база данных обработает код, результаты будут переданы обратно в ваш браузер в виде структуры JSON, на базе которой веб-приложение сформирует страницу. Этот цикл происходит каждый раз, когда вы запрашиваете или отправляете страницу.



Все компоненты могут быть развернуты как в рамках одного сервера, так и разнесены по разным физическим или виртуальным серверам.

Для высоконагруженных систем можно легко произвести горизонтальное масштабирование. Пример высоконагруженной архитектуры выглядит следующим образом:



3.2 Системные требования

3.2.1 Среда исполнения

Поддерживаемые архитектуры:

- x86-64

- ARM
- Loongson
- e2k

Поддерживаемые ОС:

- **DEB**-based – любые
- **RPM**-based – любые
- Debian 13 – рекомендуемая

Базы данных:

- PostgreSQL 13+ (17 - рекомендуемая)
- Firebird
- MySQL / MariaDB
- MSSQL
- Oracle

HTTP/Proxy Server:

- Apache 2.4+
- NGINX 19+

3.2.2 Системные требования

XRAD - сервер:

- CPU - 1 Ядро
- RAM - 100 Мб
- HDD - 100 Мб + Логи

XRAD DB:

- CPU - 1 Ядро
- RAM - 50 Мб
- HDD - 10 Мб БД PostgreSQL

Установка системы виртуализации/контейнеризации, операционной системы, базы данных осуществляется на усмотрение Администратора исходя из потребностей.

Установка и настройка

4.1 Установка XRAD

Подробная настройка ОС и установка всех компонентов кроме XRAD рассмотрены в документации PGHS. В данной документации рассмотрим непосредственно развертывание XRAD.

Примечание: примеры приводятся для случая, когда пользователь находится в каталоге с дистрибутивом, все действия выполняются с правами суперпользователя (root).

Для установки XRAD копируем исполняемые файлы из дистрибутива в каталог `/usr/local/xsquare.xrad`

Например:

1. Копируем все компоненты

```
cp -R ./usr /
```

или

2. Копируем непосредственно дистрибутив XRAD

```
cp -R ./usr/local/xsquare.xrad /usr/local/
```

Примечание: назначьте права на исполнение

```
chmod +x /usr/local/xsquare.xrad/xrad
```

Создаем сервис

```
vi /etc/systemd/system/xsquare.xrad.service
```

```
[Unit]
Description=XRAD Services
After=syslog.target network.target
After=postgresql.service
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```
[Service]
Type=simple
ExecStart=/usr/local/xsquare.xrad/xrad
WorkingDirectory=/usr/local/xsquare.xrad
Restart=on-failure
RestartSec=3

[Install]
WantedBy=default.target
```

Примечание: файл сервиса по умолчанию можно скопировать из дистрибутива. Например:

```
cp -R ./etc/systemd /etc/
```

Запускаем XRAD как службу и проверяем статус

```
systemctl start xsquare.xrad.service
systemctl enable xsquare.xrad.service
systemctl --no-pager status xsquare.xrad.service
```

Далее настроим HTTP proxy server на примере NGINX и Apache2

Копируем из дистрибутива файлы веб-контроллера XRAD

```
cp -R ./var/www/xrad.xsquare*
```

4.2 Настройка NGINX

Копируем из дистрибутива конфигурационные файлы nginx

```
cp -R ./etc/nginx /etc/
```

и редактируем файл /etc/nginx/conf.d/xrad.xsquare.conf, внося необходимые изменения vi /etc/nginx/conf.d/xrad.xsquare.conf

```
server {
    listen 8080;
    server_name xrad.xsquare;
    root /var/www/xrad.xsquare;
    index index.html;
    location /ds{
        proxy_pass http://127.0.0.1:8889/ds;
    }
    location / {
        try_files $uri $uri/ =404;
    }
}
```

Для применения новых настроек перезапускаем nginx

```
systemctl restart nginx
systemctl enable nginx
```

проверяем его состояние

```
systemctl --no-pager status nginx
```

4.3 Настройка Apache2 на DEB-based ОС

Копируем из дистрибутива конфигурационные файлы apache2

```
cp -R ./etc/apache2 /etc/
```

