

Установка сервиса рассылок

- [Установка сервиса рассылок](#)
- [Обновление сервиса рассылок](#)
- [Логирование сервиса рассылок](#)
 - [Настройка сервиса рассылок](#)
- [Установка и настройка сервиса push-уведомлений](#)
- [Сервис уведомлений о сгорании бонусов](#)
 - [Настройка отправки уведомлений об изменении баланса](#)



Устанавливается на систему начиная с Ubuntu 14.

Для установки сервера рассылок сначала необходимо обратиться за сертификатом.

После с помощью команды добавить его:

Команда добавления сертификата

```
sudo mkdir -p /etc/docker/certs.d;echo ключ=|base64 --decode|sudo tar -zxf - -C /etc/docker/certs.d/
```

Далее необходимо локально добавить ключ в `cashserver.arix.su`.

Установка сервиса рассылок

1. Для установка сервиса нужно скачать следующий скрипт следующей командой:

```
curl -L http://downloads.artix.su/static/downloads/docker/delivery/install.sh -o install.sh
```

2. Сделать файл исполняемым:

```
chmod u+x install.sh
```

3. Запустить скрипт установки с правами суперпользователя **sudo install.sh**
4. Скрипт установит `docker` со всем окружением, скачает `docker-compose` и запустит скрипт развертывания образов для работы с рассылками и запустит их.

Обновление сервиса рассылок

Для обновления сервиса рассылок необходимо:

1. Перейти в директорию, где находится `docker-compose.yml`
2. Выполнить команду для обновления образов:

```
docker-compose pull
```

3. Выполнить команду для обновления контейнеров:

```
docker-compose up -d
```

Логирование сервиса рассылок

Логи рассылок хранятся в директории `var/lib/docker/volumes/delivery_delivery_logs/`. Запись логов ведется в несколько файлов:


Файл	Описание
delivery-gateway.log	Лог сервиса по отправке сообщений
delivery-service.log	Лог сервиса рассылок. Сервис, в котором формируются сообщения для отправки.

Настройка сервиса рассылок

Для того чтобы сервис мог получать информацию о сгорании бонусов, нужно его сконфигурировать. Всю конфигурацию сервису можно передать через ["Конфигурирование рассылок"](#).

 Обязательно в Artix Loyalty Management должны быть корректные настройки подключения к КС и БС.


После установки сервиса рассылок нужно зайти в Artix Loyalty Management, в разделе "Управление", перейти во вкладку ["Конфигурирование рассылок"](#), далее ["Настройки сервера рассылок"](#).

 Для работы рассылок о начислении и списании бонусов нужно установить сервис [Push-уведомлений](#).

Установка и настройка сервиса push-уведомлений

Данный сервис нужен для работы уведомлений:

- о начислении бонусных баллов;
- о списании бонусных баллов;
- с рассылками.

 Поддержка сервиса push-уведомлений для accounting-сервера начинается с версии сборки 3.0.250.

Сервис запускается только под java8.

Для работы сервиса необходимо в конфигурационном файле `/etc/mongodb.conf` установить параметр:

```
master = true
```

и перезапустить `mongodb`, выполнив команду:

```
sudo service mongodb restart
```

Установка и запуск сервиса:

```
# для установки или обновления выполните команду:
sudo apt-get install accounting-event-notifier
# После установки, нужно произвести корректные настройки в файле application.properties, который должен находиться по пути /opt/accounting-event-notifier/application.properties (про настройки написано ниже)
# После установки корректных настроек, нужно перезапустить сервис командой:
sudo service accounting-event-notifier restart
```

Логирование работы сервиса находятся в файле `/var/log/accounting-event-notifier.log`.

Настройка работы сервиса может быть осуществлена в конфигурационном файле `application.properties`, который помещается в ту же директорию, что и `jar`-файл.

