
Интеграция Artix-УНФ

Выпуск 0.41

PashkovKV

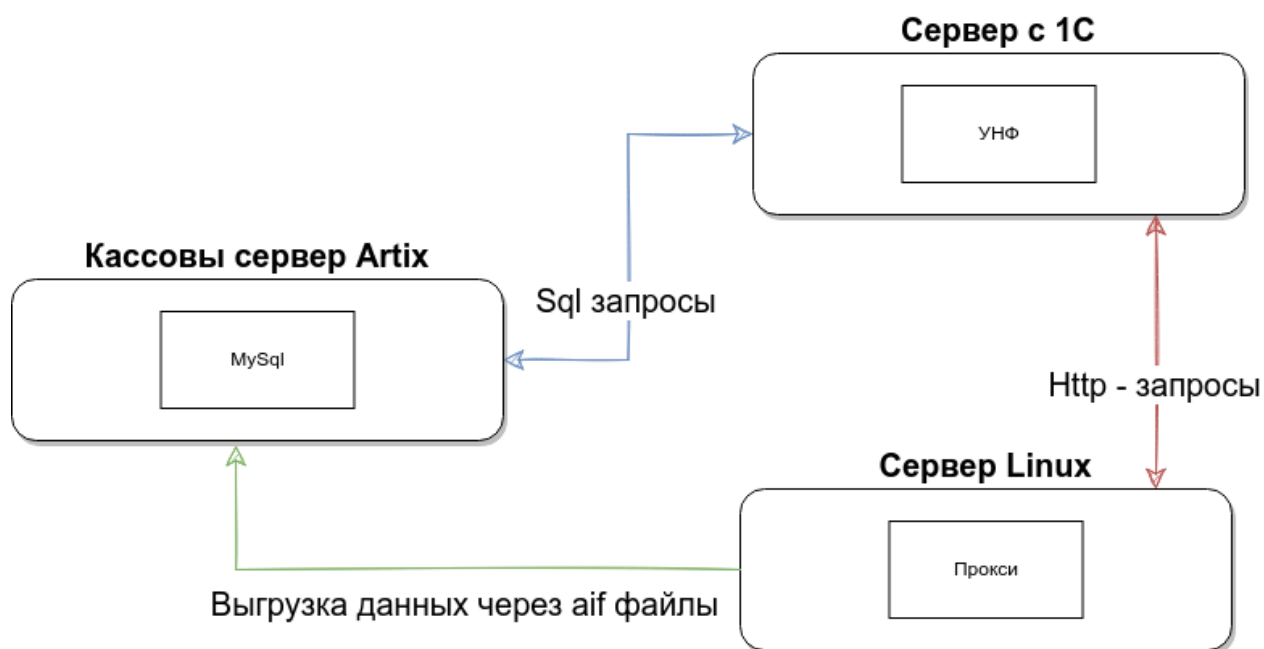
мая 04, 2023

Содержание:

1	Общее описание	1
2	Реализация в УНФ	3
3	Внешняя программа	5
4	Настройки	7
5	Сайт с документацией	17
6	Установка и настройка Gitea	21
7	Файлы	25
8	Индексы и таблицы	27

1.1 Документация по блоку обмена между Artix и УНФ в части выгрузки остатков из УНФ в кассовый сервер Artix и загрузки кассовых смен и продаж в УНФ.

Связь между 1С и Artix осуществляется с помощью Http-сервисов реализованных на стороне УНФ. Данные по продажам 1С забирает из базы данных Кассового сервера с помощью Sql запросов. Таблицы из БД Кассового сервера подключены в УНФ в качестве внешнего источника данных. Прокси сервер реализован на Python и располагается на отдельном сервере с Linux.



Работа происходит следующим образом: В УНФ в справочнике «крю _Настройки» в предопределенном реквизите «КаталогКонтроляОбменаArtix» хранится путь до каталога интеграции с Artix, который имеет следующую структуру -

```
| /
|_ unf_teleport..... корневой каталог
|   |_ 21..... каталог магазина (соответствует номеру магазина)
|   |_ 22..... каталог магазина (соответствует номеру магазина)
|   |_ 23..... каталог магазина (соответствует номеру магазина)
|   |_ 25..... каталог магазина (соответствует номеру магазина)
|   |_ 78..... каталог магазина (соответствует номеру магазина)
```

При проведении документа товародвижения срабатывает подписка на событие и в каталоге соответствующем номеру магазина создается файл-флаг вида: «1681466893_chg», где число это псевдослучайное число образованное разностью дат, для достижения уникальности имени файла.

Программа Brocker запускаясь с указанной настройкой сервера периодичностью начинает обрабатывать кассовые смены. Вначале в УНФ передаются данные по вновь открытым сменам в Artix. УНФ получая информацию создает новые документы «Кассовая смена» со статусом «Открытая» и автоматически создается документ «ОРП» у которого заполнена только шапка и имеется ссылка на кассовую смену. После этого Brocker отправляет информацию по закрытым кассовым сменам. УНФ заполняет оставшиеся реквизиты в документе «Кассовая смена» и присваивает документу статус «Закрыта (чеки за архивированы)». Далее согласно созданному в УНФ расписанию запускается обработка по формированию «ОРП». Данные для заполнения табличных частей «ОРП» берутся запросом из кассового сервера, таблицы которого подключены в 1С в качестве внешнего источника данных.

