

Утвержден
НБД.00048-01 32 01-ЛУ

**КАРДИОМЕТР-МТ
КАРДИОСЕРВЕР 3.0**

Руководство системного администратора

НБД.00048-01 32 01

Листов 41

СОДЕРЖАНИЕ

1	Требования к аппаратно-программной платформе.....	3
2	Установка Docker Community Edition.....	4
2.1	Пример установки Docker Community Edition для Debian 9.....	5
2.2	Пример установки Docker Community Edition для CentOS 7.....	6
3	Установка Docker Compose.....	7
4	Установка Кардиосервера 3.0.....	7
5	Кардиосервер 3.0.....	9
5.1	Структура файлов.....	9
5.2	Запуск.....	10
5.3	Остановка.....	10
5.4	Перезагрузка.....	11
5.5	Обновление.....	11
5.5.1	Миграция данных с Кардиосервера 3.0 версии до 20170724.....	12
5.6	Настройка.....	12
5.6.1	Конфигурационный файл «.env».....	13
5.6.2	Конфигурационный файл «docker-compose.yml».....	13
5.6.3	Конфигурационный файл «config.json».....	13
5.6.4	Настройка интеграции с МИС по стандарту DICOM.....	14
5.6.5	Настраиваемый модуль интеграции.....	18
5.6.6	Импорт пациентов из МИС.....	30
5.6.7	Синхронизация обследований с МИС.....	36
5.6.8	Интеграция с КМИС «Дамумед».....	36
5.7	Резервное копирование.....	39
5.8	Восстановление из резервной копии.....	39
6	Настройка межсетевого экрана.....	39
7	Клиентское программное обеспечение.....	40
8	Обработка персональных данных.....	41

1 Требования к аппаратно-программной платформе

Кардиосервер 3.0 является серверным программным обеспечением комплекса Кардиометр-МТ. Кардиосервер 3.0 может быть размещён на выделенном сервере или в облачной инфраструктуре (IaaS) со следующими характеристиками:

- процессор с не менее, чем четырьмя вычислительными потоками (ядрами), с разрядностью машинного слова 64 бита;
- объём ОЗУ не менее 4 Гб;
- твердотельный накопитель (SSD) объёмом не менее 120 Гб (одна ЭКГ длительностью 24 секунды занимает в среднем 40 Кб);
- операционная система Linux: Debian 9.8 или новее, CentOS 7 или новее.

Для функционирования Кардиосерверу 3.0 необходимо следующее системное программное обеспечение:

- Docker Community Edition версии 17.03 или новее (www.docker.com);
- Docker Compose версии 1.13.0 или новее (www.docker.com).

Кардиосервер 3.0 рассчитан на эксплуатацию в составе программно-технического комплекса заказчика. Техническая и физическая защита аппаратных компонентов, носителей данных, бесперебойное энергоснабжение, резервирование ресурсов, техническое обслуживание реализуется техническими и организационными средствами, предусмотренными в информационно-технологической инфраструктуре заказчика.

Кардиосервер 3.0 может быть размещён в медицинском учреждении или региональном дата-центре. В обоих случаях на объекте размещения Кардиосервера 3.0 должны быть выполнены следующие требования:

- приняты меры по резервированию аппаратно-программного обеспечения, позволяющие восстановить работоспособность системы после сбоя;
- обеспечен круглосуточный доступ к сети Интернет для приёма входящих запросов и получения результатов автоматической интерпретации электрокардиограмм; доступ в сеть Интернет должен производиться с фиксированным внешним IP-адресом;

- обеспечено круглосуточное стабильное и бесперебойное электропитание;
- сетевое оборудование должно быть корректно настроено для обеспечения достаточного уровня безопасности и подключения клиентского ПО к Кардиосерверу 3.0.

Обязанности по техническому обслуживанию Кардиосервера 3.0 и оборудования, на котором он устанавливается, выполняются системным администратором заказчика, включая следующие работы:

- установка и настройка Кардиосервера 3.0;
- резервирование аппаратно-программного обеспечения;
- резервное копирование и восстановление данных;
- ведение учетных записей пользователей;
- управление правами доступа пользователей;
- настройка клиентского программного обеспечения;
- модернизация, настройка и мониторинг работоспособности технических средств (серверов, рабочих станций);
- диагностика типовых неисправностей;
- замена базовых узлов периферийных устройств, имеющих ограниченный ресурс;
- настройка вычислительной сети;
- контроль доступа к сетевым ресурсам;
- обеспечение электропитания;
- ремонт и замена неисправного оборудования.

2 Установка Docker Community Edition

Кардиосервер 3.0 функционирует на базе технологии изоляции программной среды - контейнеры Docker (www.docker.com). Необходимо использовать Docker CE версии не ниже 17.3. Установку проводить согласно официальной документации, размещённой на сайте docs.docker.com.

Далее приведён пример стандартного процесса установки Docker CE на операционные системы Debian 9 (Stretch) 64-bit и CentOS 7 64-bit.

Также доступен альтернативный вариант установки Docker CE с помощью официального скрипта компании Docker Inc. Скрипт необходимо исполнять с административными правами (root):

```
$ curl -sSL https://get.docker.com/ | sh
```

2.1 Пример установки Docker Community Edition для Debian 9

1) Обновить информацию об актуальных версиях пакетов.

```
$ sudo apt-get update
```

2) Установить пакеты для подключения менеджера пакетов **apt** к репозиторию по протоколу HTTPS.

```
$ sudo apt-get install \  
    apt-transport-https \  
    ca-certificates \  
    curl \  
    gnupg2 \  
    software-properties-common
```

3) Добавить официальный PGP-ключ репозитория Docker.

```
$ curl -fsSL https://download.docker.com/linux/debian/gpg | sudo  
apt-key add -
```

4) Переключить репозиторий на ветку стабильных версий.

```
$ sudo add-apt-repository \  
    "deb [arch=amd64] https://download.docker.com/linux/debian \  
    $(lsb_release -cs) \  
    stable"
```

5) Обновить информацию об актуальных версиях пакетов.

```
$ sudo apt-get update
```

6) Установить актуальную версию Docker.

```
$ sudo apt-get install docker-ce
```

7) Включить автоматическую загрузку службы Docker при старте операционной системы.

```
$ sudo systemctl enable docker
```

8) Запустить службу Docker.

```
$ sudo systemctl start docker
```

9) Добавить пользователя в группу Docker

```
$ sudo usermod -aG docker $USER
```

2.2 Пример установки Docker Community Edition для CentOS 7

1) Установить инструментарий для настройки менеджера пакетов.

```
$ sudo yum install -y yum-utils
```

2) Настроить репозиторий на ветку стабильных версий.

```
$ sudo yum-config-manager \  
  --add-repo \  
  https://download.docker.com/linux/centos/docker-ce.repo
```

3) Обновить информацию об актуальных версиях пакетов.

```
$ sudo yum makecache fast
```

4) Установить актуальную версию Docker.

```
$ sudo yum install docker-ce
```

5) Включить автоматическую загрузку службы Docker при старте операционной системы.

```
$ sudo systemctl enable docker
```

6) Запустить службу Docker.

```
$ sudo systemctl start docker
```

7) Добавить пользователя в группу Docker

```
$ sudo usermod -aG docker $USER
```

3 Установка Docker Compose

Перед установкой Docker Compose необходимо установить Docker Community Edition (см. раздел 2).

