# Приложение №1

# Техническое задание

# на продукт "Система Продаж 2.0"

**Цель продукта**: создание серверного приложения (сервиса продаж) осуществляющего создание и контроль сертификатов в электронном виде.

**Термины и понятия**:

*Сертификат* – электронный бланк, состоящий из предварительно загруженной подложки (фирменного бланка) и впечатанного программным способом данных.

*Сервис продаж* – набор специализированных микросервисных приложений выполняющих роль: создание продажи, генерации бланка сертификата, взаимодействие с агентами продаж посредством API, управление тарифами.

**Требования к технологиям**:

Сервер: Ubuntu 18.04

Язык: PHP 7.x

БД: MySQL >= v 5.7.x

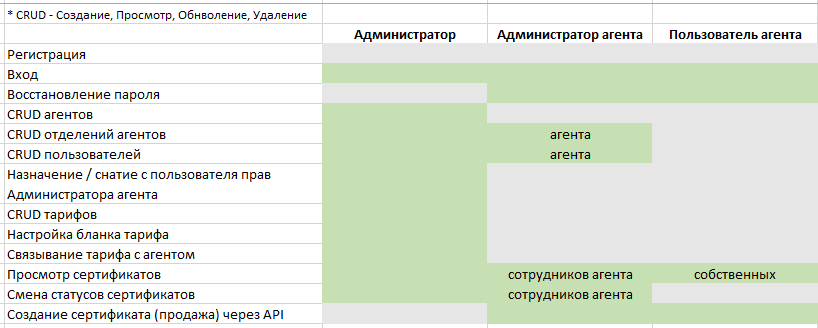
Framework: Symfony 3.4

**Требования к форматам генерируемых бланкам сертификатов**:

- jpg

- pdf

Схема прав пользователей (см. Excel)



**Администратор**

Создается через консоль на сервере, смена пароля также происходит через консоль (в целях безопасности).

**Пользователь | Администратор агента**

Пользователь создается Администратором после создания Агента.

Права Администратора агента назначаются путем установки галочки «Администратор агента» при создании пользователя. Смена роли для созданного пользователя на текущий момент не предусмотрена.

При создании пользователя на указанный email отправляется письмо со ссылкой для сброса пароля, перейдя по которой и установив пароль он будет иметь возможность доступа к своей учетной записи в системе согласно своих прав.

При создании пользователя ему назначается api\_key (используемый при запросах создания сертификата).

Api\_key доступен пользователю на странице Информации.

Пользователь может быть заблокирован. В данном случае он не сможет зайти в панель администратора и осуществлять оформление сертификатов.

Для пользователей используется SoftDelete при удалении (для восстановления в случае необходимости)

**Описание взаимодействия серверного приложения:**

Серверное приложение должно общаться с агентами продаж через специально выделенный url вида: [http://url-сервиса/api/<код](about:blank) агента продаж>/

**Описание состава серверного приложения**:

Каждое ядро реализуется в отдельном **bundle**.

1. **Ядро Rest** – отвечает за роутинг, прием запросов и отправку ответов. Является в архитектуре входным/выходным шлюзом.
2. **Ядро Продаж** – отвечает за генерацию сертификатов и сопоставлении продажи (сертификата) с пользователем агента продажи.
3. **Ядро Тарифов** – отвечает за создание и наполнение тарифа, а также за связи его с агентом продаж. Тариф является "родительской" сущностью для продажи.
4. **Ядро Агентов** – отвечает за создание сущности агента продажи, а также за сущности пользователей, которые ассоциируются с агентом продаж. В частности соотношение сущности агента продажа к сущности пользователя "один-ко-многим".
5. **Ядро API** – набор команд, которые позволяют создавать продажу, изменять свойства продажи.

# Ядро Rest

Предусмотреть варианты отправки ответов в зависимости от настроек в конфигурационном файле: предусмотреть ответы в 2-х форматах: json и xml, для этого использовать конфигурацию FosRestBundle.

Чтобы клиент мог управлять самостоятельно, в каком формате он хочет получать ответ, то в заголовке необходимо указывать:

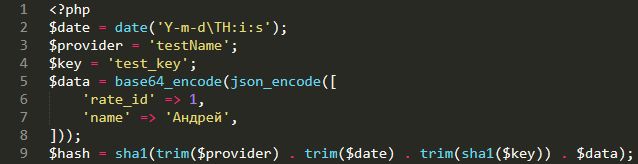
Accept: application/xml или Accept: application/json

Режим Security будет осуществляться в 2-х факторном режиме:

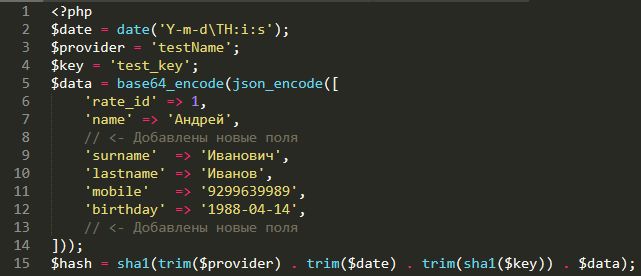
* фильтрация по IP-адресу, путем проверки в конфигурационном файле security.yml в разделе access\_control параметром ips;
* проверкой api\_password с пользователем агента продаж.

Ядро **Rest** выполняет только 1 команду на прием/создание продажи и отвечает на данный запрос ответом в виде xml файла.

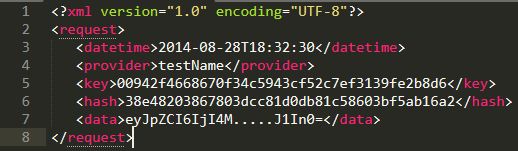
Рассмотрим протокол формирования запроса на создание продажи (генерации сертификата) к ядру **Rest** со стороны **Пользователя** продаж:



В переменной provider хранится логин **пользователя**, а в функцию sha1() передается api\_key. rate\_id - идентификатор тарифа сущность **Тариф,** по нему производится идентификация тарифа, по которому необходимо будет сформировать продажу. Поле: ‘name’ имя, которые отправляется и будет впоследствии заполнено в соответствующее поле бланка сертификата см. модуль генерации бланка сертификата (сколько мы отметим полей в форме, столько параметров должны ожидать для генерации продажи). В частности, если партнеру необходимо, чтобы в продаже учитывались иные поля т.к. фамилия, имя, отчество и др. то необходимо предусмотреть в разделе редактирования Тарифа следующий раздел отображения полей, и если указанные поля были отмечены, то поля должны быть обязательными к заполнению продажи и должны попадать в конструкцию:



Далее получившийся hash подставляется в структуру xml документа в раздел hash, а закодированные base64\_encode в поле data xml (см. пример)



После того, как пользователь на url отправляет методом POST xml, он будет ожидать соответствующий код

В ответ на запрос партнера сервер формирует **XML-документ**, содержащий̆ статус обработанного запроса (успешен или ошибочен). Ответ так же снабжается датой обращения и **хешем**, генерируемым на основе уникального секретного ключа партнера, во избежание искажений или подмены информации.

В одном из ваших писем был вопрос касаемо hash-ответа: все верно, вы формируете ответ см. таблицу ниже пометил строку зеленым маркером.

