Руководство пользователя

электронного сервиса

Сервис информирования модуля безналичной оплаты питания и СКУД

Дата: 02.09.13

Версия: 1.4.1

Листов:

АННОТАЦИЯ

Данный документ является руководством пользователя электронного сервиса информирования. Сервис позволяет получать данные из модуля безналичной оплаты питания и СКУД.

СОДЕРЖАНИЕ

[1 Общие сведения 10](#_Toc377571531)

[1.1 Руководящие документы 10](#_Toc377571532)

[1.2 Описание электронного сервиса 10](#_Toc377571533)

[1.3 Операции (методы) электронного сервиса 10](#_Toc377571534)

[1.4 Сценарии использования и схема взаимодействия 13](#_Toc377571535)

[1.5 Связи с другими электронными сервисами 13](#_Toc377571536)

[2 Руководство пользователя 13](#_Toc377571537)

[2.1 Операция «Получение данных о лицевом счете» 13](#_Toc377571538)

[2.1.1 Общие сведения 13](#_Toc377571539)

[2.1.2 Описание входных параметров 13](#_Toc377571540)

[2.1.3 Описание выходных параметров 13](#_Toc377571541)

[2.1.4 Ошибки 14](#_Toc377571542)

[2.1.5 Контрольные примеры 14](#_Toc377571543)

[2.2 Операция «Получение данных о лицевом счете через идентификатор» 15](#_Toc377571544)

[2.2.1 Общие сведения 15](#_Toc377571545)

[2.2.2 Описание входных параметров 15](#_Toc377571546)

[2.2.3 Описание выходных параметров 15](#_Toc377571547)

[2.2.4 Ошибки 15](#_Toc377571548)

[2.2.5 Контрольные примеры 16](#_Toc377571549)

[2.3 Операция «Получение данных о покупках с лицевого счета» 17](#_Toc377571550)

[2.3.1 Общие сведения 17](#_Toc377571551)

[2.3.2 Описание входных параметров 17](#_Toc377571552)

[2.3.3 Описание выходных параметров 17](#_Toc377571553)

[2.3.4 Ошибки 17](#_Toc377571554)

[2.3.5 Контрольные примеры 18](#_Toc377571555)

[2.4 Операция «Получение данных о пополнениях лицевого счета» 19](#_Toc377571556)

[2.4.1 Общие сведения 19](#_Toc377571557)

[2.4.2 Описание входных параметров 19](#_Toc377571558)

[2.4.3 Описание выходных параметров 19](#_Toc377571559)

[2.4.4 Ошибки 19](#_Toc377571560)

[2.4.5 Контрольные примеры 20](#_Toc377571561)

[2.5 Операция «Получение информации о меню столовой» 21](#_Toc377571562)

[2.5.1 Общие сведения 21](#_Toc377571563)

[2.5.2 Описание входных параметров 21](#_Toc377571564)

[2.5.3 Описание выходных параметров 21](#_Toc377571565)

[2.5.4 Ошибки 21](#_Toc377571566)

[2.5.5 Контрольные примеры 22](#_Toc377571567)

[2.6 Операция «Получение данных посещениях образовательного учреждения» 27](#_Toc377571568)

[2.6.1 Общие сведения 27](#_Toc377571569)

[2.6.2 Описание входных параметров 27](#_Toc377571570)

[2.6.3 Описание выходных параметров 27](#_Toc377571571)

[2.6.4 Ошибки 27](#_Toc377571572)

[2.6.5 Контрольные примеры 28](#_Toc377571573)

[2.7 Операция «Включить/отключить СМС-информирование» 29](#_Toc377571574)

[2.7.1 Общие сведения 29](#_Toc377571575)

[2.7.2 Описание входных параметров 29](#_Toc377571576)

[2.7.3 Описание выходных параметров 29](#_Toc377571577)

[2.7.4 Ошибки 29](#_Toc377571578)

[2.7.5 Контрольные примеры 29](#_Toc377571579)