Далее программа прокси отправляет запрос («GetChangeShop») в УНФ для получения номеров магазинов по которым были изменения остатков в ту или иную сторону. УНФ «просматривает» каталоги в папке интеграции и возвращает список номеров магазинов в каталогах которых, были обнаружены файлы флаги. Затем программа Brocker перебирая номера магазинов формирует в УНФ запрос остатков по каждому. Получив остатки, Brocker обрабатывает данные - для номенклатуры с аналогами переносит суммируя все остатки на головную номенклатуру, а аналоги удаляет из выгрузки, таким образом на кассы падает в остатках, только головная номенклатура. После формирования остатков, на основе этих данных формируется файл формата aif и выгружается в соответствующий каталог магазина. Кассовый сервер через определенное время «забирает» файл и формирует номенклатуру с остатками на магазине.

План: Планируется добавить модуль контроля товаро-остатков пришедших из УНФ и выгруженных в кассовый сервер

Функционал работы с Artix со стороны УНФ реализован с помощью:

1. Подписки на события для документов товародвижения:

- 1) Поступление товаров
- 2) Списание товаров
- 3) Оприходование товаров
- 4) Перемещение товаров ?
- 5) Возврат товаров поставщику
- 6) Отчет о розничных продажах ?

2. Http-сервисы:

1. Http-сервис Workshift
2. GetChangeShop
3. GetProductRemains

3. Фоновых заданий и обработок

Обработка поформированию ОРП подключается в дополнительные обработки и для нее настраивается расписание запуска. Обработка ищет непроведенные документы ОРП, которые связаны с документом «Кассовая смена», со статусом «Закрыта». И выполняет заполнение этих документов продажами из таблиц базы данных «Кассового сервера» Artix, подключенными в программу, как внешние источники данных.

Настройки публикации базы

Публикация на веб-сервере [X]

Основные OpenID Прочие

Имя:

Веб-сервер:

Каталог:

☒ Публиковать доступ для клиентских приложений

☒ Публиковать стандартный интерфейс OData

☒ Публиковать систему аналитики

Web-сервисы HTTP сервисы

☒ Публиковать HTTP сервисы по умолчанию

	Имя	Корневой URL
<input checked="" type="checkbox"/>	ПередачаДанных	dt
<input checked="" type="checkbox"/>	СервисЛояльности	loyaltyservice
<input checked="" type="checkbox"/>	СинхронизацияМПУНФ	synchronization
<input checked="" type="checkbox"/>	УдалитьvatsYandexTelephonyApi	yandex-api
<input checked="" type="checkbox"/>	ЭДО	edi
<input checked="" type="checkbox"/>	test_s	test_s
<input checked="" type="checkbox"/>	ctrl	ctrl
<input checked="" type="checkbox"/>	GetChangeShop	GetChangeShop
<input checked="" type="checkbox"/>	Workshift	Workshift

☐ Публиковать HTTP сервисы расширений по умолчанию

☐ Публиковать дистрибутив тонкого клиента

Расположение публикуемого дистрибутива:

Адрес перехода при окончании работы веб-клиента:

Наименование прогрессивного веб-приложения:

Опубликовать Отключить Сохранить Загрузить Закрывать Справка

Внешняя программа

Внешняя программа реализована на языке Python и расположена на отдельном сервере с установленной серверной операционной системой Linux Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-69-generic x86_64)

Программа состоит из двух блоков, первый отвечает за получение остатков товаров из обработки и передачи на кассовый сервер Artix, второй за отслеживание открытия и закрытия смен на кассах и периодической передачи этих данных в УНФ.

Общее описание

Работа программы начинается с чтения файла с настройками `strset`

1. Установка и настройка
2. Кассовые смены
3. Работа с остатками

4.1 Порядок внедрения касс в работу с Артикс:

4.1.1 1. Создать магазин + касса на сервере.

1. Создание Магазина - Вводить Код, Название, Формат справочников (из 1С), метка дисконта (выбор из созданных)
2. Создание Кассы - Вводить Код, Название, Хост (IP), Тип кассы (Рабочее место кассира), Формат выгрузки продаж (с КС в 1С), Категория справочников (если используется), Состояние обмена (рекоменд. Выключен). При использовании Конфигураций - Выбрать конфигурацию, и окружение (для рабочей кассы - Боевое)
 - Создать структуру Магазин-Кассы можно/нужно заранее.

4.1.2 2. Настройка ПОС

Полуручной:

1. Самый простой способ предварительно настроить 1 пос, и сохранить нужные файлы настроек (настройки, оборудование и т.д.). Далее скопировать на другую ПОС в требуемые директории.

Автоматически:

1. Использовать менеджер-конфигураций (что есть в КС), одна тонкость - возможно не все настройки можно произвести через него.
2. Включить обмен на сервере и запустить полную выгрузку справочников с сервера. После полной прогрузки справочников - перезапустить кассовую программу на ПОС

4.2 Добавление группы сертификатов со сроком действия через REST API.