и редактируем файл конфигурации VirtualHost /etc/apache2/sites-available/xrad.xsquare.conf, внося необходимые изменения:

```
vi /etc/apache2/sites-available/xrad.xsquare.conf

<VirtualHost *:80>
ServerAdmin info@xsquare.ru
ServerName xrad.xsquare
ServerAlias xrad.xsquare
DocumentRoot /var/www/xrad.xsquare

Alias /files "/var/www/xrad.xsquare.files.local"
<Directory /var/www/xrad.xsquare.files.local>
    Options FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

ProxyPass /ds http://127.0.0.1:8889/ds
ProxyPassReverse /ds http://127.0.0.1:8889/ds

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Применяем новую конфигурацию

```
a2ensite xrad.xsquare.conf
```

и перезапускаем apache2

```
systemctl restart apache2
```

4.4 Настройка Apache2 на RPM-based ОС

Копируем из дистрибутива конфигурационные файлы httpd

```
cp -R ./etc/httpd /etc/
```

и редактируем файл конфигурации VirtualHost /etc/httpd/conf.d/xrad.xsquare.conf, внося необходимые изменения:

```
vi /etc/httpd/conf.d/xrad.xsquare.conf
<VirtualHost *:80>
ServerAdmin info@xsquare.ru
ServerName xrad.xsquare
ServerAlias xrad.xsquare
DocumentRoot /var/www/xrad.xsquare

Alias /files "/var/www/xrad.xsquare.files.local"
<Directory /var/www/xrad.xsquare.files.local>
    Options FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

ProxyPass /ds http://127.0.0.1:8889/ds
ProxyPassReverse /ds http://127.0.0.1:8889/ds

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Отключаем Security-Enhanced Linux для HTTP запросов

```
setsebool -P httpd_can_network_connect 1
```

и перезапускаем apache

```
systemctl restart httpd
```

4.5 Настройка XRAD

Настройка XRAD осуществляется редактированием конфигурационных файлов. Существует два основных файла конфигурации:

- config.json - определяет настройки подключения к базе с метаданными xrad, а также настройки подключения к источникам данных.
- auth.json - определяет опции аутентификации в приложении xrad.

Подробнее смотреть раздел [Конфигурационные файлы](#).

Конфигурационные файлы

5.1 Файл конфигурации config.json

Для работы XRAD необходимо, чтобы в каталоге с исполняемым файлом присутствовал файл конфигурации config.json.

Файл конфигурации содержит 3 раздела: Описатель «app», где можно определить базовые настройки сервера:

```
"app": {
  "port": "8889"
}
```

Атрибут	Тип	Описание
port	Строка	Определяет номер сетевого порта, на котором будет запущен сервер (по умолчанию - 8889)

Описатель «XRAD», где определяются настройки для работы с БД XRAD:

```
"XRAD": {
  "login": "xrad_user",
  "password": "xrad_user",
  "host": "127.0.0.1",
  "port": 5432,
  "minCons": 1,
  "maxCons": 5,
  "dbName": "xraddb"
}
```

Параметр	Тип	Описание
login	Строка	Имя пользователя для подключения к базе данных XRAD.
password	Строка	Пароль пользователя для подключения к базе данных XRAD.
host	Строка	IP-адрес сервера базы данных XRAD.
port	Число	Номер порта, на котором работает сервер базы данных XRAD.
dbName	Строка	Имя базы данных, к которой требуется подключиться XRAD.
minCons	Число	Минимальное количество открытых соединений с базой данных.
maxCons	Число	Максимальное количество одновременных соединений с базой данных.

Описатель «datasources» определяет массив источников данных, используемых сервером приложений.

Например, в следующем блоке определяется несколько источников данных для разных типов подключений:

```
"datasources": [
  {
    "name": "DEFAULT_APP",
    "dbType": "postgresql",
    "login": "app_user",
    "password": "app_user",
    "host": "127.0.0.1",
    "port": 5432,
    "minCons": 1,
    "maxCons": 15,
    "dbName": "pghs",
    "runtimeOptions": {
      "LC_NUMERIC": "ru_RU.UTF-8"
    }
  },
  {
    "name": "ORACLE",
    "dbType": "oracle",
    "login": "admin",
    "password": "admin",
    "host": "127.0.0.1",
    "port": 1521,
    "minCons": 1,
    "maxCons": 15,
    "dbName": "xora",
    "runtimeOptions": {
      "NLS_NUMERIC_CHARACTERS": "'", "'"
    },
    "connOptions": {
      "ssl": "true"
    }
  },
  {
    "name": "MS_SQL",
    "dbType": "mssql",
    "login": "admin",
    "password": "admin",
    "host": "127.0.0.1",
    "port": 1433,
    "minCons": 1,
```