[2.8 Операция «Изменить номер мобильного телефона для СМС-информирования» 31](#_Toc377571580)

[2.8.1 Общие сведения 31](#_Toc377571581)

[2.8.2 Описание входных параметров 31](#_Toc377571582)

[2.8.3 Описание выходных параметров 31](#_Toc377571583)

[2.8.4 Ошибки 31](#_Toc377571584)

[2.8.5 Контрольные примеры 31](#_Toc377571585)

[2.9 Операция «Включить/отключить E-Mail-информирование» 33](#_Toc377571586)

[2.9.1 Общие сведения 33](#_Toc377571587)

[2.9.2 Описание входных параметров 33](#_Toc377571588)

[2.9.3 Описание выходных параметров 33](#_Toc377571589)

[2.9.4 Ошибки 33](#_Toc377571590)

[2.9.5 Контрольные примеры 33](#_Toc377571591)

[2.10 Операция «Изменить адрес e-mail для информирования» 35](#_Toc377571592)

[2.10.1 Общие сведения 35](#_Toc377571593)

[2.10.2 Описание входных параметров 35](#_Toc377571594)

[2.10.3 Описание выходных параметров 35](#_Toc377571595)

[2.10.4 Ошибки 35](#_Toc377571596)

[2.10.5 Контрольные примеры 35](#_Toc377571597)

[2.11 Операция «Применить код активации» 37](#_Toc377571598)

[2.11.1 Общие сведения 37](#_Toc377571599)

[2.11.2 Описание входных параметров 37](#_Toc377571600)

[2.11.3 Описание выходных параметров 37](#_Toc377571601)

[2.11.4 Ошибки 37](#_Toc377571602)

[2.11.5 Контрольные примеры 37](#_Toc377571603)

[2.12 Операция «Сформировать код активации» 39](#_Toc377571604)

[2.12.1 Общие сведения 39](#_Toc377571605)

[2.12.2 Описание входных параметров 39](#_Toc377571606)

[2.12.3 Описание выходных параметров 39](#_Toc377571607)

[2.12.4 Ошибки 39](#_Toc377571608)

[2.12.5 Контрольные примеры 39](#_Toc377571609)

[2.13 Операция «Сформировать код активации и отправить по номеру лицевого счета» 41](#_Toc377571610)

[2.13.1 Общие сведения 41](#_Toc377571611)

[2.13.2 Описание входных параметров 41](#_Toc377571612)

[2.13.3 Описание выходных параметров 41](#_Toc377571613)

[2.13.4 Ошибки 41](#_Toc377571614)

[2.13.5 Контрольные примеры 41](#_Toc377571615)

[2.14 Операция «Сформировать код активации и отправить по e-mail и SMS» 43](#_Toc377571616)

[2.14.1 Общие сведения 43](#_Toc377571617)

[2.14.2 Описание входных параметров 43](#_Toc377571618)

[2.14.3 Описание выходных параметров 43](#_Toc377571619)

[2.14.4 Ошибки 43](#_Toc377571620)

[2.14.5 Контрольные примеры 43](#_Toc377571621)

[2.15 Операция «Изменить размер дневного ограничения на затраты» 45](#_Toc377571622)

[2.15.1 Общие сведения 45](#_Toc377571623)

[2.15.2 Описание входных параметров 45](#_Toc377571624)

[2.15.3 Описание выходных параметров 45](#_Toc377571625)

[2.15.4 Ошибки 45](#_Toc377571626)

[2.15.5 Контрольные примеры 46](#_Toc377571627)

[2.16 Операция «Прикрепить СНИЛС опекуна по номеру ЛС» 47](#_Toc377571628)

[2.16.1 Общие сведения 47](#_Toc377571629)

[2.16.2 Описание входных параметров 47](#_Toc377571630)

[2.16.3 Описание выходных параметров 47](#_Toc377571631)

[2.16.4 Ошибки 47](#_Toc377571632)

[2.17 Операция «Прикрепить СНИЛС опекуна по СНИЛС клиента» 48](#_Toc377571633)