1. Добавление группы сертификатов со сроком действия через REST API.

- Добавить группу сертификатов в БД КС, сформировав запрос типа POST на url: `http://<host>:<port>/Csrest/rest/dictionaries/certificates`
- Тело запроса в формате Json должно быть подобного формата:

```
{
    "code": "1",
    "rangefrom": 30,
    "rangeto": 40,
    "name": "Сертификаты",
    "sum": 300,
    "inputmask": 31,
    "discountcampaign": "test campaign"
}
```

- Атрибут свойства discountcampaign должен содержать yaml акции, в которой как раз можно указать сроки действия. Например, если смотреть на уже существующую группу сертификатов, созданную вручную через web-интерфейс LM с указанием срока действия (БД КС ArtixAll, таблица certificateTemp), то в поле discountcampaign увидим запись вида:

```
!!python/object:artixds.domain.DiscountCampaign
active: true
beginBonusesDateCalculate: null
beginBonusesIntervalCondition: !!python/unicode 'date'
beginBonusesIntervalMetric: !!python/unicode 'day'
beginBonusesIntervalValue: 0
beginDate: null
beginDateWithBeginCurrentMetric: !!python/unicode 'day'
beginDateWithBeginNextMetric: !!python/unicode 'day'
beginRangeDate: 2022-02-17
beginTime: null
checkedTags: null
daysOfWeek: []
discounts: []
endBonusesDateCalculate: null
endBonusesIntervalCondition: !!python/unicode 'date'
endBonusesIntervalMetric: !!python/unicode 'day'
endBonusesIntervalValue: 0
endDate: null
endRangeDate: 2022-02-17
endTime: null
id: 57026161735168
interactionType: all
labels: null
manualActivate: null
name: !!python/unicode ''
parent: null
parentGroup: null
```

(continues on next page)

(продолжение с предыдущей страницы)

```
priority: null
qualifiers: !!set {}
tagsIsAllowed: false
weight: 300
```

То есть в этом случае срок действия для группы сертификатов определен свойствами beginRangeDate и endRangeDate (от 2022-02-17 до 2022-02-17).

Берем этот Yaml как шаблон и изменяем даты beginRangeDate и endRangeDate на нужные.

- d. Далее этот Yaml необходимо переделать в однострочный формат. Для этого после каждой строки добавляем \n, и переносим все строки в одну. В итоге должна получиться строка следующего вида:

```
!python/object:artixds.domain.DiscountCampaign\r\nactive: true\r\n
↪nbeginBonusesDateCalculate: null\r\nnbeginBonusesIntervalCondition: !!
↪python/unicode 'date'\r\nnbeginBonusesIntervalMetric: !!python/unicode 'day
↪'\r\nnbeginBonusesIntervalValue: 0\r\nnbeginDate: null\r\n
↪nbeginDateWithBeginCurrentMetric: !!python/unicode 'day'\r\n
↪nbeginDateWithBeginNextMetric: !!python/unicode 'day'\r\nnbeginRangeDate:
↪2022-02-17\r\nnbeginTime: null\r\nncheckedTags: null\r\nndaysOfWeek: []\r\n
↪ndiscounts: []\r\nnendBonusesDateCalculate: null\r\n
↪nendBonusesIntervalCondition: !!python/unicode 'date'\r\n
↪nendBonusesIntervalMetric: !!python/unicode 'day'\r\n
↪nendBonusesIntervalValue: 0\r\nnendDate: null\r\nnendRangeDate: 2022-02-17\r\n
↪r\nnendTime: null\r\nnid: 62331387691573\r\nninteractionType: all\r\nnlabels:
↪null\r\nnmanualActivate: null\r\nnname: !!python/unicode ''\r\nnparent: null\r\n
↪r\nnparentGroup: null\r\nnpriority: null\r\nnqualifiers: !!set {}\r\n
↪ntagsIsAllowed: false\r\nnweight: 300
```

И в таком виде эту строку можно добавить в тело POST-запроса (пункт 1.1) вместо записи test campaign.

2. Отправляем POST-запрос, указав данные для авторизации (как от WEB КИ)

Welcome	Request1
---------	----------

POST
http://192.169.1.159:8080/Csrest/rest/dictionaries/certificates

JSON	HEADERS	QUERY STRING	BASIC	ASSERTION
------	---------	--------------	-------	-----------

USERNAME

PASSWORD

☒ Show Password

Далее убеждаемся что запрос прошел без ошибок и проверяем в web-интерфейсе LM, что создалась группа сертификатов. Открываем эту группу на редактирование и убеждаемся, что сроки действия прописаны корректно.