Установку Docker Compose проводить согласно официальной документации, размещённой на сайте docs.docker.com/compose/install/. Необходимо использовать Docker Compose версии не ниже 1.13.0.

Пример установки Docker Compose 1.13.0:

1) Загрузить программу.

```
$ curl -L  
https://github.com/docker/compose/releases/download/1.13.0/docker-  
compose-`uname -s`-`uname -m` > /usr/local/bin/docker-compose
```

2) Добавить разрешение на запуск программы.

```
$ sudo chmod +x /usr/local/bin/docker-compose
```

3) Проверить работу docker-compose запросив его версию. Если в ответе получена ошибка, значит необходимо повторить установку Docker Compose проверив правильность набора команды.

```
$ docker-compose --version  
docker-compose version 1.13.0, build e12f3b9
```

4 Установка Кардиосервера 3.0

Перед установкой Кардиосервера 3.0 необходимо установить Docker Community Edition (см. раздел 2), Docker Compose (см. раздел 3) и пакеты программного обеспечения python и python-yaml.

Пример установки дополнительного программного обеспечения для Debian 9.

```
$ sudo apt-get install -y python python-yaml
```

Пример установки дополнительного программного обеспечения для CentOS 7.

```
$ sudo yum install -y python python-yaml
```

Установку Кардиосервера 3.0 проводить в следующем порядке:

1) Подготовить каталог для размещения Кардиосервера 3.0.

```
$ mkdir cardioserver && cd cardioserver
```

2) Загрузить менеджер дистрибутива Кардиосервера 3.0.

```
$ curl --location --remote-name  
http://micard.ru/distrib/cardioserver3
```

3) Добавить разрешение на запуск программы.

```
$ sudo chmod +x cardioserver3
```

4) Поместить в каталог с Кардиосервером 3.0 (смотри шаг 1) файл лицензии. Файл временной или постоянной лицензии можно получить у официальных дистрибьютеров продукции АО «МИКАРД-ЛАНА».

5) Запустить менеджер дистрибутива с ключом install.

```
$ ./cardioserver3 install
```

6) В операционных системах, использующих технологию SELinux, необходимо для файлов дистрибутива установить контекст.

```
$ sudo ./setcontext.sh
```

7) Установить системное время и дату согласно географическому размещению Кардиосервера, обязательно учитывать часовой пояс. Допускается использовать время UTC установив часовой пояс в ноль.

8) Определить в настройках Кардиосервера параметр `update_url` — ссылку высылаемую Кардиосервером клиентскому приложению. По этой ссылке клиентское приложение будет запрашивать для себя обновления. Подробнее о настройке параметра `update_url` смотри раздел 7.

9) Перед первым запуском Кардиосервера 3.0 в конфигурационных файлах `conf.cardioserver/config.json` и `docker.cardioserver/sconfig.json` определить учётную запись для доступа к базе данных в соответствии с требуемой политикой

безопасности.

5 Кардиосервер 3.0

5.1 Структура файлов

№ п/п	Файл / Каталог	Примечание
1	./cardioserver3	Менеджер дистрибутива. Позволяет установить, запустить, остановить Кардиосервер 3.0, а так же выполняет другие сервисные функции.
2	./docker-compose.yml	Конфигурационный файл Docker Compose. Описывает правила взаимодействия между контейнерами Docker и основной операционной системой.
3	./setcontext.sh	Скрипт для поддержки технологии SELinux.
4	./backup.sh	Скрипт резервного копирования базы данных Кардиосервера 3.0 и его дистрибутива.
5	./env	Конфигурационный файл с основными настройками Кардиосервера.
6	./docker.updater/	Каталог с исходными данными для сборки контейнера с механизмом обновления клиентского программного обеспечения комплекса Кардиометр-МТ.
7	./docker.cardioserver/	Каталог с исходными данными для сборки контейнера с Кардиосервером 3.0.
8	./conf.cardioserver/ config.json	Файл настроек Кардиосервера относительно файловой системы контейнера.
9	./conf.mysql/micard.cnf	Файл настроек базы данных Кардиосервера.
10	./conf.web/nginx.conf	Файл настроек web-сервера для механизма

		обновления клиентского программного обеспечения комплекса Кардиометр-МТ.
11	./conf.integration/	Каталог для хранения конфигурационных файлов механизма интеграции с медицинскими информационными системами.
12	./logs/	Каталог для хранения журнала событий Кардиосервера.
13	./mysql/	Каталог размещения файлов служебной базы данных.
14	./storage/cardiobank/	Каталог для хранения базы обследований.
15	./storage/softbank/	Каталог для хранения клиентского программного обеспечения.
16	./storage/distrib/	Каталог для хранения дистрибутивов Кардиосервера 3.0.

5.2 Запуск

Запуск Кардиосервера 3.0 осуществляется выполнением команды в каталоге его размещения:

```
$ ./cardioserver3 start
```

Первый запуск Кардиосервера 3.0 может продолжаться значительное время. При первом запуске и после обновления менеджер дистрибутива заново собирает контейнеры.

5.3 Остановка

Остановка Кардиосервера 3.0 осуществляется выполнением команды в каталоге его размещения:

```
$ ./cardioserver3 stop
```

5.4 Перезагрузка

Перезагрузка Кардиосервера 3.0 осуществляется выполнением команды в каталоге его размещения:

```
$ ./cardioserver3 restart
```

5.5 Обновление

Внимание! Для обновления Кардиосервера 3.0 версии ниже 20170724 необходимо выполнить процесс миграции на новый сервер (смотри пункт 5.5.1).

Перед запуском процесса обновления необходимо произвести резервное копирование файлов Кардиосервера, базы данных и файлов обследований.

Обновление Кардиосервера 3.0, начиная с версии 20170724, осуществляется выполнением следующих команд в каталоге его размещения:

```
$ curl --remote-name http://micard.ru/distrib/cardioserver3
$ chmod +x cardioserver3
$ ./cardioserver3 update
```

Процесс обновления может занять значительное время, при котором Кардиосервер будет функционировать, но не отвечать на запросы клиентских приложений.

Во время обновления запрещено выключать или перезагружать Кардиосервер!