Спецификация кодов ответов:

| **Код ошибки** | **Описание** | **Комментарий** |
| --- | --- | --- |
| 10 | Batch request has errors | Пакетный запрос содержит ошибки Пустой запрос |
| 20 | Empty request | Пустой запрос |
| 30 | Incorrect XML | Неверный формат XML-документа |
| 35 | Incorrect agent code | Неверный код агента |
| 40 | Incorrect provider name | Неверное имя провайдера |
| 45 | Incorrect secret key | Неверный ключ |
| 50 | Неверный ключ | Пустой элемент "datetime" |
| 51 | The value of "datetime" should be in "YYYY-MM-DD hh:mm:ss" format | Неверный формат переданной даты |
| 60 | Impossible time value in "datetime" | Несуществующая дата |
| 70 | Empty value in "data" | Пустой элемент «data» |
| 80 | Value in "data" must be in Base64 | Данные в элементе «data» не кодированны в Base64 |
| 90 | Empty value in "hash" | Пустой элемент «hash» |
| 100 | Check of hash failed | Ошибка проверки хеша |
| 200 | Wrong data sale | Ошибка данных для формирования/обновления продажи |
| 210 | Empty “number\_prefix” | Пустой элемент “number\_prefix” |
| 211 | Incorrect "number\_prefix" | Неверное значение "number\_prefix" |
| 220 | Empty "number" | Пустой элемент “number” |
| 221 | Incorrect "number" | Неверный элемент “number” |
| 230 | Empty "activation\_code" | Пустой элемент "activation\_code" |
| 231 | The value of "activation\_code" should be in numeric format and length should be 6 digits | Неверный формат кода активации. Формат кода активации — 6 цифр. |
| 240 | Empty "status" | Пустой элемент "status" |
| 241 | The value of "status" should be "active", "reserv" or "terminated" | Значение «status» должно быть только "active", "reserv" или "terminated" |
| 250 | Empty Rate | Пустой тариф |
| 270 | The value of "created\_at" should be in "YYYY-MM-DDThh:mm:ss" format | Неверный формат даты создания продажи |
| 290 | Wrong data in data field | Неверный список полей для генерации сертификата |
| 500 | Unknown error | Неизвестная системная ошибка |
| 510 | Forbidden Deny | Ограничение по правам (попытка добавления продажи к не своему тарифу и агенту) |
| 511 | Requested rate is forbidden | Запрошенный тариф не доступен (ограничение по правам) |
| 560 | Failed to generate image | Ошибка генерации сертификата |

Ответ следует формировать следующим образом: 

Пояснения по xml-ответу

| **Элемент** | **Описание** | **Пример заполнения** |
| --- | --- | --- |
| *response* | Корневой элемент XML-документа. | <response>  <status>0</status> <provider>test-name</provider> <datetime>2017-08-  22T16:05:18</datetime> <hash>8a7c809defc06a6b8a2bd4d5d0ad690 88e548618</hash>  <number>test-1010735</number>  <images> <item>https://domain.ru/uploads/certificates /1503407118default-test- 1010735.jpg</item>  </images> </response> |
| *datetime* | Дата и время обращения, в формате UNIX timestamp. | 2014-08-28T18:32:30 |
| *status* | Статус запроса. Содержит код обработки: 0 — запрос успешен, иначе код ошибки | 0 |
| *error* | Описание ошибки, обнаруженной при проверке запроса. Опционально. При валидном запросе элемент отсутствует. | Value in "data" must be in Base64 |
| *images* | Ссылки на изображения сформированные по шаблонам (бланк сертификата, памятка и т. д. В зависимости от настроек тарифа) | <images>  <item>https://domain.ru/uploads/certificates /1503407118default-test- 1010735.jpg</item>  </images> |
| *errors* | Опционально. Появляется при пакетном создании продаж, в случае, если есть ошибки в данных, по которым создается продажа | <errors> <item>  <sale\_data>{"rate\_id":"260","name":"Иван" ,"surname":"Иванович","lastname":"Ивано в","created\_at":"2017- 00:00","mobile":"9299639989","number\_pr efix":"TEST"}</sale\_data>  <messages> <item>  <status>220</status>  <error>Empty "number"</error>  <hash>70876f2f582318940b6f026a7ba1f81 5a1e19c0b</hash>  </item> <item>  <status>270</status>  <error>The value of "created\_at" should be in  "YYYY-MM-DDThh:mm:ss" format</error> <hash>0244aeb39c7dc2ae32231fab76e03f3 68ab59f26</hash>  </item> </messages>  </item> </errors> |
| *hash* | Эквивалент контрольной суммы ответа, формируемый по алгоритму:sha1(<provider> + <datetime> + <secret key> + <status>),  где <secret key> - уникальный секретный ключ, известный только  контрагенту и поставщику услуги. | efb473f3c4d3e1de4bd7adcb874357fb24759 a96 |

**Логирование**

Реализована система логирования ошибок при обращении по API.

Детальные логи расположены в storage/logs/certificates/

# Ядро "Продаж"

Главной сущностью продажи является сертификат, который характеризуется следующими обязательными параметрами:

* статус заказа, могут быть следующие статусы (резерв, активирован, аннулирован, удален), (тип string, not null)
* полный номер сертификата состоящий из серии и номера разделенные "разделителем"
* связь один к одному с сущностью **Пользователь**
* связь с один к одному с сущностью **Тариф**
* дата создания
* стоимость (нельзя брать по связи от стоимости тарифа, при создании продажи происходит вычисление от опции "агентское вознаграждение", если оно имеет null или 0, то происходит копирование значения из тарифа. Это делается для того, что стоимость тарифа может со временем поменяться, и чтобы могли строить адекватную статистику по продажам, поэтому цена за тариф не должна перебивать стоимость на момент совершения продажи) (float not null). В нашем бизнес-кейсе Агентское вознаграждение равно 0.

При создании продажи статус заказа формируется, как "Активирован"

У сертификата есть 4 статуса, как уже было сказано выше. Все статусы при их создании будут иметь статус “Активирован”.

Пояснения по статусам: маркетинговый жизненный цикл статусов: Зарезервирован -> Активирован -> Расторгнут (и спец статус “Удален” - сертификат помечается как удаленный)

Статус и операция с ним несут на себе 2 смысловые нагрузки:

 Когда статус имеет значение “Активирован”, то он синхронизируется с системой колл-центра, и операторы видят, что он реализован клиенту и в любой момент клиент сможет позвонить;

 Второе значение статусов, если предусмотрен механизм Резервирования (т.е. клиент заявил о

намерении его приобрести, то сертификат резервируется с указанным номером) и когда клиент внес денежные средства, то статус меняется на Активирован и далее происходит его синхронизация с нашей внутренней системой.

**Генерация сертификата.**

Для каждой успешной продажи генерировать файл сертификата и хранить его в папке на сервере. Принцип наименования папок:

https://domain.ru/uploads/certificates/год/месяц/день/{серия сертификата}-{номер сертификата}.jpg

Либо вариант:

http://domain.ru/certificate/<хэш>

Создаются путем запроса по API. Все роли, кроме пользователя, имеют возможность изменять статус сертификата. Сертификат связан с Пользователем. При удалении пользователя Сертификат остается в системе без связи с каким-либо пользователем. При удалении агента сертификат остается без связи с агентом. При удалении тарифа – сертификат остается без связи с тарифом.

Реализована возможность получения поля number целочисленного и использования его в номере сертификата. При получении существующего в системе поля number – происходит возврат информации о созданном сертификате.

Стоимость сертификата формируется как Тариф + Агентское вознаграждение. Если установлена галочка Процент – Стоимость = Тариф + (Тариф\*Агентское вознаграждение) / 100

Изображения генерируются при создании сертификата и хранятся на сервере. В целях безопасности путь в ссылке к ним скрыт.

# Ядро Тарифов

Тариф – это именованный информационный продукт. Сущность тарифа имеет непосредственную связь с Агентом продаж.

Создаются Администратором системы. После создания тарифа необходимо произвести его связь с Агентами и Настроить бланк. Если тариф не связан с Агентом – он будет недоступен пользователям Агента при попытки создания сертификата.

Для тарифов используется SoftDelete при удалении (для восстановления в случае необходимости)

Редактирование бланка тарифа расширено относительно ТЗ в рамках удобства: добавлено поле Тестовое содержимое (отображается только при редактировании бланка, служит для использования максимально реальных значений в полях для установки размеров); добавлено поле Цвет шрифта, в котором должны быть значения в формате #000000 - #FFFFFF (HEX)

Реализована логика с Обязательными полями и Значением по умолчанию. Все поля обязательны. Выбранные поля обязательны, если поле отмечается как необязательное – необходимо ввести его значение. Если поле отмечено как необязательное – оно может отсутствовать в запросе по API (тогда будет использовано значение по умолчанию). Значением по умолчанию может быть пустая строка – тогда поле не будет отображено.