[Не сохранять группу сертификатов](#)

Редактирование группы сертификатов

Код группы	<input type="text" value="3"/>	Номинал	
Название группы *	<input type="text" value="Сертификаты"/>	Минимальная сумма	
Начало диапазона номеров	<input type="text" value="30"/>	Максимальная сумма	
Окончание диапазона номеров	<input type="text" value="40"/>	Кратность	<input type="text"/>
Код налога	<input type="text"/>	Код товара	<input type="text"/>
Признак предмета расчета	<input type="text"/>	Код отдела	<input type="text"/>
Начало действия сертификатов - 17 февраля 2022 Окончание действия сертификатов - 17 февраля 2022		<input type="checkbox"/> Полное списание <input type="checkbox"/> Игнорировать после	

[Изменить](#)

3. Затем добавляем сертификат на сервер POST-запросом `http://<хост>:<порт бонусного сервера/сервера сертификатов>/ACC/rest/v1/cards`.

Тело запроса должно быть вида:

```
{
  "number": "30",
  "accountNumber": "30",
  "status": "EARN_PAY"
}
```

Важно, чтобы значения `number` и `accountNumber` входили в диапазон номеров созданной группы сертификатов.

4.3 Дополнительные материалы:

4.3.1 1. Список директорий и файлов основных настроек.

1. `/linuxcash/cash/conf/`
 - 1) `ncash.ini` - основной файл настроек
 - 2) `bcode.ini` - правила разбора ШК
 - 3) `finit.conf` - инициализация ФР

- 4) cliche.txt и tail.txt - Клише/подвал чека
- 5) artixsb.conf

2. /linuxcash/cash/conf/ncash.ini.d/

- 1) artixbonus.ini - работа с бонусным сервером
- 2) artixcertificate.ini
- 3) deferredcheck.ini - отложенные чеки
- 4) fr.ini
- 5) gui.ini - выбор интерфейса ПО
- 6) motp.ini - сигареты
- 7) markedgoods.ini - другая маркировка

3. /linuxcash/cash/conf/drivers/ - Настройки подключения оборудования

Сначала настраивается одна ПОС, полученные настройки используем как эталонные. После установки ПО на новую ПОС копируем настройки, включая “Подключаемое оборудование”, проверяем и при необходимости вносим правки.

4.3.2 2. Особенности настройки “Сбербанк”

Для работы с процессингом сбербанк необходимо установить пакет - *artix45-plugin-sb*. В директорию `”/linuxcash/cash/paysystems/sb/”` разместить файлы приложения сбербанк полученного от сотрудников банка. Настроить подключение ПинПад согласно инструкции банка (pinpad.ini). Следует обратить внимание что при подключении пинпад через USB, необходимо правильно создать символьную ссылку на устройство:

```
ln -s /dev/serial/by-id/{Имя_Устройства} /linuxcash/cash/paysystems/sb/ttyS99
```

Для работы с терминалом PAX SP30 - дополнительно установить пакет *ttypos*

4.3.3 3. Ссылка на скачивание дистрибутива yuki

<http://update.artix.su/static/yuki/>

4.3.4 4. Остатки

в конфигурационном файле `/linuxcash/cash/conf/ncash.ini` Изменить:

```
;calculateRemainsOfGoods = false
.....
;considerRemainsOfGoods = false
```

на:

```
calculateRemainsOfGoods = true
.....
considerRemainsOfGoods = true
```

После изменения настроек надо перезапустить кассовое ПО. Тогда касса будет считать остатки и учитывать их при продаже

Для товара, у которого не нужно контролировать остатки (кулинария) при выгрузке не указываем дату остатков - *remaindate*, а поле *remain* оставляем **пустым!!!** (не «0»!!!).

4.3.5 5. Дисконтные карты (создание)

1. создать счет на бонусном сервере
 2. привязать номер карты к счету
 3. создать карту на кассовом сервере (группа должна быть создана заранее)
- 1 и 2 это работа с API бонусного сервера
- 3 это работа с файлом аиф, или через REST API загрузки справочников с Кассовым сервером.
- <https://docs.artix.su/pages/viewpage.action?pageId=27530936>

4.3.6 6. Настройка действий и контекста клавиш

4.3.7 7. Конфигурирование дисконтной системы

Ссылка на сайт с полной инструкцией

Дисконтная система поставляется отдельным пакетом `artix45-discountsystem`.

Наличие минимальной цены влияет на методы распределения скидок на чек и наборы. Минимальные цены не учитываются на этапе расчета скидок, дополнительное ограничение минимальной цены накладывается на этапе применения.

Сумма скидки на чек распределяется в два этапа:

- распределение по позициям пропорционально стоимости каждой позиции,
- размещение остатка максимально возможными суммами начиная с первой позиции.

Правила распределения остатка определяется параметром **distributeReceiptDiscountFully** в конфигурационном файле `/linuxcash/cash/conf/ncash.ini` в секции `[Discount]`:

- при значении параметра `true` остаток скидки распределяется максимально возможными суммами, учитывая ограничения минимальной ценой товара. Выполняется только первый этап распределения скидки;
- при значении параметра `false` сумма скидки на чек распределяется по позициям пропорционально их стоимости.