Наименование	Тип данных	Возможные значения	Описание	Примечания
server.port	числовой		Порт, на котором будет поднят rest для управления сервисом	Управление состоянием сервера можно осуществлять через rest-api, который доступен для просмотра по адресу <code>http://<host>:<server.port>/swagger-ui.html</code>

su.artix.accounting.event.notifier.group.duration	числовой		Задержка в секундах, в течении которого будут группироваться операции для почечкового объединения	
su.artix.accounting.event.notifier.mongo.send.event.expired.seconds	числовой		Время в секундах, в течение которого информация об отправленных транзакциях будет храниться	
su.artix.accounting.event.notifier.begin.from.current	логический	<ul style="list-style-type: none"> • true • false 	Если в БД не сохранено время последнего прочитанного события - начинать искать записи с текущего момента: <ul style="list-style-type: none"> • При значении параметра true все события, которые были созданы до первого запуска сервиса будут игнорироваться. • При значении параметра false все события, которые были созданы до первого запуска сервиса, будут сформированы и отправлены на внешний сервис. 	По умолчанию true
su.artix.accounting.event.notifier.type	строковый	<ul style="list-style-type: none"> • ARTIX • TDERA 	Тип сервиса, на который будут отправляться события	
ARTIX	su.artix.accounting.event.notifier.rest.sender.prefix	строковый	http-префикс внешнего сервиса, на который будут отправляться полученные события	К примеру, при указании http:10.10.10.10:80/a/b/c/event/ сервисом будет генерироваться POST запрос по адресу http:10.10.10.10:80/a/b/c/event/send
	su.artix.accounting.event.notifier.rest.sender.user	строковый	Логин пользователя для basic-авторизации на внешнем сервисе	По умолчанию admin
	su.artix.accounting.event.notifier.rest.sender.password	строковый	Пароль пользователя, от имени которого будет производиться basic-авторизация на внешнем сервисе	По умолчанию admin
TDERA	su.artix.accounting.event.notifier.rest.tdera.url	строковый	URL, на который будет отправляться POST-запрос	К примеру, при указании http:10.10.10.10:80/a/b/c/event/ сервисом будет генерироваться POST запрос по адресу http:10.10.10.10:80/a/b/c/event/update
	su.artix.accounting.event.notifier.rest.tdera.auth	строковый	Заголовок в POST-запросе для авторизации	Значение из параметра в неизменном виде будет передано в заголовке Authorization
	su.artix.accounting.event.notifier.rest.tdera.sender	строковый	Название отправителя	По умолчанию artix
su.artix.accounting.event.notifier.data.protocol.version	строковый	VERSION_1	Внутренняя версия протокола между accounting-сервером и данным сервисом push-уведомлений. Не влияет на формат сообщения, посылаемый внешнему сервису	В настоящее время доступна только VERSION_1.
su.artix.accounting.event.notifier.data.mongodb.collection.name	строковый		Наименование коллекции, из которой будут извлекаться данные о событиях	
spring.data.mongodb.uri	строковый		Доступ до orlog MongoDB, с которой требуется генерировать события	<ul style="list-style-type: none"> • При размещении сервиса на одной машине с бонусным сервером используется mongodb://localhost/local • При размещении сервиса на отдельной машине необходимо указать адрес до orlog MongoDB
su.artix.accounting.event.notifier.retry.count	числовой		Количество ошибок, по достижении которого останавливается обработка ошибок.	По умолчанию 50