По окончании обновления Кардиосервер выдаст в логах параметры своего запуска:

```
$ docker logs m_cardioserver
wait-for-it.sh: waiting 15 seconds for db_mysql:3306
wait-for-it.sh: db_mysql:3306 is available after 1 seconds
CardioServer work process [20181119]

server version: 20181119
server listen:48004
```

```
cardiobank path: /storage/cardiobank
postfix:      10000
server type:   local
server id:     10000
db pool size:  4
cluster ids:
```

5.5.1 Миграция данных с Кардиосервера 3.0 версии до 20170724

Начиная с версии 20170724 Кардиосервер 3.0 изменил свою структуру. Новая структура не позволяет использовать данные предыдущих серверов. В связи с этим необходимо выполнить переустановку Кардиосервера 3.0 в другой каталог:

- 1) Выполнить шаги 1-4 раздела 4 (Установка Кардиосервера 3.0).
- 2) Запустить менеджер дистрибутива с ключом `install` и параметром `--migration <path_to_old_server>`, где `<path_to_old_server>` указывает на каталог размещения текущей версии Кардиосервера.

```
$ ./cardioserver3 install --migration <path_to_old_server>
```

Пример:

```
$ ./cardioserver3 install --migration /home/micard/micard-lana/
```

- 3) Выполнить шаг 6 раздела 4 (установка Кардиосервера 3.0).

5.6 Настройка

Кардиосервер 3.0 представляет собой набор контейнеров Docker. Контейнеры Docker — это технология изоляции программной среды. Контейнеры изолируют программное обеспечение контейнера от других контейнеров и основной операционной системы.

Настройка параметров Кардиосервера 3.0 осуществляется через конфигурационный файл `./conf.cardioserver/config.json`.

Настройка правил взаимодействия контейнеров и основной операционной

системы осуществляется через конфигурационный файл `./docker-compose.yml`.

В связи с тем, что правила взаимодействия контейнеров могут изменяться от версии к версии Кардиосервера, основные настройки вынесены в отдельный файл `./env`.

5.6.1 Конфигурационный файл «`env`»

В конфигурационном файле `env` хранятся переменные окружения для настройки контейнеров (для файла `docker-compose.yml`):

- `M_STORAGE` — размещение каталога с файлами обследований и дистрибутивов клиентского программного обеспечения;
- `M_CARDIOSERVER_PORT` — сетевой порт, по которому Кардиосервер будет доступен для клиентского программного обеспечения;
- `M_WEB_PORT` — сетевой порт, по которому Кардиосервер будет предоставлять обновления для клиентского программного обеспечения;
- `M_MYSQL_PORT` — сетевой порт для администрирования базы данных Кардиосервера;
- `M_UID` — системный идентификатор пользователя, от имени которого запускаются Docker-контейнеры Кардиосервера;
- `M_GID` — системный идентификатор группы пользователя, от имени которого запускаются Docker-контейнеры Кардиосервера.

5.6.2 Конфигурационный файл «`docker-compose.yml`»

Актуальная справочная информация по структуре файла `docker-compose.yml` размещена на сайте производителя Docker Compose - docs.docker.com/compose/compose-file/.

5.6.3 Конфигурационный файл «`config.json`»

Конфигурационный файл `./conf.cardioserver/config.json` определяет работу Кардиосервера 3.0 и его функциональные возможности.

Группа настроек "проху" заполняется при использовании проху сервера при подключении к сети интернет.

Группа настроек "service_interpretation" содержит параметры подключения к сервису автоматической интерпретации электрокардиограмм.

Группа настроек "software_update" содержит параметр "update_url". Параметр определяет адрес, по которому клиентские приложения загружают обновления и получают информацию по актуальным версиям. Подробнее параметр описан в пункте 7.

После внесения изменений в конфигурационный файл Кардиосервера 3.0 необходимо его перезагрузить.

5.6.4 Настройка интеграции с МИС по стандарту DICOM

Взаимодействие между аппаратурой и медицинскими информационными системами по международным стандартам предполагает использование стандарта DICOM.

Кардиосервер 3.0 может экспортировать результаты обследований в PACS как по назначениям (worklist), так и без них. Экспорт результатов обследований производится только в случаях появления заключения, автоматического и(или) врачебного. В PACS отправляется структурированное текстовое заключение (Structured Report), ЭКГ сигнал в формате DICOM Waveform и PDF-документ с заключением и графиками.

Кардиосервер 3.0 поддерживает следующие сервисы (Service Classes) стандарта DICOM:

- Modality Worklist Management Service (SCU);
- Modality Performed Procedure Step Service (SCU);
- Storage Service (SCU).

Для включения интеграции Кардиосервера 3.0 в группе настроек "integration" конфигурационного файла ./conf.cardioserver/config.json в качестве значения параметра "dicom" необходимо указать true, произвести настройки в группе "dicom" и перезапустить Кардиосервер 3.0. Пример части конфигурационного файла с

подгруппами "dicom" смотрите в конце раздела.

Подгруппа "servers" содержит имена (Application Entity Title), IP адреса и порты серверов.

Подгруппа "services" содержит список сервисов, которые планируется использовать. Доступны следующие сервисы:

- "mpps" – сервис Modality Performed Procedure Step;
- "pacs" – сервис архивации DICOM изображений (Storage Service, далее PACS). На этот сервер будут перенаправляться электрокардиограммы и заключения по ним;
- "wls" – сервис Modality Worklist Management.
- "pms" – сервис поиска пациентов.

Каждый сервис содержит следующие параметры:

- "called_ae_title" – имя сервера из подгруппы "servers", с которым будет производиться взаимодействие;
- "calling_ae_title" – имя Кардиосервера 3.0 в рамках конкретного сервиса. Для сервисов "mpps" и "pacs" в конец имени можно добавить номер устройства съема ("device_number"), либо идентификатор медучреждения ("hfc_id"), либо идентификатор архива ("archive_id"). Сформированное имя не должно превышать 16 символов;
- "redirect" – используется в сервисе "pacs". Определяет типы перенаправляемых изображений. Доступны следующие значения: "sr" (врачебное и автоматическое заключения), "ecg" (изображение электрокардиограммы), pdf (изображение электрокардиограммы в формате pdf).
- "transport_settings" – используется для указания имени и пароля пользователя.

Примечание. Если нет необходимости в использовании одного из сервисов, то соответствующий раздел в конфигурационном файле необходимо удалить. Сервис "mpps" не используется без сервиса "pacs".

Подгруппа "ecg_filters" определяет фильтры, применяемые к электрокардиограммам перед их передачей на PACS. Доступны три вида фильтров:

- "daplf" – фильтр наводки. Позволяет очистить ЭКГ от влияния наводки 50 Гц,

возникающей от расположенных вблизи электроприборов и проводки;

- "drift" – фильтр дрейфа. Фильтр выравнивает изолинию, при применении данного фильтра возможно незначительное изменение формы сегмента ST-T и амплитуды зубца T;
- "tremor" – фильтр тремора. Используется для уменьшения влияния мышечного тремора на ЭКГ.

Применение фильтра уменьшает амплитуду шумов, но может исказить форму QRS-комплекса за счет уменьшения амплитуды сигнала.

Возможные значения фильтров: true (активен) и false (не применяется). Активируемые фильтры рекомендуется согласовать с врачами кардиологами.