Размер скидки на позицию ограничивается минимальной ценой товара. В случае предоставления скидок на наборы минимальную цену на товары можно игнорировать для того, чтобы иметь возможность задавать акции вида «купи две штуки и одну получи в подарок», когда скидка назначается только на один из товаров. Возможность игнорирования минимальной цены при предоставлении скидки на наборы задается параметром **ignoreMinimalPriceForKit** в конфигурационном файле `/linuxcash/cash/conf/ncash.ini` в секции `[Discount]`:

- при значении параметра `true` минимальная цена при применении скидки на набор игнорируется,
- при значении параметра `false` скидка на комплект будет предоставлена с учетом минимальной цены на товар.

Клавиши			
Код	Значение	Действие	Контекст
1068	ALT+D	Ввести 00	Все режимы
330	DEL	Сторно текущей позиции	Формирование товарной части документа
324	ALT+F12	Подитог	Формирование товарной части документа
320	ALT+F8	Выбрать по наименованию	Формирование товарной части документа
319	ALT+F7	Выбрать по штрих-коду	Формирование товарной части документа
318	ALT+F6	Карта: добавить	Формирование товарной части документа
316	ALT+F4	Сторно текущей позиции	Формирование товарной части документа
315	ALT+F3	Возврат по чеку продажи	Формирование товарной части документа
314	ALT+F2	Сторно всех позиций	Формирование товарной части документа
295	CTRL+F7	Выход	Формирование товарной части документа
288	SHIFT+F12	Применить модификаторы к позиции (Кол-во и.т.д.)	Формирование товарной части документа
278		Выбрать тип оплаты	Оплата документа
276	F12	Изменить количество	Формирование товарной части документа
272	F8	Оплатить бонусами	Оплата документа
271	F7	Подитог по карте	Формирование товарной части документа
270	F6	Выбрать тип оплаты	Оплата документа
84	SHIFT+T		Формирование товарной части документа
83	SHIFT+S		Формирование товарной части документа
82	SHIFT+R		Формирование товарной части документа
81	SHIFT+Q		Формирование товарной части документа
80	SHIFT+P		Формирование товарной части документа
79	SHIFT+O		Формирование товарной части документа
78	SHIFT+N		Формирование товарной части документа
77	SHIFT+M	Пицца3	Формирование товарной части документа
76	SHIFT+L		Формирование товарной части документа
75	SHIFT+K		Формирование товарной части документа
74	SHIFT+J		Формирование товарной части документа
73	SHIFT+I		Формирование товарной части документа
72	SHIFT+H		Формирование товарной части документа
71	SHIFT+G		Формирование товарной части документа
70	SHIFT+F	Келлерс	Формирование товарной части документа
69	SHIFT+E	Пицца1	Формирование товарной части документа
68	SHIFT+D		Формирование товарной части документа
67	SHIFT+C		Формирование товарной части документа
66	SHIFT+B		Формирование товарной части документа
65	SHIFT+A		Формирование товарной части документа
47	/	Итог	Все режимы

В основе дисконтной системы заложена идея использования любых доступных элементов чека (товарная позиция, дисконтная карта и т.д.) для проверки условий и вычисления ставок. Так как условия скидок могут быть сложными, то после изменения состава чека выполняется отмена ранее примененных скидок, последующий пересчет и применение новых. Пересчет скидок может занимать продолжительное время, это может происходить из-за недостатка ресурсов машины, на которой осуществляется работа, или из-за сложности самих скидок. Кассовое ПО Artix может быть настроено таким образом, чтобы расчет скидок происходил как в процессе формирования чека, так и только при переходе к оплате документа. Момент расчета скидок задается параметром **recalcDiscInFormingMode** в конфигурационном файле `/linuxcash/cash/conf/ncash.ini` в секции `[Check]`:

- при значении параметра `true` скидки пересчитываются в режиме формирования документа,
- при значении параметра `false` скидки пересчитываются при переходе в подитог.

Если расчет скидок происходит в режиме формирования документа, то пересчет скидок выполняется после:

- добавления или сторнирования товарной позиции,
- добавления или удаления карты,
- изменения модификаторов (цена, количество и т.п.) товарной позиции.

При использовании акций со ставкой 0% (например, рекламных) можно контролировать запись информации о сработавшем дисконтном воздействии на позицию в БД `Documents` таблицу `Discitem`. Возможность сохранения скидки 0% на позицию в БД задается параметром **saveNullDiscountForPosition** в конфигурационном файле `/linuxcash/cash/conf/ncash.ini` в секции `[Discount]`:

- при значении параметра `true` скидки 0% на позицию сохраняются в БД,
- при значении параметра `false` скидки 0% на позицию не сохраняются в БД.

