# Задача на создание ресурсного метода микросервиса “ПродуктовыйУчет”

Метод создания продуктового регистра (ПР) через POST запрос: corporate-settlement-account/create

1. Глоссарий
2. [Назначение и цели](#Bookmark6)
3. [Соглашения и ограничения](#Bookmark7)
4. Описание процесса
5. [Свойства вызова](#Bookmark8)
6. [API](#Bookmark11)
   1. [URL](#Bookmark12)
   2. Формат запроса
   3. Формат ответа
   4. Коды и обработка ответов

# Глоссарий

|  |  |
| --- | --- |
| **Сокращение** | **Описание** |
| БД | База Данных |
| РО ЮЛ | Расчетное Обслуживание Юридического Лица |
| ПП | **Продуктовый Процессор** РО ЮЛ, логический процесс, трансформированный в системные объекты, определяющих операционную деятельность по РО ЮЛ |
| НСО | **Неснижаемый остаток**, продукт, оформленный, как Сделка к Генеральному Соглашению.  Либо, как Дополнительное Соглашение к Экземпляру Продукта (Договору) |
| ЕЖО | **Ежедневный Остаток,** оформляется, как Дополнительное Соглашение к Экземпляру Продукта (Договору) |
| СМО | **Среднемесячный Остаток**, оформляется, как Дополнительное Соглашение к Экземпляру Продукта (Договору) |
| ДБДС | **Договор без Дополнительного Соглашения**, оформляется, как Дополнительное Соглашение к Экземпляру Продукта (Договору) |
| ЭП | **Экземпляр Продукта, *отношение 1ЭП:NПР,***  объект, отвечающий за ЖЦ продукта в стадии обслуживания, содержащий информацию об актуальном состоянии параметров обслуживания продукта и актуальные остатки по требованиям и обязательствам. Эквивалентами данной сущности являются: ***договор***, в части финансовых (расчетных) параметров, а также сделки - НСО, ЕЖО, СМО, ДБДС.   * *Договор* может включать (не обязательно) финансовые (расчетные) параметры. * *Сделка* всегда связана с ЭП (договором). |
| ПР | **Продуктовый Регистр**. непосредственная часть ЭП, отвечающая за учет конкретных финансовых и нефинансовых требований и обязательств, ***отношение 1ПР:1Счет.***  Эквивалентами данной сущности являются лицевые (клиентские) и внутрибанковские счета. Экземпляр продукта должен иметь, как минимум, один продуктовый регистр. |
| ГС | **Генератор счета**, формирует параметры счета для ПР |

# Назначение и цели

Метод POST corporate-settlement-account/create предназначен для создания нового объекта ПРОДУКТОВЫЙ РЕГИСТР (ПР)

# Соглашения и ограничения

* Передача транзакции в БД ПП РО ЮЛ выполняется сразу всех созданных сущностей (ПР, остатки разных типов). Если в процессе алгоритма возникает любая ошибка, то транзакция в БД не происходит.

# Описание процесса

Чтение/запись в таблицы:

ЭП – Экземпляров Продукта (Договора), сортировка по ИД ЭП

ДС – Дополнительные Соглашения (сделки), сортировка по ИД ЭП

ПР – Продуктовый регистр (типы счетов, обслуживающих Экземпляр Продукта (договор), сортировка по ИД ЭП

КП – Каталог Продуктов заливается в БД, в дальнейшем обогащается по Кафке, сортировка по *ProductCode*

Для статуса 200 возвращается JSON с результатом

Для статуса 500 возвращается стек вызываемых процедур/функций + Текст (Ошибки)

Для статусов 400 и 404 возвращается Текст (Ошибки)