[2.17.1 Общие сведения 48](#_Toc377571634)

[2.17.2 Описание входных параметров 48](#_Toc377571635)

[2.17.3 Описание выходных параметров 48](#_Toc377571636)

[2.17.4 Ошибки 48](#_Toc377571637)

[2.18 Операция «Открепить СНИЛС опекуна по номеру ЛС» 49](#_Toc377571638)

[2.18.1 Общие сведения 49](#_Toc377571639)

[2.18.2 Описание входных параметров 49](#_Toc377571640)

[2.18.3 Описание выходных параметров 49](#_Toc377571641)

[2.18.4 Ошибки 49](#_Toc377571642)

[2.19 Операция «Открепить СНИЛС опекуна по СНИЛС клиента» 50](#_Toc377571643)

[2.19.1 Общие сведения 50](#_Toc377571644)

[2.19.2 Описание входных параметров 50](#_Toc377571645)

[2.19.3 Описание выходных параметров 50](#_Toc377571646)

[2.19.4 Ошибки 50](#_Toc377571647)

[2.20 Операция «Получить список прикрепленных лицевых счетов по СНИЛС опекуна» 51](#_Toc377571648)

[2.20.1 Общие сведения 51](#_Toc377571649)

[2.20.2 Описание входных параметров 51](#_Toc377571650)

[2.20.3 Описание выходных параметров 51](#_Toc377571651)

[2.20.4 Ошибки 51](#_Toc377571652)

[2.20.5 Контрольные примеры 51](#_Toc377571653)

[2.21 Операция «Получить типы уведомлений» 52](#_Toc377571654)

[2.21.1 Общие сведения 52](#_Toc377571655)

[2.21.2 Описание входных параметров 52](#_Toc377571656)

[2.21.3 Описание выходных параметров 52](#_Toc377571657)

[2.21.4 Ошибки 53](#_Toc377571658)

[2.22 Операция «Получить настройки уведомлений клиента по номеру ЛС» 53](#_Toc377571659)

[2.22.1 Общие сведения 53](#_Toc377571660)

[2.22.2 Описание входных параметров 53](#_Toc377571661)

[2.22.3 Описание выходных параметров 53](#_Toc377571662)

[2.22.4 Ошибки 54](#_Toc377571663)

[2.23 Операция «Установить настройки уведомлений клиента по номеру ЛС» 55](#_Toc377571664)

[2.23.1 Общие сведения 55](#_Toc377571665)

[2.23.2 Описание входных параметров 55](#_Toc377571666)

[2.23.3 Описание выходных параметров 55](#_Toc377571667)

[2.23.4 Ошибки 55](#_Toc377571668)

[2.24 Операция «Получить статистику по ЛС» 57](#_Toc377571669)

[2.24.1 Общие сведения 57](#_Toc377571670)

[2.24.2 Описание входных параметров 57](#_Toc377571671)

[2.24.3 Описание выходных параметров 57](#_Toc377571672)

[2.24.4 Ошибки 57](#_Toc377571673)

[2.25 Перевод средств клиента между субсчетами 58](#_Toc377571674)

[2.25.1 Общие сведения 58](#_Toc377571675)

[2.25.2 Описание входных параметров 58](#_Toc377571676)

[2.25.3 Описание выходных параметров 58](#_Toc377571677)

[2.25.4 Ошибки 59](#_Toc377571678)

[2.26 Получение списка учащихся, которые после оплаты платного плана питания ушли в минус 59](#_Toc377571679)

[2.26.1 Общие сведения 59](#_Toc377571680)

[2.26.2 Описание входных параметров 59](#_Toc377571681)

[2.26.3 Описание выходных параметров 59](#_Toc377571682)

[2.26.4 Ошибки 60](#_Toc377571683)

[2.27 Операция «Получение данных о покупках с субчета по абоненскому питанию лицевого счета» 60](#_Toc377571684)

[2.27.1 Общие сведения 60](#_Toc377571685)