Наименование	Тип данных	Возможные значения	Описание	Примечания
<code>distributeReceipt</code>	логический	<code>true</code> , <code>false</code>	Полностью распределять по позициям сумму скидки на чек	По умолчанию <code>true</code>
<code>ignoreMinimalPri</code>	логический	<code>true</code> , <code>false</code>	Игнорировать минимальную цену при применении скидки на комплект	По умолчанию <code>true</code>
<code>recalcDiscInForm</code>	логический	<code>true</code> , <code>false</code>	Пересчитывать скидки в режиме формирования чека	По умолчанию <code>true</code>
<code>saveNullDiscount</code>	логический	<code>true</code> , <code>false</code>	Сохранять скидку 0% на позицию в базу данных	По умолчанию <code>false</code>

S

Пример настроек

```
[Discounts]
;
; Скидки
;
...
; Полностью распределять по позициям сумму скидки на чек
distributeReceiptDiscountFully = false
...
; Сохранять скидку 0% на позицию в базу данных
```

(continues on next page)

(продолжение с предыдущей страницы)

```
saveNullDiscountForPosition = true
; Игнорировать минимальную цену при применении скидки на комплект
ignoreMinimalPriceForKit = false

[Check]
;
; Настройки чека
;
...
; Пересчитывать скидки в режиме формирования чека, по-умолчанию true.
recalcDiscInFormingMode = true
```

Сайт с документацией

5.1 Разворачивание сайта:

Git должен уже быть установлен в системе - gitinst

5.1.1 1. Копирование Git репозитория.

```
$ mkdir ~/Doc  
$ cd Doc/  
$ git clone https://github.com/krugersu/Artix_Global
```

5.1.2 2. Установка Apache.

Ссылка на сайт с полной [инструкцией](#)

Установка без настройки брандмауэра.

Установка:

```
$ sudo apt update  
$ sudo apt install apache2  
$ sudo systemctl status apache2
```

Получаем вывод вида:

```
apache2.service - The Apache HTTP Server  
Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)  
Active: active (running) since Thu 2020-04-23 22:36:30 UTC; 20h ago
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Docs: https://httpd.apache.org/docs/2.4/
Main PID: 29435 (apache2)
Tasks: 55 (limit: 1137)
Memory: 8.0M
CGroup: /system.slice/apache2.service
├─29435 /usr/sbin/apache2 -k start
├─29437 /usr/sbin/apache2 -k start
└─29438 /usr/sbin/apache2 -k start
```

Согласно этому выводу сервис успешно запустился. Но лучше ещё запросить страницу Apache, чтобы убедиться, что веб-сервер работает правильно.

5.1.3 3. Управление процессами Apache.

Чтобы остановить веб-сервер, введите:

```
$ sudo systemctl stop apache2
```

Запустить:

```
$ sudo systemctl start apache2
```

Перезапуск:

```
$ sudo systemctl restart apache2
```

По умолчанию Apache добавлен в автозагрузку.

5.1.4 4. Настройка виртуальных хостов.

На веб-сервере Apache вы можете использовать виртуальные хосты (в Nginx это блоки `server`) для изоляции настроек и размещения нескольких доменов на одном сервере. Здесь используется условный домен `example.com`, который нужно заменить доменом вашего сайта.

В Ubuntu 22.04 веб-сервер Apache по умолчанию предоставляет один включенный виртуальный хост, который обслуживает каталог `/var/www/html`. Этого хватит для работы одного сайта, но если вы хотите разместить несколько сайтов, вам нужно создать новые виртуальные хосты. Создайте структуру каталогов в `/var/www` для сайта `example.com`, а `/var/www/html` оставьте как каталог по умолчанию, который будет обслуживаться, если запрос клиента не соответствует другим сайтам.

Создайте каталог для `your_domain`:

```
$ sudo mkdir /var/www/krudoc
```

Затем укажите права на каталог с помощью переменной `$USER`:

```
$ sudo chown -R $USER:$USER /var/www/krudoc
```

Права должны быть предоставлены, если вы не меняли значение `umask`, которое устанавливает права файлов по умолчанию. Чтобы убедиться, что права предоставлены и пользователь может работать с файлами, нужно ввести команду:

```
$ sudo chmod -R 755 /var/www/krudoc
```

Далее копируем содержимое каталога `~/Doc/Artix_Global/build/html` в `/var/www/krudoc`

```
$ cp -r ~/Doc/Artix_Global/build/html/. /var/www/krudoc
```

Нужно создать файл виртуального хоста с правильными директивами, чтобы Apache мог обслуживать этот контент. Чтобы напрямую не изменять файл конфигурации по умолчанию, расположенный в `/etc/apache2/sites-available/000-default.conf`, мы создадим новый в `/etc/apache2/sites-available/your_domain.conf`:

```
$ sudo nano /etc/apache2/sites-available/krudoc.conf
```

Вставьте в файл следующий блок настроек. Он похож на конфигурации по умолчанию, но содержит правильный домен и каталог:

```
<VirtualHost *:80>
    ServerAdmin pk69@kruger.su
    ServerName krudoc
    ServerAlias www.krudoc
    DocumentRoot /var/www/krudoc
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Обратите внимание, мы обновили DocumentRoot на наш новый каталог, а ServerAdmin на адрес электронной почты, к которому может получить доступ администратор сайта `your_domain`. Также мы добавили две директивы: ServerName, которая устанавливает базовый домен, который должен соответствовать этому определению виртуального хоста, и ServerAlias – определяет дополнительные имена, которые должны соответствовать базовому имени.

После того, как закончите, сохраните и закройте файл.

Включим файл с помощью `a2ensite`:

```
$ sudo a2ensite krudoc.conf
```

Отключите сайт по умолчанию, определенный в `000-default.conf`:

```
$ sudo a2dissite 000-default.conf
$ systemctl reload apache2
```

Далее проверим ошибки настроек:

```
$ sudo apache2ctl configtest
```

Вывод должен быть следующим:

```
$ Syntax OK
```

Перезапустите Apache, чтобы изменения вступили в силу:

```
$ sudo systemctl restart apache2
```

Теперь Apache должен обслуживать ваше доменное имя. Для проверки перейдите по адресу <http://krudoc>, вы должны увидеть что-то вроде этого:

5.1.5 5. Важные файлы и каталоги Apache.

Контент

#. */var/www/html*: Фактический веб-контент, который по умолчанию состоит только из стандартной страницы Apache (ранее вы её уже видели), обслуживается из каталога */var/www/html*. Это можно изменить, внося корректировки в файлы конфигурации Apache.

Настройка сервера

1. */etc/apache2*: каталог конфигурации Apache. Все файлы настроек Apache находятся здесь.
2. */etc/apache2/apache2.conf*: **основной файл настройки Apache. Его можно корректировать**
для внесения изменений в глобальную конфигурацию Apache. Этот файл отвечает за загрузку многих других файлов в каталоге конфигурации.
3. */etc/apache2/ports.conf*: **В этом файле указаны порты, которые прослушивает Apache.**
По умолчанию Apache прослушивает порт 80 и дополнительно порт 443, если включен SSL.
4. */etc/apache2/sites-available/*: **каталог, в котором могут храниться виртуальные хосты**
для каждого сайта. Apache не будет использовать файлы настроек из этого каталога, если они не связаны с каталогом *sites-enabled*. Обычно вся конфигурация блока выполняется в этом каталоге, а затем включается путем ссылки на другой каталог с помощью команды *a2ensite*.
5. */etc/apache2/sites-enabled/*: **каталог, в котором хранятся включенные виртуальные хосты**
для каждого сайта. Обычно они создаются по ссылке на файлы настроек из каталога *sites-available* с помощью команды *a2ensite*. Apache считывает конфигурационные файлы и ссылки, находящиеся в этом каталоге, при запуске или перезагрузке для компиляции полной настройки.
6. */etc/apache2/conf-available/*, */etc/apache2/conf-enabled/*: **Эти каталоги имеют те же отношения,**
что и каталоги *sites-available* и *sites-enabled* но используются для хранения фрагментов настроек, которые не принадлежат виртуальному хосту. Файлы в каталоге *conf-available* можно включить с помощью команды *a2enconf*, а отключить – *a2disconf*.
7. */etc/apache2/mods-available/*, */etc/apache2/mods-enabled/*: **эти каталоги содержат доступные**
и включенные модули соответственно. Файлы, заканчивающиеся на *.load*, содержат фрагменты для загрузки определенных модулей, а которые заканчиваются на *.conf*, содержат настройки этих модулей. Модули можно включать и отключать с помощью команд *a2enmod* и *a2dismod*.

Логи сервера

1. */var/log/apache2/access.log*: **каждый запрос к вашему веб-серверу записывается в этот лог по умолчанию,**
если Apache не настроен иначе.
2. */var/log/apache2/error.log*: **по умолчанию все ошибки записываются в этот файл.**
Директива *LogLevel* в конфигурации Apache определяет, насколько подробную информацию будут содержать журналы ошибок.

Установка и настройка Gitea

Git должен уже быть установлен в системе - gitinst

Ссылка на сайт с полной [инструкцией](#)

6.1 1. Подготовка

Gitea поддерживает SQLite, PostgreSQL и MySQL / MariaDB в качестве серверной части базы данных.

Мы будем использовать SQLite в качестве базы данных для Gitea. Если SQLite не установлен в вашей системе Ubuntu, вы можете установить его, введя следующие команды от имени пользователя sudo :

```
$ sudo apt update  
$ sudo apt install sqlite3
```

6.2 2. Установка Gitea

Проверьте установку, отобразив версию Git:

```
$ git --version
```

```
$ git version 2.34.1
```

6.3 3. Создайте пользователя Git

Создайте нового системного пользователя, который будет запускать приложение Gitea, набрав:

```
$ sudo adduser --system --group --disabled-password --shell /bin/bash --home /home/git --  
→gecos 'Git Version Control' git
```

Команда создаст нового пользователя и группу с именем git и установит домашний каталог в /home/git. Результат будет выглядеть примерно так:

```
Adding system user `git' (UID 111) ...  
Adding new group `git' (GID 116) ...  
Adding new user `git' (UID 111) with group `git' ...  
Creating home directory `/home/git' ...
```

6.4 4. Скачать бинарный файл Gitea

Посетите страницу загрузки Gitea и загрузите последнюю версию двоичного файла для вашей архитектуры. На момент написания последняя версия — 1.19.3. Если доступна новая версия, измените переменную VERSION в приведенной ниже команде.

Загрузите двоичный файл Gitea в каталог /tmp используя следующую команду wget :

```
$ wget sudo wget -O /tmp/gitea https://dl.gitea.com/gitea/1.19.3/gitea-1.19.3-linux-amd64
```

gitea файл gitea можно запускать из любого места. Мы будем следовать соглашению и переместим двоичный файл в каталог /usr/local/bin :

