

Установка сервиса push-уведомлений

- [Настройка сервиса push-уведомлений](#)
- [Сервис уведомлений о сгорании бонусов](#)
 - [Настройка отправки уведомлений об изменении баланса](#)
- [Особенности работы](#)
- [Примеры конфигурации](#)

Настройка сервиса push-уведомлений

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

Сервис запускается только под java8.

Следующий пункт для mongo 2.4

Для mongo 3.4 и выше не нужно

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 artixcs-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 artixcs-accounting-event-notifier restart
```

Логи 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 почечового 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 oplog MongoDB, с которой требуется генерировать события	<ul style="list-style-type: none"> • При размещении сервиса на одной машине с бонусным сервером используется <code>mongodb://localhost/local</code> • При размещении сервиса на отдельной машине необходимо указать адрес до oplog 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
```

Логи работы сервиса по умолчанию выводятся 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"}}
```

В сообщении передаются поля:

Поле	Тип данных	Описание	Примечания
number	string	Номер карты	
account Number	string	Номер счета	
amount Add	long	Сумма начислений, на которую изменился баланс карты после последних событий	
amount Sub	long	Сумма списаний, на которую изменится баланс карты после последних событий	Всегда не больше 0
balance	long	Активный баланс карты после последних событий	Указывается при наличии данных Accounting-сервер передает информацию только при включенной настройке

balanceNative	long	Неактивный баланс карты после последних событий	
date	date	Дата и время выполнения операции с бонусами	Формат даты YYYY-MM-DDThh:mm:ss+0000. При почечковом объединении начислений сервис отправляет несколько транзакций, поэтому date берется для одной из входящих в объединение транзакций
cashId	string	Идентификатор кассы	Значения полей получаются из sessionId. Если этих данных нет, то они не передаются в запросе.
shiftNum	integer	Номер смены	
checkNum	integer	Номер чека	
sessionId	string	Идентификатор начислений	При почечковом объединении начислений сервис отправляет несколько транзакций, поэтому sessionId указывается для одной из входящих в объединение транзакций

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 штатная 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 отправку 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
<code>accounting.changeBalance.event.enabled</code>	логический	<ul style="list-style-type: none">• true• false	Генерировать события по изменению баланса	По умолчанию false
<code>accounting.changeBalance.event.protocol.version</code>	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
<code>accounting.changeBalance.event.mongodb.uri</code>	строковый		Параметры подключения к БД Mongo, в которую производится запись событий об изменении баланса	
<code>accounting.changeBalance.event.mongodb.database</code>	строковый		Название БД, в которую записываются события об изменении баланса	
<code>accounting.changeBalance.event.sendBalance</code>	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
```

Особенности работы

Информация о сообщениях, отправленных на внешний сервис, хранится в дополнительных коллекциях `afterSendTs` и `beforeSendTs`. Срок хранения ограничен по времени, которое задается настройкой `su.artix.accounting.event.notifier.mongo.send.event.expired.seconds`, значение по умолчанию составляет 1 час.

Получить состояние сервиса можно, выполнив запрос:

```
curl <host>:8090/v1/state
```

в ответ на запрос возвращается объект:

```
{ "state": "WORKING", "lastReadTimestamp": "2019-01-12T09:09:17.000+0000", "lastSendTimestamp": "2019-01-12T09:00:47.000+0000", "countOfReadingEvents": 4468, "countOfSendingEvents": 1850, "countOfErrorEvents": 0 }
```

Если в ответе возвращается "state": "ERROR", то сервис не работает. Для восстановления работоспособности сервис требуется перезапустить, выполнив запрос:

```
curl -XPUT <host>:8090/v1/restart
```

Сообщения, которые по какой-либо причине не были отправлены сервисом будут отправляться повторно.

В случае, когда принимающий сервис примет сообщение, но в ответ вернется код, не лежащий в диапазоне [200...300], то такое сообщение также будет отправлено повторно.

Если сообщение было получено принимающим сервисом, но сервис отправки не получил ответ (отсутствие связи, таймаут и т.д.), то сообщение будет отправлено повторно.

Примеры конфигурации



Поддержка сервиса Push-уведомлений для accounting-сервера начинается с версии 3.0.250

Для работы сервиса Push-уведомлений с Accounting-сервером версии 3.x необходимы следующие части:

- Accounting-сервер версии выше 3.0.250. Нужно указать следующие настройки в файле конфигурации /opt/virgo/repository/usr/accounting-logic-properties.properties, после чего перезапустить сервис:
 - accounting.changeBalance.event.enabled = true
 - accounting.changeBalance.event.protocol.version = VERSION_1
 - accounting.changeBalance.event.mongodb.uri = mongodb://2.2.2.2
 - accounting.changeBalance.event.mongodb.database = accountingEvent
- СУБД MongoDB версии выше 2.x (предположим, что она запущена на машине с IP: 2.2.2.2, причем accounting-сервер и сервер push-уведомлений имеет до нее доступ; при необходимости для БД можно настроить аутентификацию, в примере она не используется).
- Сервис push-уведомлений. Нужно указать следующие настройки в файле конфигурации application.properties (указаны параметры, отвечающие за подключение к коллекции с изменением баланса):
 - su.artix.accounting.event.notifier.data.protocol.version = VERSION_1
 - su.artix.accounting.event.notifier.data.mongodb.collection.name = event
 - spring.data.mongodb.uri = mongodb://2.2.2.2/accountingEvent