Подгруппа "log" содержит параметры ведения журнала событий. Доступны следующие параметры:

- "level" – уровень детализации журнала. Возможные значения: "off", "fatal", "error", "warn", "info", "debug", "trace", "all". Значения перечислены в порядке возрастания "многословности". Рекомендуется использовать уровень детализации не ниже "warn";
- "appender" – определяет устройство вывода журнала событий. Возможные значения: "file" (только файл), "console" (только консоль), "file/console" (одновременно и в файл, и в консоль). По умолчанию используется "file";
- "logs_out_file" – имя файла, куда будет сохраняться журнал событий. Можно использовать как абсолютное, так и относительное имя файла. Если файл не существует, создается новый. Максимальный размер файла 10Мб. После достижения файла данного размера, создается новый файл. Максимальное количество файлов – три, четвертый (который был создан раньше всех) будет удален. Данный параметр не имеет значения, если параметр "appender" равен "console". По умолчанию используется "./logs/dcmlog.log";
- "immediate_flush" – определяет необходимость сброса журнала событий в файл после каждого вывода. Возможные значения: true и false. Рекомендуется использовать false, но в этом случае после аварийного завершения сервера могут быть не сохранены последние события. По умолчанию используется

false;

- "verbose" – определяет необходимость вывода полной информации о ходе работы. Возможные значения: true и false. Рекомендуется использовать false. По умолчанию используется false.

Образец заполнения подгрупп "dicom" конфигурационного файла Кардиосервера 3.0.

```
...
"dicom": {
  "services": {
    "mpps": {
      "calling_ae_title": "MICARD_%device_number",
      "called_ae_title" : "mpps_server"
    },
    "pacs": {
      "calling_ae_title": "MICARD_%device_number",
      "called_ae_title" : "pacs_server",
      "redirect": ["sr", "ecg"],
      "transport_settings": {
        "password": "micard",
        "user": "micard"
      }
    },
  },
  "wls": {
    "calling_ae_title": "MICARD",
    "called_ae_title" : "wls_server"
  },
  "servers": [
    {
      "ae_title": "pacs_server",
      "host": "127.0.0.1",
      "port": 104
    },
  ],
}
```

```
{
  "ae_title": "mpps_server",
  "host": "127.0.0.1",
  "port": 11112
},
{
  "ae_title": "wls_server",
  "host": "127.0.0.1",
  "port": 104
} ],
"ecg_filters": {
  "daplf": false,
  "drift": false,
  "tremor": false
},
"log": {
  "level": "error"
}
},
...
"integration": {
  "dicom": true
}, ...
```

5.6.5 Настраиваемый модуль интеграции

Кардиосервер 3.0 поддерживает передачу в МИС ЭКГ, врачебных и автоматических заключений (далее объекты) в форматах XML, JSON и сообщений с составным типом содержимого (multipart/form-data). Перечень передаваемых атрибутов и их представления настраиваются в конфигурационных файлах для каждого передаваемого объекта отдельно. Для передачи в МИС результатов обследований используются следующие методы.

- Сетевой протокол HTTP (метод POST). Поддерживается передача обследований в форматах XML, JSON и обследований с составным типом

содержимого.

- Сетевой протокол AMQP. Поддерживается передача обследований в форматах XML, JSON.

Для включения интеграции Кардиосервера 3.0 в группе настроек "integration" конфигурационного файла `./conf.cardioserver/config.json` необходимо добавить параметр "mpost", указать значение true и перезапустить Кардиосервер 3.0. В конфигурационном файле `./conf.cardioserver/config.json` появится группа настроек "mpost". Произвести настройки в группе "mpost" и перезапустить Кардиосервер 3.0. После перезапуска в корневом каталоге Кардиосервера 3.0 в директории "conf.integration" будут созданы конфигурационные файлы, определяющие содержимое передаваемых объектов. Произвести настройки и перезапустить Кардиосервер 3.0 еще раз. Пример части конфигурационного файла приведен в конце раздела.

Настройка параметров интеграционного модуля (файл `./conf.cardioserver/config.json`, группа "mpost").

Подгруппа "servers" содержит имена, IP адреса и порты серверов.

Подгруппа "local" содержит следующие параметры:

- "charset" – кодировка, используемая для конструирования объектов. Доступные кодировки: "utf-8", "Windows-1250" до "Windows-1258", "ISO 8859-1" до "ISO 8859-10". По умолчанию используется "ISO_IR 192", что соответствует кодировке "utf-8". Значение "charset" применяется при отправке HTTP запроса в качестве значения параметра HTTP заголовка (Content-Type: text/xml; charset=utf-8).
- "ae_title" – имя узла Кардиосервера 3.0. В текущей версии не используется.
- "language" – язык, используемый для построения протоколов описания ЭКГ. Возможные значения: "ru" (русский) и "en" (английский). По умолчанию используется "en".

Подгруппа "log" содержит параметры ведения журнала событий. Доступны следующие параметры:

- "level" – уровень детализации журнала. Возможные значения: "fatal", "error",

"warn", "info", "debug", "trace". Значения перечислены в порядке возрастания "многословности". Рекомендуется использовать уровень детализации не ниже "warn";

- "appender" – определяет устройство вывода журнала событий. Возможные значения: "file" (только файл), "console" (вывод в стандартный поток), "file/console" (одновременно и в файл, и в стандартный поток). По умолчанию используется "file";
- "logs_out_file" – имя файла, куда будет сохраняться журнал событий. Можно использовать как абсолютное, так и относительное имя файла. Если файл не существует, создается новый. Данный параметр не имеет значения, если параметр "appender" равен "console". По умолчанию используется "./.logs/mpost.log";
- "immediate_flush" – определяет необходимость сброса журнала событий в файл после каждого вывода. Возможные значения: true и false. Рекомендуется использовать false, но в этом случае после аварийного завершения сервера могут быть не сохранены последние события. По умолчанию используется false;
- "verbose" – определяет необходимость вывода полной информации о ходе работы. Возможные значения: true и false. Рекомендуется использовать false. По умолчанию используется false.

Подгруппа "services" содержит список сервисов, которые планируется использовать. Доступен один сервис: "pacs" – сервис передачи в МИС ЭКГ, врачебных и автоматических заключений.

Сервис "pacs" содержит следующие параметры.

- "called_ae_title" – имя сервера из подгруппы "servers", с которым будет производиться взаимодействие;
- "calling_ae_title" – имя Кардиосервера 3.0 для конкретного сервиса. В текущей версии не используется.
- "redirect" – определяет перечень событий, по которым Кардиосервер будет формировать передаваемые объекты. Для каждого объекта в каталоге

"conf.integration" создается конфигурационный файл, определяющий содержимое передаваемого объекта ("multipart_report", "multipart_auto_report", "multipart_exam" – объекты с составным типом содержимого, используется только с протоколом http; "xml_report", "xml_auto_report", "xml_exam" – объекты в формате xml; "json_report", "json_auto_report", "json_exam" – объекты в формате json; постфиксы report, auto_report, exam определяют типы передаваемых объектов: врачебное заключение, автоматическая интерпретация, ЭКГ).