[2.27.2 Описание входных параметров 60](#_Toc377571686)

[2.27.3 Описание выходных параметров 60](#_Toc377571687)

[2.27.4 Ошибки 61](#_Toc377571688)

[2.27.5 Контрольные примеры 61](#_Toc377571689)

[2.28 Операция «Получение данных о пополнениях субсчета по абоненскому питанию» 62](#_Toc377571690)

[2.28.1 Общие сведения 62](#_Toc377571691)

[2.28.2 Описание входных параметров 62](#_Toc377571692)

[2.28.3 Описание выходных параметров 62](#_Toc377571693)

[2.28.4 Ошибки 62](#_Toc377571694)

[2.28.5 Контрольные примеры 63](#_Toc377571695)

[2.29 Перевод средств клиента между субсчетами 64](#_Toc377571696)

[2.29.1 Общие сведения 64](#_Toc377571697)

[2.29.2 Описание входных параметров 64](#_Toc377571698)

[2.29.3 Описание выходных параметров 64](#_Toc377571699)

[2.29.4 Ошибки 65](#_Toc377571700)

[2.29.5 Контрольные примеры 65](#_Toc377571701)

[2.30 Подключение подписки на АП клиенту 65](#_Toc377571702)

[2.30.1 Общие сведения 65](#_Toc377571703)

[2.30.2 Описание входных параметров 66](#_Toc377571704)

[2.30.3 Описание выходных параметров 66](#_Toc377571705)

[2.30.4 Ошибки 66](#_Toc377571706)

[2.30.5 Контрольные примеры 67](#_Toc377571707)

[2.31 Получение данных об активной подписке АП по номеру ЛС 67](#_Toc377571708)

[2.31.1 Общие сведения 67](#_Toc377571709)

[2.31.2 Описание входных параметров 67](#_Toc377571710)

[2.31.3 Описание выходных параметров 68](#_Toc377571711)

[2.31.4 Ошибки 68](#_Toc377571712)

[2.31.5 Контрольные примеры 68](#_Toc377571713)

[2.32 Приостановка действия подписки на АП по номеру ЛС 69](#_Toc377571714)

[2.32.1 Общие сведения 69](#_Toc377571715)

[2.32.2 Описание входных параметров 69](#_Toc377571716)

[2.32.3 Описание выходных параметров 69](#_Toc377571717)

[2.32.4 Ошибки 70](#_Toc377571718)

[2.32.5 Контрольные примеры 70](#_Toc377571719)

[2.33 Возобновление подписки на АП по номеру ЛС 70](#_Toc377571720)

[2.33.1 Общие сведения 70](#_Toc377571721)

[2.33.2 Описание входных параметров 71](#_Toc377571722)

[2.33.3 Описание выходных параметров 71](#_Toc377571723)

[2.33.4 Ошибки 71](#_Toc377571724)

[2.33.5 Контрольные примеры 71](#_Toc377571725)

[2.34 Редактирование плана питания по АП (создание новой циклограммы) по номеру ЛС 72](#_Toc377571726)

[2.34.1 Общие сведения 72](#_Toc377571727)

[2.34.2 Описание входных параметров 72](#_Toc377571728)

[2.34.3 Описание выходных параметров 72](#_Toc377571729)

[2.34.4 Ошибки 74](#_Toc377571730)

[2.34.5 Контрольные примеры 74](#_Toc377571731)

[2.35 Получение данных о текущем плане питания по АП по номеру ЛС 75](#_Toc377571732)

[2.35.1 Общие сведения 75](#_Toc377571733)

[2.35.2 Описание входных параметров 75](#_Toc377571734)

[2.35.3 Описание выходных параметров 75](#_Toc377571735)

[2.35.4 Ошибки 77](#_Toc377571736)

[2.35.5 Контрольные примеры 77](#_Toc377571737)

[2.36 Получить список комплексов, которые участвуют в АП, по номеру ЛС 78](#_Toc377571738)

[2.36.1 Общие сведения 78](#_Toc377571739)