Параметры сущности **Тариф**:

* наименование тарифа (тип string – not unique, not null)
* серией (может быть любой длины и содержать в себе, как буквы, так и цифры) (тип string –unique, not null)
* номер (состоит только из цифр, и длина номера может быть любой). При генерации номера происходит последовательное увеличение на 1. (При взаимодействии по API номер может генерироваться как последовательно, так и создаваться тем номером, который указан в поле number)
* код тарифа набор букв и цифр с допустимыми знаками "\_" (string unique not null)
* ссылка на файл бланк тарифа (изображение формата png, jpeg) (string default null)
* стоимость тарифа (float default 0.00)
* агентское вознаграждение (float default null)
* опция "Процент" к агентскому вознаграждению. Если установлен в 1 то значение "Агентское вознаграждение" вычисляется как % от стоимости тарифа (tinyint default 0)

Модуль генерации бланка тарифа:

За основу берется ссылка на файл (см. параметр сущности **Тариф**)

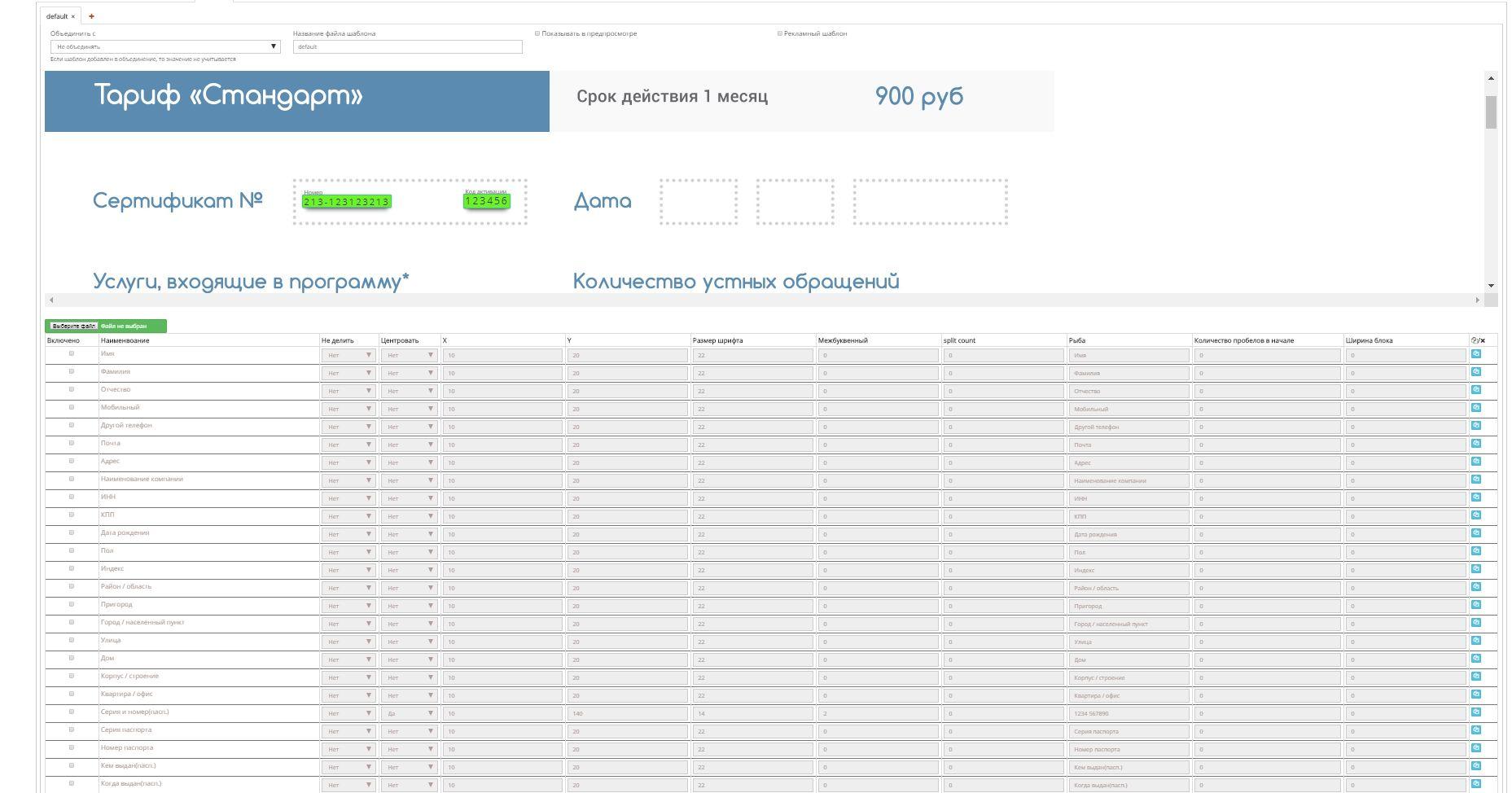
Суть работы модуля, он представляет собой интерфейс, состоящий из 2-х блоков:

Первая часть (визуальная), вторая – форма.

В визуальной части отображается загруженный файл-изображение на который накладываются элементы, которые отмечаются опцией checkbox из формы.

В части отображения формы параметризованная форма.

См. пример редактора ниже:



Форма состоит из предустановленных полей (которые не активны, по умолчанию они все деактивированы, при их активации опцией checkbox появляется соответствующее поле в визуальной части в левом верхнем углу, и далее его можно перемещать при помощи мыши):

* Имя (код name)
* Фамилия (код *lastname*)
* Отчество (код *surname*)
* Мобильный (код *mobile*)
* Пол (код *gender*)
* Адрес (код *address*)
* Номер (см. раздел Продажи – это полный номер сертификата) (код *number*)
* Код активации (при создании продажи создается random число 6 разрядов)

Каждое поле характеризуется следующими свойствами:

* размер шрифта (font\_size)
* Наименование отображаемое в визуальной части (is\_available)
* код поля (допускаются латинские буквы, цифры знак “\_”)
* координаты по Х в пикселях (при перемещении элемента в визуальной части при помощи мыши, вычисляется положение верхнего левого прямоугольника объекта и записывается в соответствующую координатное значение)
* координаты по Y в пикселях (при перемещении элемента в визуальной части при помощи мыши, вычисляется положение верхнего левого прямоугольника объекта и записывается в соответствующую координатное значение)

В общих настройках тарифа предусмотреть выбор расширения, в котором необходимо будет сгенерировать сертификат. Будет выпадающий список из значений на выбор:

* jpg
* pdf (значение по умолчанию)

Чтобы в дальнейшем было удобнее кастомизировать модуль рекомендуется свойства формы первоначальные сохранить, как поля таблицы, т,е. назовем для примера сущность модуля tariff\_form\_properties. Полями сущности будут: свойство активности (is\_available), размер шрифта - font\_size, код поля – code (Код поля отвечает за название, которое будет участвовать в обмене в модуле Rest, к примеру у Фамилии код будет lastname), и X и Y соответственно (отвечают за положение поля на бланке Тарифа по соответствующим координатам). А записями сущности будут: имя, фамилия, отчество и т.д. см. выше предустановленные поля формы. Таким образом, при кастомизации формы нам достаточно будет добавить новую запись в данную таблицу.

После того, как будет произведена настройка тарифа и его визуальной части сохраняем настройки отображения полей на визуальности части в связанной сущности tariff\_certificate\_view\_properties в которой храним связь тарифа, идентификатора поля свойства из сущности tariff\_form\_properties и дополнительным полем json настройку для соответсвующего поля. Например:

Имеется тариф: “Забота” с идентификатором id: 1 в сущности tariff. После настройки бланка сертификата мы приняли решение отобразить на нем поле “Имя” в которое подставится значение по коду name. Таким образом запись сохраниться в сущности: tariff\_certificate\_view\_properties в следующем виде:

|  |  |  |  |
| --- | --- | --- | --- |
| id | tariff\_id | tariff\_form\_properties\_id | tariff\_form\_properties\_field\_proprty |
| 1 | 1 | 1 | [“font\_size”: “12”, “is\_available”: 1, “code”: “name”, “x”: 1023, “y”: 500] |

Получается, что при совершении продажи формируется бланк сертификата, на который впечатывается имя клиента (это имя передается через соответствующее поле в API name) имеет размер шрифта 12px и располагается по указанным координатам на подложке.