- "transport" – определяет протокол передачи данных, поддерживаются протоколы "http" (метод POST), "amqp".
- "transport_settings" – перечень настроек зависит от указанного протокола передачи в "transport".

Если для передачи используется протокол http, подгруппа "transport_settings" содержит следующие настройки:

- "path" – определяет URL-путь сервиса, куда будут перенаправляться обследования.

Для приведенных ниже настроек будет использоваться URL `http://some_host:8080/some_path`.

```
"servers": [  
  {  
    "ae_title": "pacs_server",  
    "host": "some_host",  
    "port": 8080  
  }  
],  
...  
"services": {  
  ...  
  "called_ae_title": "pacs_server",  
  "transport_settings": {  
    "path": "some_path"  }  
}
```

```

    }
  }
  ...

```

Если для передачи используется протокол amqp, подгруппа "transport_settings" содержит следующие настройки:

- "exchange" – название точки обмена;
- "routing_key" – название ключа маршрутизации;
- "user" – имя пользователя;
- "password" – пароль пользователя;
- "vhost" – название виртуального хоста;
- "exchange_options" – параметры соединения с точкой обмена, доступны "durable", "passive", "no-wait";
- "exchange_type" – тип точки обмена, доступны "direct", "fanout";
- "timeout" – максимальное время ожидания отправки обследования, если отправить обследование не удастся, передача производится позже.

Передаваемые объекты содержат данные о пациенте (таблица 1), обследовании (таблица 2), данные врача, зарегистрировавшего ЭКГ и данные медучреждения (таблица 3), данные врачебного заключения (таблица 4).

В таблицах поле "требование" определяет ограничения на обязательность поля: Н – поле необязательное (значение может быть пустым), О – поле обязательное (поле обязательно будет включено в ответ и содержимое не будет пустым).

Таблица 1 – Пациент

Название поля	Треб.	Тип	Комментарий
patient_exam.patient_number	О	строка	идентификатор
patient_exam.patient_sex	О	перечисление	«male», либо «female», либо «other»
patient_exam.patient_birthday	О	дата-время	дата рождения
patient_exam.patient_family_name	Н	строка	фамилия
patient_exam.patient_personal_name	Н	строка	имя

patient_exam.patient_patronymic	Н	строка	отчество
patient_exam.patient_passport	Н	строка	серия и номер паспорта
patient_exam.patient_insurance	Н	строка	номер страхового полиса
patient_exam.patient_snils	Н	строка	номер СНИЛС
patient_exam.patient_address	Н	строка	адрес проживания
patient_exam.patient_phone	Н	строка	номер телефона

Таблица 2 – Обследование

Название поля	Треб.	Тип	Комментарий
patient_exam.id	О	целое (64 бита)	идентификатор
patient_exam.exam_method	О	перечисление	методика обследования, всегда "EcgAtRest"
patient_exam.exam_type	О	строка	тип обследования в рамках методики: "Стандартный", "С конечностей", "По Небу", "По Кабрера"
patient_exam.exam_age	О	целое (32 бита)	возраст пациента на момент обследования (полных лет)
patient_exam.exam_height	Н	целое (32 бита)	рост пациента на момент обследования в см.
patient_exam.exam_weight	Н	целое (32 бита)	вес пациента на момент обследования в кг.
patient_exam.exam_result_report	Н	строка	текст автоматического заключения

Таблица 3 – Контекст регистрации ЭКГ

Название поля	Треб.	Тип	Комментарий
patient_exam.archive_id	О	целое (64 бита)	идентификатор архива, куда была произведена регистрация ЭКГ
patient_exam.archive_name	О	строка	название архива, куда была произведена регистрация ЭКГ

patient_exam.hcf_id	О	целое (64 бита)	идентификатор медучреждения, где была произведена регистрация ЭКГ
patient_exam.hcf_name	О	строка	название медучреждения, где была произведена регистрация ЭКГ
patient_exam.recorder_id	О	целое (64 бита)	идентификатор врача, зарегистрировавшего ЭКГ
patient_exam.recorder_hcf_id	О	целое (64 бита)	идентификатор медучреждения, где зарегистрирован врач, который произвел съем ЭКГ
patient_exam.recorder_login	О	строка	идентификатор врача, зарегистрировавшего ЭКГ
patient_exam.recorder_family_name	Н	строка	фамилия врача, зарегистрировавшего ЭКГ
patient_exam.recorder_personal_name	Н	строка	имя врача, зарегистрировавшего ЭКГ
patient_exam.recorder_patronymic	Н	строка	отчество врача, зарегистрировавшего ЭКГ
patient_exam.exam_device_number	О	строка	идентификатор устройства съема
patient_exam.exam_start_datetime	О	дата-время	дата и время начала регистрации ЭКГ
patient_exam.exam_datetime	О	дата-время	дата и время окончания регистрации ЭКГ
patient_exam.exam_geolocation	Н	строка	широта и долгота места регистрации ЭКГ, в текущей версии не используется
patient_exam.exam_oktmo	Н	строка	номер ОКТМО, где была зарегистрирована ЭКГ
patient_exam.comment	Н	строка	комментарий врача

patient_exam.accession_number	Н	строка	номер назначения, используется, если ЭКГ зарегистрирована по назначению
patient_exam.sender_hcf_id	О	целое (64 бита)	идентификатор медучреждения, приславшего обследование на консультацию

Таблица 4 – Врачебное заключение

Название поля	Треб.	Тип	Комментарий
conclusion.id	О	целое (64 бита)	идентификатор назначения
conclusion.report	О	строка	текст заключения врача
conclusion.datetime	О	дата-время	дата и время написания заключения
account.id	О	целое (64 бита)	идентификатор врача, написавшего заключение
account.login	О	строка	идентификатор врача, написавшего заключение
account.family_name	Н	строка	фамилия врача, написавшего заключение
account.personal_name	Н	строка	имя врача, написавшего заключение
account.patronymic	Н	строка	отчество врача, написавшего заключение
account.email	Н	строка	адрес электронной почты врача, написавшего заключение

В таблице 5 приведены структуры объектов ЭКГ, автоматического и врачебного заключения.

Таблица 5 – Структуры ЭКГ, автоматического и врачебного заключения

Название поля	Треб.	Тип	Комментарий
---------------	-------	-----	-------------

pdf либо jpeg	Н	изображение (строка либо двоичные данные)	изображение ЭКГ в формате pdf либо jpeg, для передачи в структурах xml и json изображение преобразуется в строковое представление (base64), в случае передачи в сообщении с составным содержимым возможна передача в строковом и двоичном представлениях
document_root	О	строка	определяет тип передаваемого объекта (ЭКГ, автоматическое заключение, врачебное заключение), и может быть использовано на принимающей стороне для правильной интерпретации полученного объекта, в случае xml определяет название документа и корневого элемента, в остальных случаях в структуру объекта добавляется поле document_root с предопределенным значением из конфигурационного файла.
пациент (таблица 1)	Н		
обследование (таблица 2)	Н		
контекст регистрации ЭКГ (таблица 3)	Н		
врачебное заключение	Н		

(таблица 4)			
-------------	--	--	--

В конфигурационных файлах (формат json), описывающих содержимое передаваемого объекта, определяются перечень полей, их представления (для полей с типами дата-время и изображение), названия, используемые при передаче. Для описания поля используется следующая структура.