(продолжается на следующей странице)

(продолжение с предыдущей страницы)

```

    "maxCons": 5,
    "dbName": "xdac_db",
    "runtimeOptions": {
      "DATEFORMAT": "dmy"
    },
    "connOptions": {
      "encrypt": "true"
    }
  },
  {
    "name": "MYSQL",
    "dbType": "mysql",
    "login": "admin",
    "password": "admin",
    "host": "127.0.0.1",
    "port": 3306,
    "minCons": 1,
    "maxCons": 15,
    "dbName": "employees",
    "connOptions": {
      "compress": "false"
    }
  },
  {
    "name": "FirebirdDB",
    "login": "admin",
    "password": "admin",
    "host": "127.0.0.1",
    "dbType": "firebird",
    "port": 3050,
    "minCons": 1,
    "maxCons": 5,
    "dbName": "/data/reddb/xadmin.fdb"
  }
]

```

Параметр	Тип	Описание
name	Строка	Имя источника данных. Должно соответствовать имени источника определенному в приложении XRAD.
dbType	Строка	Тип источника данных. Возможные значения: postgresql , mssql , oracle , firebird .
login	Строка	Имя пользователя для подключения к базе данных XRAD.
password	Строка	Пароль пользователя для подключения к базе данных XRAD.
host	Строка	IP-адрес сервера базы данных XRAD.
port	Число	Номер порта, на котором работает сервер базы данных XRAD.
dbName	Строка	Имя базы данных, к которой требуется подключиться XRAD.
minCons	Число	Минимальное количество открытых соединений с базой данных.
maxCons	Число	Максимальное количество одновременных соединений с базой данных.

продолжается на следующей странице

Таблица 3 – продолжение с предыдущей страницы

Параметр	Тип	Описание
runtimeOptions	Объект	Перечень параметров сессии которые должны быть установлены после создания нового подключения к БД. Используется для установки значений через ALTER SESSION.
connOptions	Объект	Перечень параметров используемых для подключения к БД. Используется только для установки дополнительных параметров для подключения к БД типа Oracle и Mysql.

5.2 Файл конфигурации схем аутентификации auth_config.json

XRAD поддерживает аутентификацию и авторизацию с использованием следующих схем:

- Microsoft LDAP,
- Microsoft Kerberos SSO
- LDAP
- Kerberos SSO
- Open ID Connect

При старте сервер разработки XRAD загружает схемы из файла auth_config.json и использует их при аутентификации и авторизации разработчиков.

Файл auth_config.json содержит массив описателей схем аутентификации следующего формата:

```
[
  {
    "name": "",
    "label": "",
    "type": "",
    "enabled": ,
    "order": ,
    "options": {}
  }
]
```

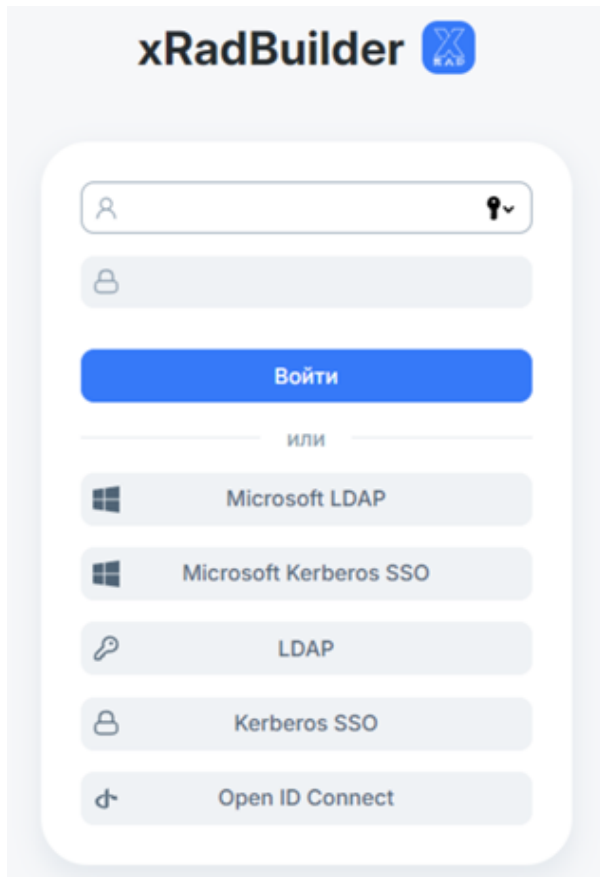
- «name» - строка. Имя схемы аутентификации.
- «label» - строка. Отображаемое на кнопке наименование схемы аутентификации.
- «type» – строка. Тип схемы аутентификации.