В разделе **Ядро Rest** были внесены дополнения, связанные с отправляемым форматом данных. В частности требования к тарифу могут быть:

* к набору полей
* управление (не-)обязательностью заполнения полей
* установление значений по умолчанию, в случае если поле (необязательно)

Для этих целей необходимо создать форму со следующими полями (смотрите скрин ниже):

1. Поле с предопределенными полями (необходимо чтобы поля совпадали с полями, которые отображаются в визуальном редакторе Тарифа, и в случае, когда необходимо будет добавить новое поле, то добавляем поле в таблицу и оно попадает, как в редактор тарифа, так и в редактор формы продажи):

− Имя (код name)

− Фамилия (код lastname)

− Отчество (код surname)

− Мобильный (код mobile)

− Пол (код gender)

− Адрес (код address)

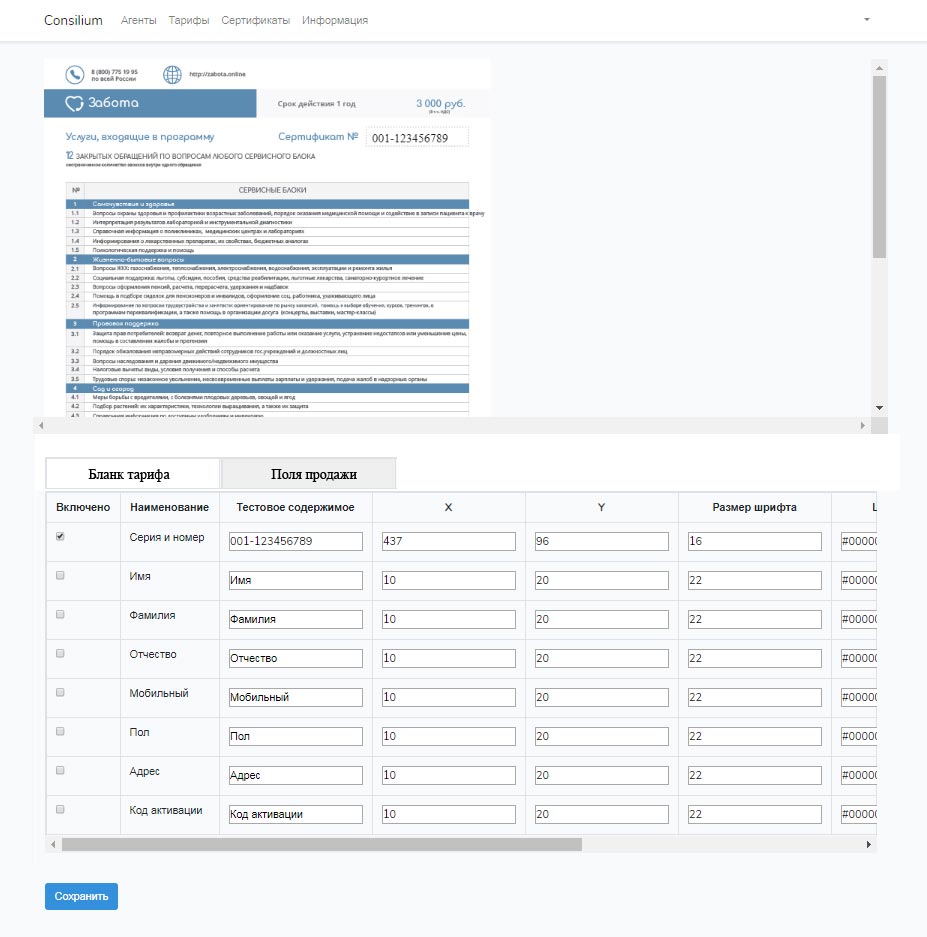
1. Поле с управляющим элементом типа Checkbox отвечающее за (не-)обязательное заполнение данного поля (аналог атрибута required в input)
2. Значение по умолчанию
3. Порядок отображения (это когда будем работать над визуальным редактором формы продажи)

Пояснения к полям "Бланк Тарифа" и "Поля Продаж":

Вкладка "Поле Продаж" повторяет набор полей, которые присутствуют в визуальном редакторе Бланка тарифа. Повтор полей связан с тем, что у каждого поля имеется код поля (и по нему будет связываться эти поля между вкладками), а также тем, что при создании продажи некоторые данные нужно будет "впечатывать" в сам бланк.

Для примера, один из партнеров захочет при создании продажи впечатывать ФИО клиента в бланк, чтобы сертификат был именным. С учетом текущего функционала это сделать не получится, т.к. мы на данный момент отображаем только сам сгенерированный номер сертификата и код активации. Поэтому будет реализована вкладка "Поля продажи", и теперь в случае создании продажи при отмеченном соответственно в разделе "Поля продажи" и "Бланк тарифа" поле (именно в обоих разделах): Фамилия Имя Отчество движок продажи должен понимать, что отмечены поля в обеих вкладках и нужно данные перенести в визуальную часть бланка.

Поля Продаж необходимы при создании продажи. Чтобы понимать смысл, стоит ассоциировать данный раздел с формой продажи, поля которой как раз будут полями из раздела "Поля продаж" со всеми атрибутами "обязательное поле к заполнению, длина поля, правила валидации". Стоит помнить о связи полей из раздела Поля Продаж и "Бланк Тарифа" по коду поля, например: name

­

При создании тарифа заполняются поля Серия и Код тарифа, их назначение:

**Серия тарифа**: буквенно-числовой набор. Смысл серии в том, чтобы операторы колл-центра моментально по серии смогли сориентироваться, по какому продукту звонит клиент, называя идентификационные данные сертификата.

Требования к серии: может быть как буквенной, так и числовой любой длина считаем не более 10 знаков. Не уникальное поле!

**Внимание**: у разных агентов могут быть одинаковые серии (однако мы такого стараемся не допускать, как правило, мы выделяем одинаковую серию для тестовых сред). ОДНАКО: стоит учитывать, что начало и конец номера не должны пересекаться, даже у разных агентов, иными словами, если сложить все начало-конец с одной серией у всех агентов, то получится сплошной диапазон с непересекающимися отрезками.

**Код тарифа**: необходим для дальнейшей синхронизации с нашими внутренними системами.

Требование к коду тарифа: латинские буквы с цифрами и знаком подчеркиванием, ограничение по длине 255 символов, УНИКАЛЬНОЕ ПОЛЕ!

Номер сертификата состоит из серии и номера.

Пояснение: серия сертификата проставляется Администратором (требования к серии см. Выше в

разделе Тарифы). Стратегия формирования номера 2 способа:

* Последовательное увеличение номера на 1;
* Присваивается тот номер, которые был прислан в API, если номер такой уже есть в системе, то просто происходит возврат ссылки на бланк сертификата (и при наличии полей они обновляют поля сертификата)

# Ядро Агентов

Сущность **Агент (agent)** характеризует в предметном мире организацию и повторяет ее структуру и имеет следующие поля:

* Название Агента (string 255, not null)
* Код агента (string 255, not null, unique)

Связанная сущность **agent\_tariff** хранит связь агента с Тарифами агента

Сущность **Пользователь** связывается с сущностью **Агент** “много-к-одному”, т.е. много пользователей Банка являются сотрудниками Банка **Агента** “ВТБ”.

Сущность **Пользователь (agent\_users)** характеризуется следующими значениями:

* логин или provider(string 255, not null, unique)
* пароль (string 255, not null)
* Имя (string 255, not null)
* Фамилия (string 255, not null)
* Отчество (string 255, allow null)
* api\_key (string 255, not null, unique)
* отделение организации если есть (связь с отделениями Агента по-умолчанию null)
* связь с сущностью **Агент**

# Ядро Синхронизации

Ядро предназначено, после совершения успешного создания продажи, для синхронизации с внутренней ERP системы.

Необходимо в конфигурационном файле проекта зарезервировать URL, по которому будет происходить синхронизация.