...

```
"название поля из таблиц 1, 2, 3, 4, 5":{  
  "name": "название поля, используемое при передачи в МИС",  
  "type": "тип поля",  
  "format": "представление"  
}
```

...

Параметр представления "format" доступен для полей с типами изображение и дата-время. Для изображения доступно "base64" и "raw", по умолчанию используется "raw" (двоичное представление). Для полей с типом дата-время применяются специальные символы, определяющие представление: %Y – год (четыре символа), %m – месяц (два символа), %d – день месяца (два символа), %H – час (два символа), %i – минута (два символа), %s – секунда (два символа).

Пример настройки интеграции.

Шаг первый. Установка поля "mpost" (файл ./conf.cardioserver/config.json) в значение true и перезапуск Кардиосервер 3.0. После перезапуска в файле ./conf.cardioserver/config.json появится группа настроек "mpost".

...

```
"integration": {  
  "dicom": false,  
  "mpost": true  
},
```

...

Шаг второй. Настройка Кардиосервера 3.0 на передачу результатов обследований в МИС (файл ./conf.cardioserver/config.json). Ниже приведен пример

настройки. На сетевой сервис по адресу http://192.168.0.2:8080/api/micard_study.php будут перенаправляться объекты "json_report" (врачебное заключение в формате json) и "xml_auto_report" (автоматическое заключение в формате xml). После настройки необходимо перезапустить Кардиосервер 3.0.

```
"mpost": {
  "local": {
    "ae_title": "MICARD",
    "charset": "UTF-8",
    "language": "ru"
  },
  "log": {
    "level": "warn"
  },
  "servers": [
    {
      "ae_title": "pacs_server",
      "host": "192.168.0.2",
      "port": 8080
    }
  ],
  "services": {
    "pacs": {
      "called_ae_title": "pacs_server",
      "calling_ae_title": "MICARD",
      "redirect": [
        "json_report",
        "xml_auto_report"
      ],
      "transport": "http",
      "transport_settings": {
        "path": "api/micard_study.php"
      }
    }
  }
}
```

```
}, ...
```

Шаг третий. Настройка полей в передаваемых объектах. Конфигурационные файлы располагаются в директории `conf.integration`. Ниже приведен пример настройки объекта `"xml_auto_report"` (файл `"mpost_xml_auto_report.json"`). После настройки необходимо перезапустить Кардиосервер 3.0.

```
{
  "patient_exam.exam_datetime": {
    "format": "%Y%m%d-%H%i%s",
    "name": "exam_datetime",
    "type": "datetime"
  },
  "patient_exam.exam_result_report": {
    "name": "report",
    "type": "plaintext"
  },
  "patient_exam.id": {
    "name": "exam_id",
    "type": "plaintext"
  },
  "patient_exam.patient_number": {
    "name": "patient_id",
    "type": "plaintext"
  },
  "pdf": {
    "name": "pdf_image",
    "type": "binary",
    "format": "base64"
  },
  "document_root": {
    "name": "auto_report",
    "type": "plaintext"
  }
}
```

Ниже приведен пример объекта (`"xml_auto_report"`), построение которого

производится согласно произведенным настройкам.

```
<?xml version="1.0" encoding="UTF-8"?>
<auto_report>
  <report>
    Синусовый водитель ритма.
    Нормосистолия, ритм ригидный, ЧСС = 60 уд/мин.
    Нормальное положение электрической оси сердца.
    Внутрижелудочковая блокада (QRS = 134мс).
  </report>
  <exam_datetime>20170706-172919</exam_datetime>
  <exam_id>254214819337632</exam_id>
  <patient_id>id_ivanov</patient_id>
  <pdf_image>JVBERi0xLjQK ... </pdf_image>
</auto_report>
```

5.6.6 Импорт пациентов из МИС

Кардиосервер 3.0 поддерживает импорт карточек пациентов из МИС в формате XML. Для передачи по сети используется протокол HTTP (метод POST) версии 1.1 и ниже.

Для включения интеграции Кардиосервера 3.0 в группе настроек "integration" конфигурационного файла ./conf.cardioserver/config.json необходимо добавить параметр "xml", указать значение true и перезапустить Кардиосервер 3.0. В конфигурационном файле ./conf.cardioserver/config.json появится группа настроек "xml". Произвести настройки в группе "xml" и перезапустить Кардиосервер 3.0 еще раз. Пример части конфигурационного файла приведен в конце раздела.

Подгруппы "servers", "local", "log", "services" имеют параметры, описанные в 5.6.5. Подгруппа "services" содержит список сервисов, которые планируется использовать. Доступен один сервис: "pms".

Логическая структура xml документа запроса карточек пациентов имеет следующий вид:

```
<?xml version="1.0" encoding="utf-8"?>
<patient_management>
```

```
<filtr>
  <limit>50</limit>
  <patient_family_name>Иванов</patient_family_name>
  <patient_id>id_ivanov</patient_id>
  <patient_birthday>19820602-</patient_birthday>
  <patient_sex>male</patient_sex>
</filtr>
<action>get_patient_rq</action>
</patient_management>
```

Если какой-либо фильтр не используется, то он не будет включен в элемент <filtr>. Далее приведен запрос без фильтров:

```
<?xml version="1.0" encoding="utf-8"?>
<patient_management>
  <filtr>
    <limit>50</limit>
  </filtr>
  <action>get_patient_rq</action>
</patient_management>
```

В таблице 6 приведены описания полей.

В качестве типов данных дата и время используется формат ISO 8601:2004 (YYYYMMDD и HHMMSS).

Таблица 6 – Описание полей запроса карточки пациента

Название поля	Тип	Комментарий
action	строка	всегда значение "get_patient_rq"
filtr	контейнер	элемент, содержащий фильтры
> patient_family_name	строка	фамилия пациента. Строка дублирует то, что ввел пользователь
> patient_id	строка	id пациента (строка). В качестве id пациента применяется номер карточки, номер истории болезни, номер полиса ОМС или номер СНИЛС в зависимости от

		принятого в медучреждении идентификатора пациентов.
> patient_birthday	дата	«19820602-» : с 19820602 до настоящего времени) «-19820602» : до 19820602) «19820602-20000602» : с 19820602 до 20000602
> patient_sex	перечисление	«male», либо «female», либо «other»
> limit	целое	максимальное количество карточек, которые клиент готов принять, в запросе всегда присутствует

Примечание. Правила использования шаблонов поиска (wildcard символов) для полей patient_family_name, patient_id определяются системой, формирующей ответ на запрос.

Ответ на запрос может включать 0-п пациентов. Элемент result содержит список найденных карточек пациентов.