Поддерживаемые значения:

- ldap
- kerberos_sso
- microsoft_ldap
- microsoft_kerberos_sso
- oidc
- «enabled» – логическое значение. Данный флаг определяет доступность схемы аутентификации для использования в приложении.
- «order» – число. Порядковый номер в общем списке схем при отображении на странице аутентификации.

- «options» – блок параметров схемы, который зависит от типа схемы.

Загруженные из конфигурационного файла схемы будут использоваться на странице авторизации XRAD в соответствии с флагом `enabled` и порядковым номером `order`.



Рассмотрим блок `options` для различных схем аутентификации.

5.2.1 LDAP, MICROSOFT_LDAP

Для типа схемы `ldap` и `microsoft_ldap` блок «options» описывает параметры подключения к LDAP-серверу, которые используются для аутентификации и получения информации о пользователях.

- «host» – строка. IP-адрес LDAP-сервера.
- «port» – число. Порт, используемый для связи с LDAP-сервером.
- «base» – строка. Определяет базовую точку входа в LDAP-каталог.
- «encryption» – строка. Определяет тип шифрования, используемого для связи с LDAP-сервером. Поддерживаемые значения:
 - `start_tls` – устанавливает защищенное TLS-соединение после начальной аутентификации по незашифрованному каналу. Обычно используется на порту 389.
 - `simple_tls` – устанавливает полностью зашифрованное TLS-соединение с самого начала. Обычно используется на порту 636.
 - `plain` – не использует шифрование и работает по незащищенному каналу. Обычно используется на порту 389.

- «bind_dn» – строка. Distinguished Name (DN) пользователя, используемого для аутентификации на LDAP-сервере.
- «password» – строка. Пароль, связанный с указанным bind_dn.
- «request_user_groups» – логическое значение. Определяет необходимость запроса групп, к которым принадлежит пользователь.

Например:

```
{
  "name": "User Auth LDAP",
  "type": "ldap",
  "options": {
    "host": "10.100.117.229",
    "port": 389,
    "base": "DC=ald,DC=dom",
    "encryption": "start_tls",
    "bind_dn": "uid=ldap,cn=users,cn=accounts,dc=ald,dc=dom",
    "password": "password",
    "request_user_groups": true
  }
}
```

5.2.2 KERBEROS_SSO, MICROSOFT_KERBEROS_SSO

Для типа схемы kerberos_sso и microsoft_kerberos_sso блок «options» описывает параметры подключения к LDAP-серверу, которые используются для аутентификации и получения информации о пользователях.

- «keytab» – строка. Закодированное в BASE64 содержимое keytab файла сгенерированного для сквозной доменной аутентификации.
- «request_user_groups» – логическое значение. Определяет необходимость запроса групп, к которым принадлежит пользователь.

Например:

```
{
  "name": "User Auth Kerberos SSO",
  "type": "kerberos_sso",
  "options": {
    "keytab": "BQIAABVAAIAB0FMRC5ET00ABEhUVFAAEXBnaHM0LWRldi5hbG",
    "request_user_groups": true
  }
},
```

5.2.3 OIDC

Для типа схемы oidc (Open Id Connect) блок «options» описывает конфигурацию аутентификации и авторизации пользователей с использованием OpenID Connect (OIDC).

- «scope»- строка. Список запрашиваемых областей, к которым требуется доступ при аутентификации.
- «issuer»- строка. URL-адрес сервера аутентификации (Issuer).
- «uid_field» - строка. Определяет поле, содержащее уникальный идентификатор пользователя (UID).
- «pkce»- логическое значение. Флаг, указывающий на необходимость использования Proof Key for Code Exchange (PKCE) для повышения безопасности.