В текущей настройке URL будет следующим, на который вам необходимо будет направлять запросы на синхронизацию: <http://217.21.222.36/index.php/bankapi>

Приведу пример работающего на данный момент класса в нашем сервисе синхронизации:

class **Erp** {

/\*\*

\* @var string URL получателя

\*/

private $recipientUrl;

/\*\*

\* @var $senderObject

\*/

private $senderObject;

/\*\*

\* @param $config

\* @param $senderObj

\*/

public function \_\_construct($config, $senderObj)

{

$this->recipientUrl = $config['erpURL'];

$this->senderObject = $senderObj;

}

/\*\*

\* @param $packets

\*/

public function sendERP($packets)

{

// Header for the ERP engine

$post = array(

'action' => 'activate',

'moment' => date('Y-m-d h:i:s'),

'hash' => md5(date('Y-m-d h:i:s')),

'multi' => 'multi',

'params\_multi' => array()

);

$post['params\_multi'] = array($packets);

$post['hash'] = md5($post['action'] . $post['moment'] . implode(array\_map('implode', array($packets))) . '}\_ID]cmO/zu@&K^r');

$resp = $this->senderObject->get('buzz')->post(

$this->recipientUrl,

$headers = array(),

json\_encode($post)

);

}

/\*\*

\* @param $certNum

\* @param $tariff

\* @param $actCode

\* @return array

\*/

public function prepareERPpacket($certNum, $tariff, $actCode)

{

return array(

'promocode' => 'broker',

'number' => $certNum,

'tariff' => $tariff,

'sale\_id' => 1,

'password' => $actCode,

'name' => 'Иван',

'lastname' => 'Иванов',

'surname' => 'Иванович',

'cell\_phone' => '+7 (999) 999-99-99',

'phone' => '',

'email' => '',

'address\_order' => '',

'duration\_from' => date('Y-m-d') . "00:00:00",

'finish\_at' => null

);

}

}

Пояснения по коду:

$recipientUrl – это <http://217.21.222.36/index.php/bankapi>

$senderObject – инстанс сервис-контейнер из Symfony

Пояснения для метода **prepareERPpacket**

number – серия и номер без разделителя (например если сертификат был 011-000123, то стоит направлять 011000123)

tariff – код тарифа (например test\_tariff\_code)

password – код активации

В случае, если в продаже участвуют дополнительные поля см. **Ядро Тарифов** вкладка "Поля продаж", которые отмечены в качестве обязательных для создания продажи, то их стоит передавать.

Ответ будет предоставлять собой json-строкувида:

{    
   **"answercode"**:0,  
   **"answer"**:"ok",  
   **"moment"**:"2019-05-15 14:03:10",  
   **"hash"**:"9b36f76e474f692e7e4ead8e92bb1c4d"  
}

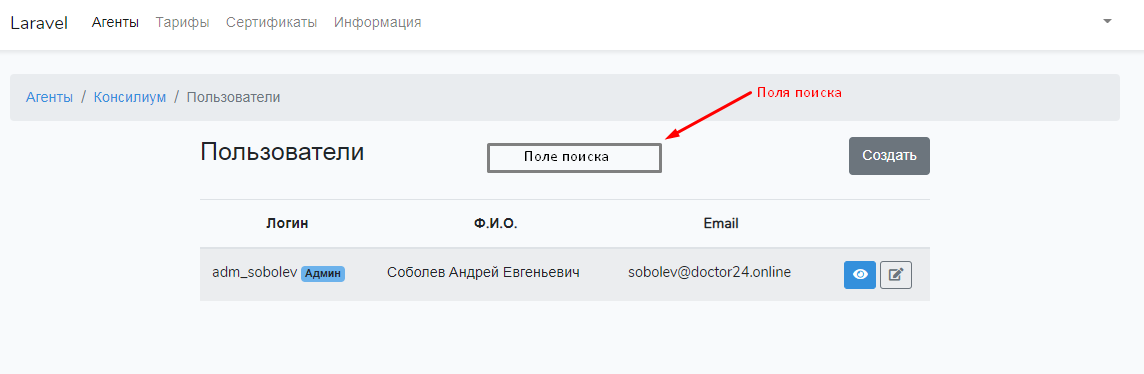
Коды ответов:

|  |  |
| --- | --- |
| **Код ответа** | **Описание ответа** |
| 0 | Операция успешно произведена |
| 10 | err\_unknown |
| 15 | Отсутствует тело запроса input |
| 20 | Ошибка в JSON структуре |
| 200 | err\_main |
| 201 | Неверный тип mode, если это поле отправляется. Используется для тестирования и имеет единственное значение **debug** |
| 202 | Отсутствует поле action, нужно отправлять **activate**(в нашем случае).  Может принимать следующие значения: 'activate', 'block', 'reserve', 'unreserve', 'check' |
| 203 | Отправлен неизвестный тип action, должен быть: 'activate', 'block', 'reserve', 'unreserve', 'check' |
| 204 | Пустое значение поля **moment** |
| 205 | Неверное значение поля **moment** |
| 206 | Пустое поле **hash** |
| 207 | Процедура проверки **hash** прошла с ошибкой (неверно сформирован hash) |
| 208 | Поле **params** пустое |
| 400 | err\_param\_ |
| 401 | Пустое поле **number** |
| 402 | err\_param\_invalid\_ |
| 403 | err\_param\_unknown\_ |
| 405 | err\_param\_multi\_ |
| 500 | err\_activation |
| 501 | err\_activation\_number\_not\_found |
| 502 | err\_activation\_number\_seems\_activated |
| 503 | err\_activation\_number\_sold |
| 550 | err\_block |
| 551 | err\_block\_number\_not\_found |
| 552 | err\_block\_number\_reserved |
| 553 | err\_block\_number\_already\_blocked |
| 554 | err\_block\_card\_has\_unknown\_tariff |
| 600 | err\_reserve |
| 601 | err\_reserve\_number\_already\_reserved |
| 602 | err\_reserve\_number\_occupied |
| 603 | err\_reserve\_unknown\_problem |
| 650 | err\_unreserve |
| 651 | err\_unreserve\_number\_not\_found |
| 652 | err\_unreserve\_number\_not\_reserved |
| 700 | err\_multi |
| 701 | При наличии поля **multi** если отсутствует поле **params\_multi** или оно пустое |

# Дополнения

Дополнения к доработке **v.1.1.** Далее по тексту + таблицей обозначу приоритет по разработке.

* + - 1. В разделе редактирования пользователя отобразить поле "API ключ" с возможностью вносить свой ключ и кнопку с возможностью перегенерировать ключ. СРОЧНО 1 этап.
      2. В разделе пользователей агента необходимо см. скрин ниже: НЕ СРОЧНО 2 этап



Поиск должен быть по полям: **ФИО** и **Email**

- в столбец **Логин** отобразить значение из email.

- добавить столбец **Роль** (в нашем случае Admin тоже является ролью, далее будут каждому пользователю назначены дополнительные роли)

- реализовать пагинацию в разделе пользователей(30 пользователей на страницу)

* + - 1. Расширить функционал отображения сертификатов СРОЧНО 1 этап.:

Ниже элементов обведенных красным квадратом необходимо реализовать либо поле для поиска (для номера сертификата, пользователя), либо выпадающий список с предопределенными значениями (агент и тариф – т.к. их будет ограниченной значение то выведем их в селектор). Суть введения указанных контроллов в том, чтобы фильтровать продажи.

Необходимо добавить столбцы:

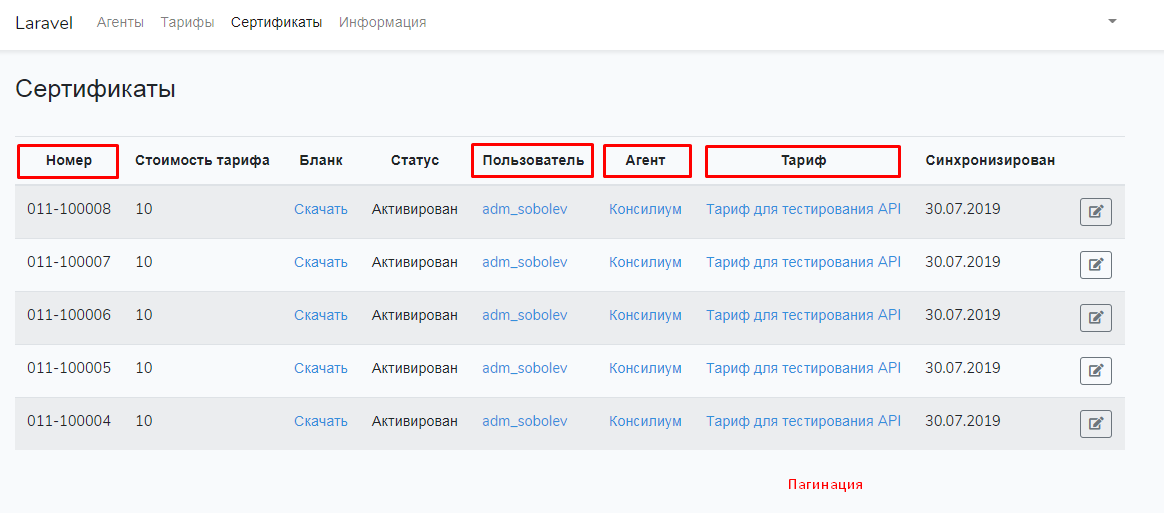
- Когда создан (с календарем, с возможностью выбирать дату за которую отобразить продажи).