|  |  |  |
| --- | --- | --- |
| **№** | **Краткое описание** | **Логика и последовательность переходов и вызовов** |
| 1 | Выполняется анализ *Кода Продукта* экземпляра продукта, для которого необходимо создать новый продуктовый регистр.  В зависимости от кода продукта станет понятно:  Какого типа и сколько на регистре нужно остатков (возможны ссылки, как на счет Клиента, так и на внутрибанковский Счет).  Выполняется создание новой сущности Продуктовый регистр со статусом "Открыт" | **ШАГ 1.**  Проверка ***Request.Body*** на обязательность.  Если не заполнено хотя бы одно обязательное поле (см. ***Request.Body***)   * вернуть Статус: **400/Bad Request**, Текст: *Имя обязательного параметра <значение> не заполнено*.   Если все обязательные поля заполнены   * Перейти на Шаг 2.   **Шаг 2.**  Проверка таблицы ПР (таблица **tpp\_product\_register**) на дубли. Для этого необходимо отобрать строки по условию ***tpp\_product\_register.product\_id == Request.Body.InstanceID***и у результата отобрать строки по условию***tpp\_product\_register.type == Request.Body.registryTypeCode.*** Если результат отбора не пуст, значит имеются повторы  Если повторы найдены   * вернуть Статус: **400/Bad Request**, Текст: *Пар*аметр *registryTypeCode тип регистра <значение\_кода> уже существует для ЭП с ИД <значение\_ИД\_ЭП>*.   Если повторов нет   * Перейти на Шаг 3.   **Шаг 3.**  Взять значение из **Request.Body.registryTypeCode** и найти соответствующие ему записи в **tpp\_ref\_product\_register\_type.value**.  Если найдено совпадение   * перейти на Шаг 4.   Если совпадений не обнаружено   * вернуть Статус: **404/Not found**, Текст: *Код Продукта <значение> не найдено в Каталоге продуктов <схема.имя таблицы БД> для данного типа Регистра*   **Шаг 4.**  Найти значение номера счета по параметрам **branchCode**, **currencyCode**, **mdmCode**, **priorityCode**, **registryTypeCode** из **Request.Body** в таблице Пулов счетов (**account\_pool**). Номер счета берется первый из пула.  Сформировать новый продуктовый регистр и записать его в БД. При этом:   * id генерируется в СУБД * productid это **Request.Body.instanceId** * productRegisterType это **tpp\_ref\_product\_register\_type.value** * account – **account.id** * currencyCode это **Request.Body.currencyCode** * state - OPEN (enum (0, Закрыт/1, Открыт/2, Зарезервирован/3, Удалён)) * accountNumber – **account.number** |
| 2 | Выполняется отправка данных в систему-источник запроса на создание Продуктового регистра | Response.body должен содержать id созданного продуктового регистра в JSON установленной формы |

# Свойства вызова

|  |  |
| --- | --- |
| Тип взаимодействия | REST |
| Формат запроса/ответа | JSON |
| Тип вызова | Синхронный |
| Протокол взаимодействия | HTTP |

# API СЕРВИСА

## 6.1. [URL](#Bookmark12)

|  |  |
| --- | --- |
| Контур | URL |
| DEV | POST https:/сервер/port/corporate-settlement-account/create |
| ИФТ |  |
| НТ |  |
| ПРЕПРОД |  |
| ПРОД |  |

## 6.2. Формат запроса

***Request.Body***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Атрибут** | **Системное имя** | **Тип [Формат]** | **Обяз +**  **Опц -** | **Описание** |
|  | Идентификатор экземпляра продукта | instanceId | integer  sequence | + | Идентификатор ЭП, к которому привязывается продуктовый регистр |
|  | Тип регистра | registryTypeCode | string | - | Тип создаваемого продуктового регистра |
|  | Тип счета | accountType | string | - | Клиентский или внутрибанковский |
|  | Код валюты | currencyCode | string | - | 3-х значный код валюты |
|  | Код филиала | branchCode | string | - | Код филиала |
|  | Код срочности | priorityCode | string | - | Всегда «00» для ПП РО ЮЛ |
|  | Id клиента | mdmCode | string | - | МДМ код клиента (КЮЛ) |
|  | Код клиента | clientCode | string | - | Только для ВИП (РЖД, ФПК). Обсуждается с клиентом (есть выбор). |
|  | Регион принадлежности железной дороги | trainRegion | string | - | Только для ВИП (РЖД, ФПК) |
|  | Счетчик | counter | string | - | Только для ВИП (РЖД, ФПК) |
|  | Код точки продаж | salesCode | string | - | Код точки продаж |

## 6.3. Формат ответа

***Response.Body***

{

"data": {

"accountId": "string" // Идентификатор продуктового регистра

}

}

## 6.4. Коды и обработка ответов

Тексты ошибок указаны в разделе 4. Описание Процесса.

|  |  |
| --- | --- |
| Статус | Описание статуса |
| 200 OK |  |
| 400 Bad Request | Ошибка синтаксиса |
| 404 Not found | Данные не найдены |
| 500 Server Error | Ошибка сервера |