- «client_options» - описатель блока параметров клиента, использующего OIDC-аутентификацию.
- «id»- строка. Идентификатор клиента(Client ID).
- «secret» - строка. Секретный ключ клиента(Client Secret).
- «redirect_uri»- строка. URL-адрес, на который будет перенаправлен пользователь после успешной аутентификации.
- «request_user_groups» – логическое значение. Определяет необходимость запроса групп, к которым принадлежит пользователь.

Функциональные характеристики

XRAD - это среда декларативной разработки, которая позволяет создавать и развертывать веб-приложения с ориентацией на базы данных. XRAD обеспечивает разработчика всеми инструментами для построения приложения в единой, расширяемой платформе на основе сервера баз данных PostgreSQL.

6.1 Что такое XRAD?

В современном мире для разработки веб-приложения часто требуется наличие нескольких разработчиков, каждый из которых отвечал бы за отдельные компоненты системы.

Обычно веб-приложение разделяют на следующие компоненты:

- пользовательский интерфейс (frontend)
- компонент для взаимодействия с базой данных (backend)
- связующий компонент, который отвечает за различную логику приложения (middleware)

Все эти компоненты взаимозависимы друг от друга и без четко определенных границ ответственности могут нарушать логику работы приложения. Данные границы и зоны ответственности определяет архитектор приложения, человек от которого требуется высокий уровень компетенции в понимании работы каждого из компонентов. Не стоит также забывать про администрирование такой системы и поддержание работоспособности, что требует наличия системного администратора.

Наличие такой команды однозначно увеличивает цену и уменьшает скорость разработки приложения.

Платформа XRAD предлагает совершенно другой подход к разработке.

XRAD - это платформа для быстрой разработки приложений, разработанная ООО «Хи-Квадрат» (Xsquare Rapid Application Development).

XRAD - является декларативной средой для разработки и развертывания веб-приложений, ориентированных на базы данных. Благодаря встроенным функциям, таким как темы пользовательского интерфейса, элементы управления навигацией, обработчики форм и гибкие отчеты, XRAD значительно ускоряет процесс разработки приложений.

6.2 Как работает XRAD?

XRAD использует стандартную 3-уровневую архитектуру, в которой запросы отправляются из браузера через сервер приложений в базу данных. Вся обработка, манипулирование данными и бизнес-логика выполняются в базе данных. После того, как база данных обработает код, результаты будут переданы веб-сервером в виде структуры JSON, на основе которой веб-приложение отобразит страницу.

Эта архитектура гарантирует доступ к данным практически с нулевой задержкой, высочайшую производительность и горизонтальную масштабируемость «из коробки».

Данный раздел описывает порядок поддержания работоспособности приложения и порядок загрузки компонентов.

7.1 Сервер приложений

Для загрузки xrad пользователю необходимо убедиться в наличие правильно настроенного конфигурационного файла config.json.

```
cat /usr/local/xsquare.xrad/config.json
```

Команда должна отобразить правильный конфигурационный файл, описанный в разделе “Конфигурационные файлы”. Команда должна отобразить правильный конфигурационный файл, описанный в разделе “Конфигурационные файлы”.

Запуск сервера приложений:

```
systemctl start xsquare.xrad.service
```

Проверка состояния сервера приложений:

```
systemctl status xsquare.xrad.service
```

В случае возникновения ошибок они будут записаны в журнал. Проверить сообщения об ошибках можно выполнив команду:

```
journalctl -u xsquare.xrad.service
```

7.2 База данных

Запуск базы данных

```
systemctl start postgresql
```

Остановка базы данных

```
systemctl stop postgresql
```

Проверка статуса сервера базы данных:

```
systemctl status postgresql
```

Необходимо проверить, что база сервер баз данных прослушивает указанный адрес и порт

```
root@pghsdb:main# cat postgresql.conf | grep listen
#listen_addresses = 'localhost'          # what IP address(es) to listen on;
root@pghsdb:main# cat postgresql.conf | grep port
port = 5432
```

Необходимо проверить, что пользователи имеют доступ подключения по ipv4

```
root@pghsdb:main# cat pg_hba.conf
# IPv4 local connections:
host    xraddb          xrad_user    127.0.0.1/32          md5
host    app_db          app_user    127.0.0.1/32          md5
```

В случае возникновения неполадок с БД необходимо обратиться к руководству администратора выбранного дистрибутива PostgreSQL.