Необходимо убрать столбцы:

- Синхронизирован

- Бланк (функционал скачивания вынести в раздел просмотра/редактирования бланка сертификата).

Реализовать пагинацию продаж. Считать 30 продаж на страницу. Отображать всегда продажи отсортированные в порядке от самых свежих продаж к поздним.



* + - 1. Функционал просмотра/редактирования сертификата НЕ СРОЧНО 2 этап.

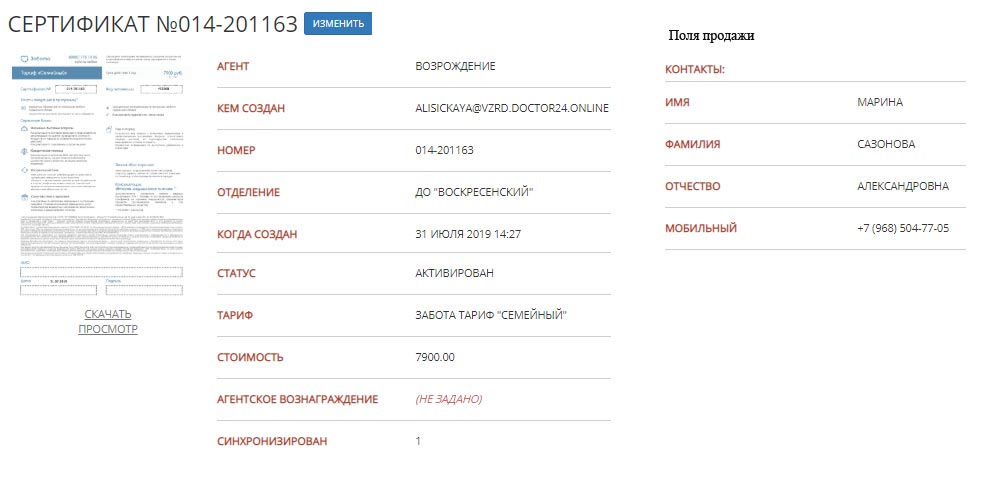
При редактировании сертификата отобразить:

- превью бланка

- параметры продажи (среднее информационное поле)

- поля продажи (правое информационное поле)

Ниже представлен пример раздела просмотра сертификатов.



В примере просмотра сертификата необходимо отобразить: "Код активации" (поле разместить ниже номера).

Раздел редактирования сертификата:

Данные раздел должен учитывать поля, отмеченные в разделе Тарифов "Поле продажи" + отобразить стоит следующие контроллы для ввода изменений:

- Отделение;

- Дата создания;

- Статус;

- Продавец (пользователь);

- Имя (обязательное поле, если даже оно не было отмечено в "Поле продажи");

- Фамилия (обязательное поле, если даже оно не было отмечено в "Поле продажи");

- Отчество (обязательное поле, если даже оно не было отмечено в "Поле продажи")

Отправка email сообщений

**Цель задачи**: отправлять на email письмо с определенной темой и текстом в теле письма с вложением в него созданного сертификата.

**Описание задачи**:

Сертификат создается в соответствии с требуемыми полями в разделе создания/редактирования Тарифа (данный раздел назовем «Редактирование бланка тарифа"). Обязательное поле для генерации сертификата включается через 2 опции: «Включено» и «Поле продажи». В раздел полей необходимо добавить еще одну опцию «Email», которая подчиняется все тем же принципам, однако этому полю добавляется еще одно правило (см. ниже).

В рамках текущего и последующих улучшений необходимо будет продумать и разработать  функциональный элемент интерфейса, который будет отвечать за настройки самого тарифа (Назовем его "Настройки тарифа").

С учетом того факта, что текущий раздел редактирования бланка тарифа больше относится к его визуальной части и отчасти к обязательности заполнения полей при создании сертификата, новый раздел будет отвечать за организационные моменты касательно характера самого тарифа.

Таким образом, новый раздел необходимо будет разделить на tabs. Первый tab будет именоваться «Уведомление клиентов».

В нем предусмотрим контролл имеющий 2 состояния: Вкл./Выкл. и с label к нему «Отправить письмо клиенту». Стоит учесть следующий момент, что при включении опции из данного раздела автоматически включает и делает обязательным полем продажи в разделе редактирования бланка тарифа (наоборот это правило не должно работать, т.е. включая поле и делая его полем продажи не означает, что письмо должно отправляться клиенту). При активации «Отправить письмо клиенту» становятся доступным раздел редактирования письма:

Раздел редактирования письма состоит из 3 контроллов:

- тема письма

- тело письма (предусмотреть редактор текста типа  WysiWyg в минимальном исполнении )

- контролл (checkbox) «Прикрепить к письму сгенерированный бланк сертификата» - соотвественно если опция выключена, то сертификат не прикрепляется.

Предусмотреть следующий обязательный момент. Чтобы в письмо можно было пробрасывать данные, собранные при создании сертификата в письмо необходимо разработать механизм "замещающих морфем» ниже пример для понимания:

Сертификат создается из 4-х обязательных полей - это ФИО и email. При этом мы активировали опцию «Отправить письмо клиенту». В заголовке письма мы пишем «Для вас подготовлен сертификат», в теле письма редактируем:

«Уважаемый, {name} {lastname}

Вами приобретен сертификат проекта Забота, он прикреплен во вложении

С уважением,

Забота, ООО Консилиум М.Д.»

Видно, что использовано в теле письма "Уважаемый, {name} {lastname}» морфемы name и lastname - они должны соответствовать кодам полей при редактировании Тарифа. В дополнение, необходимо еще разработать 2 морфемы: номер сертификат и код активации, это на случай, когда сертификат не будет отправляться во вложении, а мы будем краткую информацию о сертификате писать в теле письма.

# Пользовательская часть

## Создание продажи

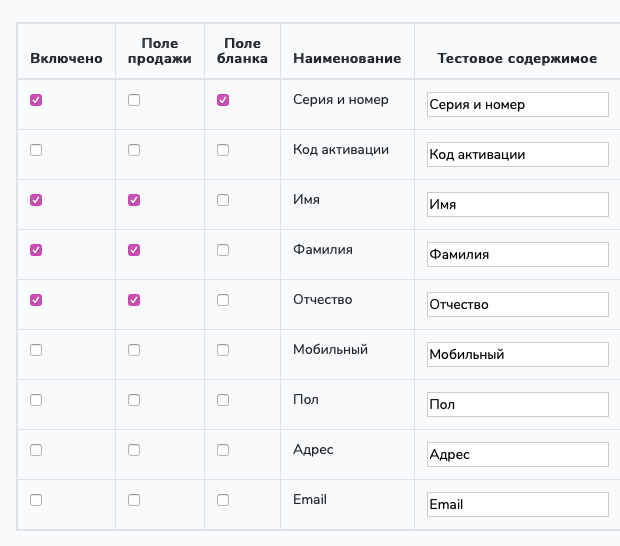
**Цель задачи** — реализация функционала по созданию и сопровождению проданного сертификата

**Описание задачи:**

Каждый пользователь агента имеет возможность по созданию продажи. Попасть пользователь в продажу через соответствующий пункт меню «Продажи» → «Создать продажу». Также стоит предусмотреть в пункте меню «Продажи» подпункт «Мои продажи» (описание см. в разделе «Права пользователя»).

При активации пункта меню «Создать продажу» пользователь попадает в форму для заполнения.