Пример ответа, в случае если найден один пациент:

```
<?xml version="1.0" encoding="utf-8"?>
<patient_management>
  <result>
    <patient>
      <patient_family_name>Иванов</patient_family_name>
      <patient_personal_name>Иван</patient_personal_name>
      <patient_patronymic>Иванович</patient_patronymic>
      <patient_id>id_ivanov</patient_id>
      <patient_birthday>19820602</patient_birthday>
      <patient_sex>male</patient_sex>
      <patient_insurance>234234234234223434</
patient_insurance>
      <patient_snils>12345678901234</patient_snils>
      <patient_phone>+7(911)-111-11-11</patient_phone>
```



```
<patient_address>г. Санкт-Петербург, Фуражный  
переулок д.3</patient_address>  
</patient>
```

```
</result>
```

```
<action>get_patient_rsp</action>
```

```
</patient_management>
```

Пример ответа, в случае если найдено два пациента:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<patient_management>
```

```
<result>
```

```
<patient>
```

```
<patient_family_name>Иванов</patient_family_name>
```

```
<patient_personal_name>Иван</patient_personal_name>
```

```
<patient_patronymic>Иванович</patient_patronymic>
```

```
<patient_id>id_ivanov</patient_id>
```

```
<patient_birthday>19820602</patient_birthday>
```

```
<patient_sex>male</patient_sex>
```

```
<patient_insurance>234234234234223434</
```

```
patient_insurance>
```

```
<patient_snils>12345678901234</patient_snils>
```

```
<patient_phone>+7(911)-111-11-11</patient_phone>
```

```
<patient_address>г. Санкт-Петербург, Фуражный  
переулок д.3</patient_address>
```

```
</patient>
```

```
<patient>
```

```
<patient_family_name>Иванов</patient_family_name>
```

```
<patient_personal_name>Константин</  
patient_personal_name>
```

```
<patient_patronymic>Иванович</patient_patronymic>
```

```
<patient_id>id_ivanov_2</patient_id>
```

```
<patient_birthday>19890101</patient_birthday>
```

```
<patient_sex>male</patient_sex>
```

```
<patient_insurance>134234234234223434</
```

```
patient_insurance>
```

НБИД.00048-01 32 01

```
<patient_snils>22345678901234</patient_snils>
<patient_phone>+7(911)-222-22-22</patient_phone>
<patient_address>г. Казань, ул. Пушкина
д.3</patient_address>
</patient>
</result>
<action>get_patient_rsp</action>
</patient_management>
```

Пример ответа, в случае если не найдено ни одного пациента (можно поле result не включать в ответ):

```
<?xml version="1.0" encoding="utf-8"?>
<patient_management>
  <result/>
  <action>get_patient_rsp</action>
</patient_management>
```

В таблице 7 приведены описания полей. Поле "требование" определяет ограничения на обязательность поля: Н — поле необязательное (значение может быть пустым или отсутствовать), О — поле обязательное (поле должно быть включено в ответ и содержимое не должно быть пустым).

Таблица 7 – Описание полей ответа на запрос карточки пациента.

Название поля	Треб.	Тип	Комментарий
result	Н	контейнер	элемент содержит список найденных пациентов
> patient	Н	контейнер	элемент содержит атрибуты карточки одного найденного пациента
>> patient_family_name	Н	строка	фамилия пациента
>> patient_personal_name	Н	строка	имя пациента
>> patient_patronymic	Н	строка	отчество пациента
>> patient_id	О	строка	идентификатор пациента
>> patient_birthday	О	дата	дата рождения пациента

>> patient_sex	О	перечислени е	«male», либо «female», либо «other»
>> patient_insurance	Н	строка	номер страхового полиса
>> patient_snils	Н	строка	номер СНИЛС
>> patient_phone	Н	строка	телефонный номер пациента
>> patient_address	Н	строка	адрес проживания пациента

Образец заполнения подгрупп "xml" конфигурационного файла Кардиосервера 3.0.

```
...
"xml": {
    "local": {
        "ae_title": "MICARD",
        "charset": "ISO_IR 192",
        "language": "ru"
    },
    "log": {
        "level": "warn"
    },
    "servers": [
        {
            "ae_title": "pms_server",
            "host": "127.0.0.1",
            "port": 8080
        }
    ],
    "services": {
        "pms": {
            "called_ae_title": "pms_server",
            "calling_ae_title": "MICARD",
            "transport_settings": {
                "path": "/api/patient_get_api.php"
            }
        }
    }
}
```

```
    }  
  }  
}
```

5.6.7 Синхронизация обследований с МИС

Настройка синхронизации старых обследований производится в конфигурационном файле (./conf.cardioserver/config.json) Кардиосервера 3.0. До синхронизации необходимо произвести настройку интеграции в соответствии с выбранной технологией и проверить правильность настроек. Перечень синхронизируемых обследований задается интервалом времени в группе настроек "integration". Пример приведен ниже.

```
...  
"integration": {  
  "dicom": true,  
  "sync": {  
    "start_date_time": "2010-01-01 00:00:00",  
    "end_date_time": "2017-07-07 23:59:59"  
  }  
}  
...  
...
```

Поля "start_date_time" и "end_date_time" определяют начало и конец интервала времени для синхронизируемых обследований и задаются в формате YYYY-MM-DD HH:MM:SS. После настроек необходимо перезапустить Кардиосервер 3.0. После перезапуска Кардиосервер 3.0 в локальной памяти сформирует список синхронизируемых обследований и автоматически удалит подгруппу "sync". Обследования, попавшие в список, будут переданы Кардиосервером 3.0 в МИС.

5.6.8 Интеграция с КМИС «Дамумед»

Программное обеспечение комплекса Кардиометр-МТ может быть интегрировано с комплексной медицинской информационной системой «Дамумед» (КМИС), разработанной ТОО "Центр Информационных Технологий «ДАМУ».

Интеграционное решение позволяет обмениваться КМИС и Кардиосерверу 3.0 назначениями на медицинские услуги и результатами их исполнения.

Интеграционное решение реализовано на уровне серверного программного обеспечения и проходит в следующем порядке:

1) Пользователи программы РМВ 3.0 получают назначения на оказание услуг и отправляют результаты в КМИС через Локальный Кардиосервер.

2) Со стороны КМИС предоставляется доступ к её сервисам через URL и логин/пароль.

3) Локальный Кардиосервер получает доступ к сервисам КМИС после авторизации.

4) При авторизации Локальный Кардиосервер получает от КМИС token, который используется при последующих запросах.

5) Администратор медицинской организации в программе РМВ 3.0 активирует функцию получения списка доступных регионов.

6) Локальный Кардиосервер запрашивает у КМИС перечень регионов.

7) Администратор в программе РМВ 3.0 выбирает регион.

8) Локальный Кардиосервер для выбранного региона запрашивает у КМИС перечень медицинских организаций.

9) Администратор в программе РМВ 3.0 выбирает медицинскую организацию.

10) Локальный Кардиосервер для выбранной медицинской организации запрашивает у КМИС перечень врачей.

11) Администратор в программе РМВ 3.0 выбирает врачей, для которых будут заведены учётные записи в комплексе «Кардиометр-МТ».

12) Врач авторизуется в программе РМВ 3.0.

13) Врач в программе РМВ 3.0 активирует функцию запроса назначений.

14) Локальный Кардиосервер запрашивает у КМИС перечень доступных назначений.

15) Врач в программе РМВ 3.0 выбирает назначение и исполняет его — регистрирует ЭКГ пациента.

16) Результаты сохраняются в Локальном Кардиосервере.

17) Когда врач в программе РМВ 3.0 сохранит врачебное заключение по ЭКГ, Локальный Кардиосервер отправит в КМИС обследование и врачебное заключение.

18) Передача результатов обследования из комплекса «Кардиометр-МТ» в КМИС возможна только при наличии назначения, врачебного заключения, а так же врача и медицинского учреждения, взятых из справочника КМИС.

Настройка интеграции производится в конфигурационном файле Кардиосервера 3.0 ./conf.cardioserver/config.json:

- В раздел integration добавить поле "external_id" — настройка позволит Кардиосерверу 3.0 получить из КМИС перечни регионов, медицинских организаций и их сотрудников.

- В раздел "integration" добавить подраздел "damumed" со следующими полями:

- "url" — адрес и порт размещения сервера КМИС;
- "login" — учётная запись для доступа сторонних информационных систем;
- "password" — пароль к данной учётной записи.

Пример:

...