```
$ sudo mv /tmp/gitea /usr/local/bin
```

Сделайте двоичный исполняемый файл:

```
$ sudo chmod +x /usr/local/bin/gitea
```

Выполните следующие команды, чтобы создать каталоги и установить необходимые разрешения и владение :

```
$ sudo mkdir -p /var/lib/gitea/{custom,data,indexers,public,log}  
$ sudo chown git: /var/lib/gitea/{data,indexers,log}  
$ sudo chmod 750 /var/lib/gitea/{data,indexers,log}  
$ sudo mkdir /etc/gitea  
$ sudo chown root:git /etc/gitea  
$ sudo chmod 770 /etc/gitea
```

Указанная выше структура каталогов рекомендована официальной документацией Gitea. Разрешения для каталога /etc/gitea установлены на 770 чтобы мастер установки мог создать файл конфигурации.

6.5 5. Создайте файл модуля Systemd

Gitea предоставляет файл модуля Systemd, который уже настроен в соответствии с нашей настройкой. Загрузите файл в каталог `/etc/systemd/system/`, набрав:

```
$ sudo wget https://raw.githubusercontent.com/go-gitea/gitea/master/contrib/systemd/
↳ gitea.service -P /etc/systemd/system/
```

После этого включите и запустите службу Gitea:

```
$ sudo systemctl daemon-reload
$ sudo systemctl enable --now gitea
```

Убедитесь, что служба запущена успешно:

```
$ sudo systemctl status gitea
```

```
gitea.service - Gitea (Git with a cup of tea)
Loaded: loaded (/etc/systemd/system/gitea.service; enabled; vendor preset: enabled)
Active: active (running) since Sat 2020-01-04 21:27:23 UTC; 3s ago
Main PID: 14804 (gitea)
Tasks: 9 (limit: 1152)
CGroup: /system.slice/gitea.service
        └─14804 /usr/local/bin/gitea web --config /etc/gitea/app.ini
...
```

6.6 6. Настроить Gitea

Теперь, когда Gitea загружена и запущена, пора завершить установку через веб-интерфейс. По умолчанию Gitea прослушивает подключения через порт 3000 на всех сетевых интерфейсах. Если на вашем сервере работает брандмауэр UFW, вам нужно открыть порт Gitea:

Откройте браузер, введите `http://YOUR_DOMAIN_IR_IP:3000`

Настройки базы данных:

1. Тип базы данных: SQLite3
2. Путь: используйте абсолютный путь, `/var/lib/gitea/data/gitea.db`

Общие настройки приложения:

1. Заголовок сайта: введите название вашей организации.
2. Корневой путь репозитория: оставьте значение по умолчанию `/home/git/gitea-repositories`.
3. Корневой путь Git LFS: оставьте значение по умолчанию `/var/lib/gitea/data/lfs`.
4. Запуск от имени пользователя: git
5. Домен SSH-сервера: введите IP-адрес вашего домена или сервера.
6. Порт SSH: 22, измените его, если SSH прослушивает другой порт
7. Порт прослушивания HTTP Gitea: 3000
8. Базовый URL Gitea: используйте http и IP-адрес вашего домена или сервера.

9. Путь к `/var/lib/gitea/log` : оставьте значение по умолчанию `/var/lib/gitea/log`

Вы можете изменить настройки в любое время, отредактировав файл конфигурации Gitea. После этого нажмите кнопку «Установить Gitea». Установка происходит мгновенно. По завершении вы будете перенаправлены на страницу входа. Щелкните ссылку «Зарегистрироваться сейчас». Первый зарегистрированный пользователь автоматически добавляется в группу администратора.

Измените разрешения файла конфигурации Gitea на доступ только для чтения, используя:

```
$ sudo chmod 750 /etc/gitea
$ sudo chmod 640 /etc/gitea/app.ini
```

Вот и все. Gitea установлена на вашем компьютере с Ubuntu.

6.7 7. Обновление Gitea

Чтобы обновить Gitea до последней версии, просто скачайте и замените двоичный файл.

1. Остановите сервис Gitea:

```
$ sudo systemctl stop gitea
```

2. Загрузите последнюю версию Gitea и переместите ее в каталог `/usr/local/bin` :

```
$ VERSION=<THE_LATEST_GITEA_VERSION>
$ wget -O /tmp/gitea https://dl.gitea.io/gitea/${VERSION}/gitea-${VERSION}-linux-amd64
$ sudo mv /tmp/gitea /usr/local/bin
```

3. Сделайте двоичный исполняемый файл:

```
$ sudo chmod +x /usr/local/bin/gitea
```

4. Перезапустите сервис Gitea:

```
$ sudo systemctl restart gitea
```

Это тест

Скачать эту документацию в pdf.

Индексы и таблицы

- `genindex`
- `modindex`
- `search`

Что реализовать:

План: Планируется добавить модуль контроля товаро-остатков пришедших из УНФ и выгруженных в кассовый сервер

(исходный элемент находится в `/home/bat/Project/docs/Artix_Global/source/description.rst`, строка 44.)