Первым элементом формы будет выпадающий список тарифов доступные для продажи пользователям агента (при этом если у тарифа установлен статус неактивный, то тариф нельзя отображать в этом выпадающем пункте). Необходимо реализовать дополнительный функционал для редактирования видимости тарифа пользователю и для API. Указанные пункты необходимо внести в раздел «Редактировать тариф» после поля «Процент» и назвать пункты, как «Видимость продавцам» и «Возможность продавать через API». При этом, если установлен чекбокс в контролле «Видимость продавцам» значит пользователь агента может продавать тариф и он появляется в выпадающем списке в форме раздела «Создать продажу». Соответственно контролл «Возможность продавать через API» также влияет на способность создавать сертификаты тарифа через API. Стоит учесть, что при создании тарифа по-умолчанию контролл «Видимость продавцам» будет не выставлен, дабы при создании тарифа не было казуса, что продавец случайно начал продавать тариф, который только еще наполняется. А контролл «Возможность продавать через API» наоборот по-умолчанию включен. Стоит учесть, что поля в форме динамические т. к. уже ранее был реализован графический механизм по настройке бланка сертификата тарифа и заранее был предусмотрен специальный раздел «Поля продажи» см.рисунок ниже.

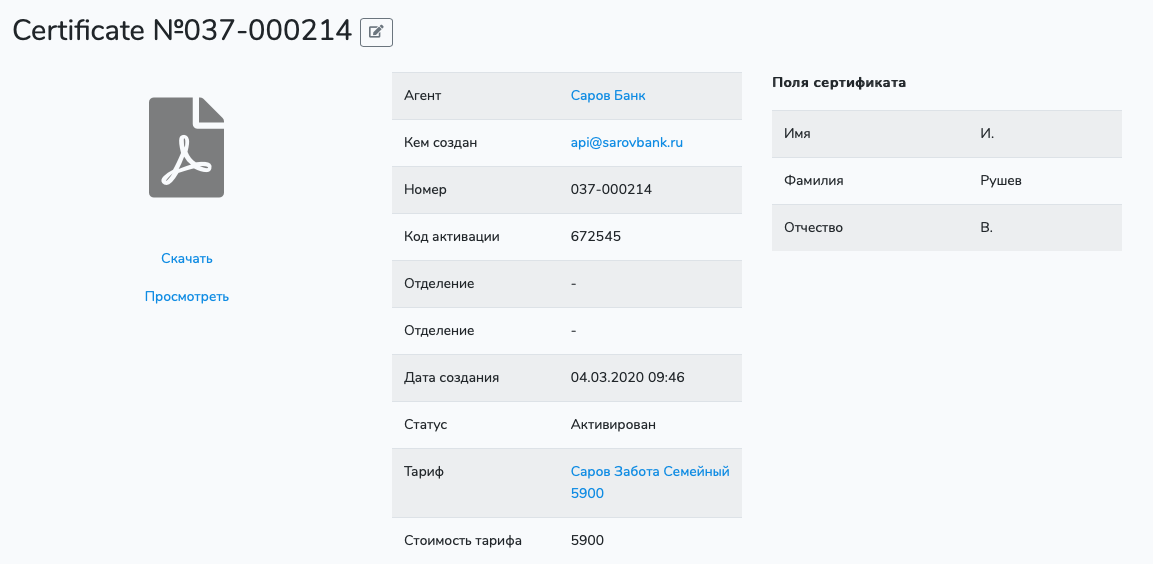
Таким образом, стоит учесть, что если установлена опция «Поле продажи», то это поле попадает в форму продажи. Стоит учитывать также поле «Обязательно» оно будет отвечать за атрибут «required»  для полей формы, иначе поле заполнятся значением при создании продажи из поля «Тестовое содержимое».

После нажатия на submit кнопку «Проверить продажу» пользователь попадает на страницу проверки введенные данных в форме, теперь данные будут уже выданы как статические поля без изменения, чтобы пользователь еще раз смог просмотреть, что он ввел. В данном разделе стоит предусмотреть 2 кнопки: «Вернуться к редактированию продажи» и «Создать продажу». Здесь все очевидно, первая кнопка снова возвращает пользователя в форму, однако все данные уже должны быть подставлены в форму (в том числе и должен быть выбран тариф из выпадающего списка), чтобы пользователь смог исправить неточности. Вторая кнопка уже создает сертификат тарифа со статусом «Резерв».

Если клиент нажал на кнопку «Создать продажу», то его перенаправляют на страницу с оповещением:

«Сертификат успешно создан! Однако он имеется статус Резерв. Чтобы изменить статус нажмите на кнопку Активировать».

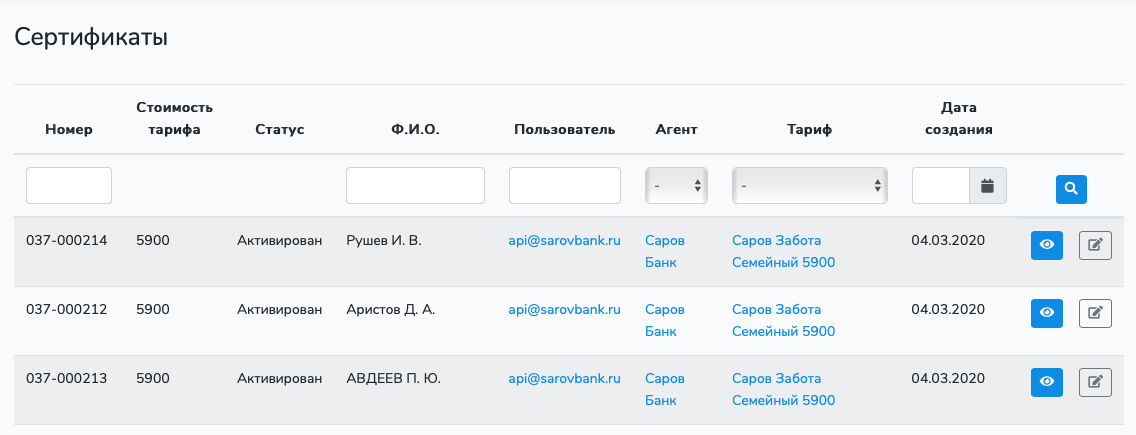
Вид страницы будет следующим:



Только в разделе Статус будет стоять статус «Резерв» и рядом с ним зеленая кнопка «Активировать». Приведенный выше пример имеет исчерпывающий вид доступный пользователю для просмотра.

**Права пользователя**:

* создать продажу
* посмотреть **СВОИ** продажи (аналогично функционалу администратора см.рисунок)
* изменить статус сертификата (статусы доступные пользователю для присваивания сертификатам: «Резерв» и «Активирован»)



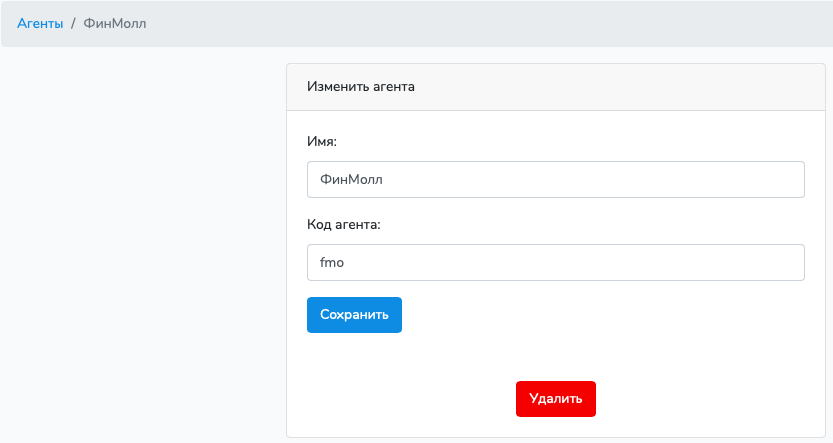
# Администраторская часть

Реализовать в администраторской части следующий функционал:

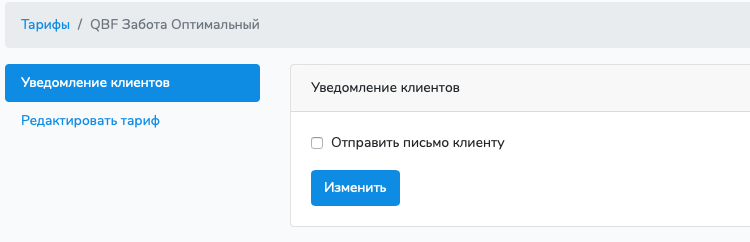
* механизм формирования иерархической структуры организации
* механизм экспорта пользователей агента

**Описание функционала**

**1. Механизм формирования иерархической структуры организации**



Реализовать табы для редактирования агента взять за пример, как реализовано при редактировании тарифа (см. пример скрина ниже, в приведенном примере смотрите только на прототип элементов, а не на названия).