Пример настройки

```
server.port=8090
#####
# Задержка в миллисекундах, в течении которого будут группироваться операции для почечкового объединения
su.artix.accounting.event.notifier.group.duration=10000
# Время (в секундах) - сколько будет храниться информация о посланных транзакциях.
# Чем больше значение - тем больше времени может простаивать "принимающий" события сервер.
# С другой стороны, чем больше это значение, тем большая нагрузка будет на СУБД. Параметр нужно подбирать эмпирически
su.artix.accounting.event.notifier.mongo.send.event.expired.seconds=3600
# Если в БД не сохранено время последнего прочитанного события - начинать искать записи с текущего момента
su.artix.accounting.event.notifier.begin.from.current=true
#####
# Тип сервиса, на который будут слаться события: [ARTIX,TDERA]
su.artix.accounting.event.notifier.type=ARTIX
#####
# Для ARTIX
su.artix.accounting.event.notifier.rest.sender.prefix=http://localhost:8083/rest/sender/
su.artix.accounting.event.notifier.rest.sender.user=admin
su.artix.accounting.event.notifier.rest.sender.password=admin
#####
# TDERA формат сообщения
su.artix.accounting.event.notifier.rest.tdera.url=http://in.gates.wp.tdera.ru/customer/balance/
su.artix.accounting.event.notifier.rest.tdera.auth=Bearer YXJ0aXhfa2Fzc2E6YXJ0aXhfa2Fzc2FfdGVzdA==
su.artix.accounting.event.notifier.rest.tdera.sender=artix
#####
#Версия протокола, согласно которому сервис будет извлекать данные из коллекции с событиями. На текущий момент доступна только VERSION_1
su.artix.accounting.event.notifier.data.protocol.version=VERSION_1
su.artix.accounting.event.notifier.data.mongodb.collection.name=oplog.$main
spring.data.mongodb.uri=mongodb://localhost/local
# После этого количества ошибок сервис уйдет в состояние ERROR и перестанет обрабатывать сообщения.
# Состояние можно получить запросом GET http://localhost:${server.port}/v1/state
su.artix.accounting.event.notifier.retry.count=50
```

Логирование работы сервиса по умолчанию выводится в консоль. Для записи логов в файл в директории с jar-файлом необходимо разместить файл logback.xml.

При настройке и запуске сервиса на внешний сервис будут отправляться сообщения вида:

- в режиме ARTIX (в качестве примера указано формирование запроса, если его посылать с помощью curl):

```
curl -uadmin:admin -XPOST -H "Content-Type: application/json; charset=UTF-8" "http://localhost:8083/rest/sender/send" \
--data '{"number":"3000","accountNumber":"3000","amountAdd":92597,"amountSub": -322,"date":"2018-01-10T08:57:21.234+0000","cashId":"1","shiftNum":4,"checkNum":5,"sessionId":"1$3000-7c39e6b2-750a-400e-9dd1-76cee3039eac$5$4"}'
```

- в режиме TDERA:

```
POST {"params":{"sender":"artix"},"doc":{"number":"3000","accountNumber":"3000","amountAdd":92597,"amountSub": -322,"date":"2018-01-10T08:57:21.234+0000","cashId":"1","shiftNum":4,"checkNum":5,"sessionId":"1$3000-7c39e6b2-750a-400e-9dd1-76cee3039eac$5$4"}}
```

Сервис уведомлений о сгорании бонусов

В accounting-event-notifier добавлен сервис, формирующий сгоревшие бонусы и добавляющий event в БД notifier.

Для того чтобы запустить сервис по отправке сгораний нужно включить настройку командой:

```
su.artix.accounting.event.notifier.bonusfetcher.enable=true
```

и перезапустить сервис:

```
sudo service accounting-event-notifier restart
```

В БД accountingEvent в коллекцию event будут поступать сообщения о сгорании. Для этих видов сообщений (которые будут отдаваться клиенту) добавлено два типа CHECK (это сообщения ADD, SUB, BACK, PAY) и FIRED. Сгорания имеют немного другой вид нежели остальные операции (ADD, SUB, BACK, PAY).

Сгорания будут формироваться из БД, которая будет указана в настройках.

Проводники (asset) группируются по accountNumber (транзакции собираются в одну по номеру счета) с lastStatus=COMMITTED и status=ADD,BACK за время с момента последнего запуска. Настройка su.artix.accounting.event.notifier.bonusfetcher.scheduler.cron отвечает за периодичность запуска сервиса формирования сгораний.