```
"integration": {  
    "dicom": false,  
    "external_id": true,  
    "damumed": {  
        "login": "xYxYx",  
        "password": "*****",  
        "url": "http://127.0.0.1:24042"  
    }  
},
```

...

5.7 Резервное копирование

Для повышения сохранности данных рекомендуется периодически делать

резервные копии файлов контейнера базы данных Кардиосервера 3.0 и его конфигурации. Хранить резервные копии необходимо на внешнем носителе информации. Резервная копия используется для восстановления работы Кардиосервера 3.0 после сбоя.

В составе дистрибутива присутствует скрипт `./backup.sh`, сохраняющий конфигурацию Кардиосервера 3.0 и файлы контейнера базы данных в каталог `./backup/`.

Резервное копирование файлов обследований, по умолчанию находящихся в каталоге `./storage/cardiobank/`, необходимо проводить вручную.

5.8 Восстановление из резервной копии

Резервные копии конфигурации Кардиосервера и файлов контейнера базы данных хранятся в каталоге `./backup/` в виде архивов `tar.gz`.

Для восстановления конфигурации Кардиосервера 3.0 и файлов контейнера базы данных из ранее созданной резервной копии необходимо распаковать архив:

```
$ tar -xzf backup/<имя архива>.tar.gz
```

Для восстановления файлов обследований необходимо записать в каталог `./storage/cardiobank/` файлы обследований из резервной копии.

В процессе восстановления Кардиосервер 3.0 должен быть остановлен.

6 Настройка межсетевого экрана

Кардиосервер 3.0 для принятия входящих соединений по умолчанию использует порт TCP:48004. Порт может быть переопределён в конфигурационном файле `.env` (смотри раздел 5.6.1).

Кардиосервер 3.0 использует исходящие соединения:

- 5.188.115.175, порт TCP:48044 — для получения результатов автоматической интерпретации обследований;
- 37.18.27.69, порт TCP:48044 — для получения результатов автоматической интерпретации обследований;

- www.micard.ru (77.222.61.135), порт TCP:80 или TCP:443 — для получения обновлений клиентских приложений.

Взаимодействие Кардиосервера 3.0 с клиентскими приложениями осуществляется по двоичному протоколу. Применение средств, модифицирующих трафик между ними, приведёт к сбоям в работе комплексов Кардиометр-МТ.

7 Клиентское программное обеспечение

В Кардиосервер 3.0 встроен механизм обновления клиентского программного обеспечения комплекса Кардиометр-МТ.

После установки и запуска Кардиосервер 3.0 загружает с сайта www.micard.ru дистрибутивы клиентского программного обеспечения и размещает их в специальном каталоге. По умолчанию это `./storage/softbank/`. Для указания иного расположения дистрибутивов необходимо изменить параметр `M_STORAGE` в файле `«.env»` (смотри раздел 5.6.1).

В конфигурационном файле `./conf.cardioserver/config.json` раздел `software_update` определяет параметр `update_url` — ссылка высылаемая Кардиосервером клиентскому приложению. По этой ссылке клиентское приложение будет запрашивать для себя обновления.

По умолчанию `update_url` указывает на архив дистрибутивов АО «МИКАРД-ЛАНА».

Для настройки обновления клиентского программного обеспечения через Кардиосервер 3.0 необходимо следующее:

- 1) Организовать доступ к Кардиосерверу 3.0 по доменному имени. Допускается вместо доменного имени использовать внешний статический IP-адрес;
- 2) В конфигурационном файле `./conf.cardioserver/config.json` изменить поле `update_url` раздела `software_update`. В поле `update_url` внести доменное имя медицинского учреждения или внешний статический IP-адрес. Например, <http://soft.med78.ru> или <http://127.0.0.1:80>.

Первоначальную установку клиентского программного обеспечения комплекса Кардиометр-МТ осуществлять из репозитория, адрес которого указан в поле

update_url раздела software_update файла ./conf.cardioserver/config.json. То есть необходимо на клиентском устройстве запустить интернет обозреватель (браузер), перейти по ссылке определённой в update_url и установить соответствующее приложение.

Документация по применению клиентского программного обеспечения комплекса Кардиомерт-МТ размещена в сети интернет по адресу <http://micard.ru/distrib/>.

8 Обработка персональных данных

Кардиосервер 3.0 принимает, хранит и выдаёт по запросу персональные данные пациентов через клиентское программное обеспечение (ПО).

Согласно законодательству Российской Федерации оператор персональных и медицинских данных должен обеспечить должный уровень защиты от несанкционированного доступа к этим данным. Кардиосервер 3.0 не оснащён механизмами защиты каналов связи с клиентским ПО. Организация защищённого канала связи между Кардиосервером 3.0 и клиентским ПО возлагается на оператора персональных данных — медицинское учреждение.

По умолчанию в клиентском ПО отключена возможность передачи персональных данных на Кардиосервер, персональные данные сохраняются только локально. О том, как включить передачу персональных данных на Кардиосервер, можно ознакомиться в документации к клиентскому ПО - <http://micard.ru/distrib/>.

Функция автоматической интерпретации обследований Комплекса «Кардиометр-МТ» выделена в отдельный сервис, размещённый в дата-центре АО «МИКАРД-ЛАНА». Связь между Кардиосервером 3.0 и сервисом автоматической интерпретации осуществляется по открытому каналу связи. На сервис интерпретации отправляются обезличенные данные: сигнал ЭКГ, пол, возраст, рост и вес пациента.