Теперь первый пункт будет «Редактирование агента», а второй «Редактирование филиалов».

Уровень вложенности у филиальной сети предлагается развернуть в структуру таблицы БД:

* **Корень (root node)**.
* **1-й уровень** — (**catalog node** / региональный уровень)
* **2-й уровень** — (**child pages node** / уровень отделений/филиалов)

Пояснения по иерархии.

Если будет выбрана однотабличная реализация со структурой связи:

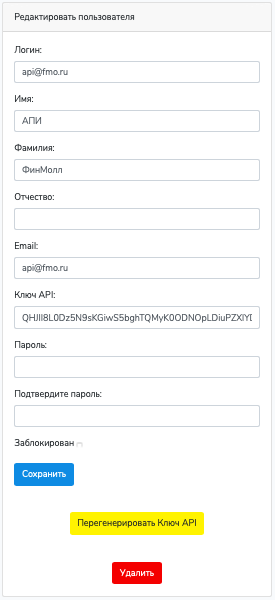
id и parentId (где parentId внешний ключ на id).

Корень - по умолчанию всегда присутствует для любого агента, если даже не предусмотрена иерархия или агент только завелся (то в таком случае имеет null).

При этом стоит учесть следующие поля для записей:

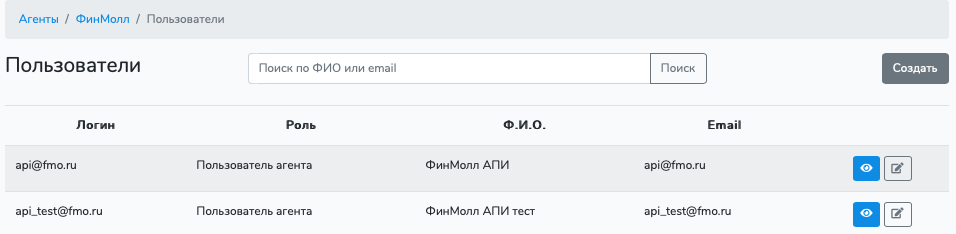
* название node (string 255 not null)
* код (string 255 unique not null)

А при редактировании пользователей агента чтобы появилось поле отделения, чтобы можно было уже сделать ассоциацию пользователя с каким-то отделом (внести отделение см.скрин ниже).



**2. Механизм экспорта пользователей агента**

Реализовать функционал массового заведения пользователей агента.

 Изменить поведение кнопки «Создать» чтобы при нажатии на нее появлялся выпадающий список, с 3-я опциями:

* Создать пользователям
* Экспорт пользователей
* Импорт пользователей (заложить, как будущий функционал, на данный момент реализовывать не нужно)

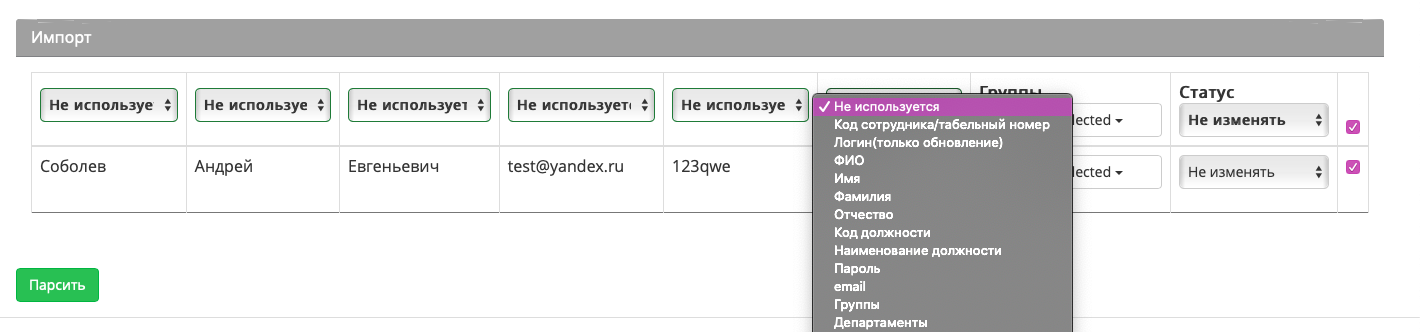
Экспорт пользователей будет происходить из файла с расширением csv в кодировке windows-1251.

При активации опции «Экспорт пользователей» попадаем на странице с формой со следущими контроллами:

* input file — для загрузки csv-файла с пользователями
* поле для ввода разделителей в csv файле (по умолчанию знак точка с запятой т.е. «;»)

После нажатия на кнопку экспорт производится анализ загруженного файла, но пока пользователи не загружаются в систему, т.к. необходимо произвести ассоциацию полей с необходимыми полями.

Для этого необходимо разработать следующий интерфейс ассоциаций:



На примере выше был загружен csv-файл со следующей структурой:

Соболев;Андрей;Евгеньевич;[test@yandex.ru](mailto:test@yandex.ru);123qwe;Москва

Разберем поэтапно процесс ассоциации поэтапно:

1. Разбор csv-файла

2. Количество найденных значений разделенных csv разделителем соответствует количеству столбцов в интерфейсе см.выше

3. Над каждым столбцом реализовать выпадающий список с заранее предопределенными значениями:

- ФИО (если в csv файле будет ФИО в одной ячейке разделенные пробелом)

- Фамилий

- Имя

- Отчество

- email (в нашем случае логин для авторизации)

- пароль (не путать с полем «Ключ API» у пользователя) При экспорте пользователей по списку данное поле не заполняется!!!

- код отделения (см. из раздела «механизм формирования иерархической структуры организации» было заложено в структуру). При выборе кода отделения при экспорте производим сверку наличия указанного кода отделения с имеющимися отделениями у агента, и если такого отделения нет, то не загружаем пользователя

- не используется (в этом случае поле не используем)

- статус пользователя (необходимо реализовать статус пользователя: Активный и не активный). Неактивному пользователю запрещается входить в систему.

4. Над каждым столбцом выбираем из выпадающего списка значение

5. Производим ассоциацию и заносим пользователей в систему.

Стоит отметить, что указанный механизм позволяет не только вносить новых пользователей, но и редактировать их массово, при этом стоит учитывать всегда email, т.к.только он обладает признаком уникальности, а все остальные поля при повторном экспорте будут перезаписывать значения.

|  |  |
| --- | --- |
| **Исполнитель**  ООО «Либертайм»  Республика Беларусь, г. Минск, пр-т Победителей 23, корп. 1, пом. 500Б  Р/с: BY82 ALFA 3012 2564 2800 3027 0000 (RUB) в ЗАО АЛЬФА-БАНК г. Минск, Ул. Сурганова, 43-47  SWIFT ALFABY2X  УНП 192294163  Тел.: +375 (17) 218 15 29  info@webspace.by  Генеральный директор  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ А.С. Аксютик | **Заказчик**  ООО «Консилиум М.Д.24»  Российская Федерация. Москва. 115114. Дербеневская наб. 11 этаж 1 офис 124 Почтовый адрес:  Российская Федерация, Москва, 115114. Дербеневская наб. 11 этаж 1 офис 124  Р/с 40702810602080002939  АО "АЛЬФА-БАНК", ИНН 7728168971, БИК 044525593  корреспондентский счет  № 30101810200000000593  в ГУ БАНКА РОССИИ ПО ЦФО  ИНН/КЛП 7725496540/772501001  ОГРН 1187746772124  тел. 8 (499) 681-8024  Генеральный директор  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ М.Ю. Лаврентьев |