Настройка сервиса

```
#####
# Включение сервиса получения и отправки сгоревших бонусов
su.artix.accounting.event.notifier.bonusfetcher.enable=true
# Настройка запуска сервиса по крону. По-умолчанию раз в сутки
# Сгоревшие бонусы формируются за период с момента последнего запуска
# Если по каким-то причинам эта информация отсутствует(нештатная ситуация),
# то сгоревшие бонусы будут сформированы за последние 24 часа
# Формирование: по счету суммируются все транзакции со статусами ADD и BACK
# На основе этой информации формируется уникальный дескриптор, чтобы не отправлять
# одни и те же сгорания дважды
su.artix.accounting.event.notifier.bonusfetcher.scheduler.cron=0 0 0 * * ?
# Формирование сгоревших бонусов - длительная операция
# Количество попыток сформировать сгоревшие бонусы. По умолчанию 1 попытка
# Пример: настройка = 2
# Сервис формирует сгоревшие бонусы раз в сутки, если произошла нештатная ситуация,
# то сервис попробует снова. Итого 2 попытки
su.artix.accounting.event.notifier.bonusfetcher.trycount=1
# Используемая БД для сервиса формирования сгоревших бонусов. Возможные значения MYSQL, MSSQL
su.artix.accounting.event.notifier.bonusfetcher.db=MSSQL
# Настройки подключения к БС
su.artix.accounting.sql.driver=com.microsoft.sqlserver.jdbc.SQLServerDriver
su.artix.accounting.sql.url=jdbc:sqlserver://localhost:1433;databaseName=artixcsAll
su.artix.accounting.sql.user=sa
su.artix.accounting.sql.passwd=A44H_test
```

Настройка отправки уведомлений об изменении баланса

Один из способов настроить отправки уведомлений покупателям об изменении баланса:

- установить Accounting-сервер
- указать подключение БС к БД Mongo (здесь имеется ввиду не хранилище данных accounting-сервера, а БД, в которую будут помещаться события об изменении баланса на счетах)
- настроить и запустить сервис отправки push-уведомлений accounting-event-notifier (смотри выше)

Настройка формирования сообщений об изменении баланса на счетах в конфигурационном файле /opt/virgo/repository/usr/accounting-logic-properties.properties. Данные об изменениях баланса сохраняются в БД Mongo.

Состав данных, которые будут передаваться на внешние сервисы, определяется выбранными настройками. При включении настройки accounting.changeBalance.event.enabled сервис будет отправлять сообщения, содержащие информацию в формате описанном выше. Для отправки в сообщении не только данных об изменении баланса, но и значение баланса, который установлен на карте после начисления/списания, необходимо для параметра accounting.changeBalance.event.sendBalance установить значение true (активный параметр влияет на производительность).

Наименование	Тип данных	Возможные значения	Описание	Примечания
accounting.changeBalance.event.enabled	логический	<ul style="list-style-type: none"> • true • false 	Генерировать события по изменению баланса	По умолчанию false
accounting.changeBalance.event.protocol.version	строковый	VERSION_1	Внутренняя версия протокола между accounting-сервером и данным сервисом push-уведомлений. Не влияет на формат сообщения, посылаемый внешнему сервису	В настоящее время доступна только VERSION_1
accounting.changeBalance.event.mongodb.uri	строковый		Параметры подключения к БД Mongo, в которую производится запись событий об изменении баланса	

accounting. changeBalance.event. mongodb.database	строковый		Название БД, в которую записываются события об изменении баланса	
accounting. changeBalance.event. sendBalance	логический	<ul style="list-style-type: none"> • true • false 	Указывать в событии баланс по счету после применения операций начисления/списания	По умолчанию false При значении true производительность accounting-сервера может сильно просесть

Пример настройки

Отправка сигналов об изменении баланса на счетах

#включить генерацию событий по изменению баланса. События генерируются на основе команды комита asset, т.е. операции сгорания бонусов в данных событиях НЕ БУДЕТ

accounting.changeBalance.event.enabled = false

#Версия протокола, согласно которому сервис будет извлекать данные из коллекции с событиями. На текущий момент доступна только VERSION_1

accounting.changeBalance.event.protocol.version=VERSION_1

#Параметры подключения к БД mongodb, в которую будут писаться события об изменении баланса

accounting.changeBalance.event.mongodb.uri = mongodb://localhost

#Название БД mongodb, в которую будут писаться события об изменении баланса

accounting.changeBalance.event.mongodb.database = accountingEvent

#Указывать в событии баланс по счету ПОСЛЕ применения операции. Внимание! Включение опции может серьезно просадить производительность!

accounting.changeBalance.event.sendBalance = false