[2.36.2 Описание входных параметров 78](#_Toc377571740)

[2.36.3 Описание выходных параметров 78](#_Toc377571741)

[2.36.4 Ошибки 78](#_Toc377571742)

[2.36.5 Контрольные примеры 79](#_Toc377571743)

[3 Приложения 81](#_Toc377571744)

[3.1 Описание сервиса (WSDL) 81](#_Toc377571745)

[3.2 Описание общих структур данных 128](#_Toc377571746)

[3.2.1 Параметр комплексного типа: ClientSummaryExt 128](#_Toc377571747)

[3.2.2 Параметр комплексного типа: ClientItem 131](#_Toc377571748)

[3.2.3 Параметр комплексного типа: PurchaseListExt 131](#_Toc377571749)

[3.2.4 Параметр комплексного типа: PurchaseExt 132](#_Toc377571750)

[3.2.5 Параметр комплексного типа: PaymentList 132](#_Toc377571751)

[3.2.6 Параметр комплексного типа: Payment 132](#_Toc377571752)

[3.2.7 Параметр комплексного типа: MenuListExt 133](#_Toc377571753)

[3.2.8 Параметр комплексного типа: MenuDateItemExt 133](#_Toc377571754)

[3.2.9 Параметр комплексного типа: MenuItemExt 133](#_Toc377571755)

[3.2.10 Параметр комплексного типа: EnterEventList 134](#_Toc377571756)

[3.2.11 Параметр комплексного типа: EnterEventItem 134](#_Toc377571757)

[3.2.12 Параметр комплексного типа: ClientNotificationSettingsItem 135](#_Toc377571758)

[3.2.13 Параметр комплексного типа: StudentsConfirmPaymentList 135](#_Toc377571759)

[3.2.14 Параметр комплексного типа: StudentMustPayItem 135](#_Toc377571760)

[3.2.15 Параметр комплексного типа: CycleDiagramIn 135](#_Toc377571761)

[3.2.16 Параметр комплексного типа: ComplexInfoList 136](#_Toc377571762)

[3.2.17 Параметр комплексного типа: ComplexInfoExt 136](#_Toc377571763)

изменения

|  |  |  |  |
| --- | --- | --- | --- |
| Версия | Дата | Автор | Изменения |
| 1.0 | 30.07.2012 | Скворцов А.В. | Первичная версия документа |
| 1.1 | 13.08.2012 | Скворцов А.В. | Добавлены операции для работы с кодами активации |
| 1.2 | 26.03.2013 | Кадыров Д.И. | Добавлена операция изменения размер дневного ограничения на затраты по номеру контракта ребенка |
| 1.3 | 03.06.2013 | Скворцов А.В. | Добавлены операции управления привязкой СНИЛС опекуна, изменения параметров уведомлений клиентов |
| 1.4 | 18.10.2013 | Кадыров Д.И. | Добавлены поля по Абоненскому питанию и добавлена опрерация перевода перевода средств с субсчетов |
| 1.4.2 | 15.01.2014 | Кадыров Д.И. | Добавлены методы по получению информации по питанию, поплнению и перевода стредств по АП |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

термины/СОКРАЩЕНИя

|  |  |
| --- | --- |
| Термин/сокращение | Описание |
| Поставщик | Организация-владелец электронного сервиса |
| WSDL | Web Services Description Language — язык описания [веб-сервисов](http://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B8%D1%81). |
| XSD | XML Schema definition - язык описания структуры XML документа. |
| SOAP | Simple Object Access Protocol - протокол обмена структурированными сообщениями в распределённой вычислительной среде. |
| XML | eXtensible Markup Language — расширяемый язык разметки. |

1. Общие сведения

Руководящие документы

Основанием для создания и использования электронного сервиса являются перечисленные ниже документы.

Нормативно-правовые документы:

* Постановление Правительства Российской Федерации от 8 сентября 2010 г. № 697 «О единой системе межведомственного электронного взаимодействия»;
* Приказ Министерства связи и массовых коммуникаций Российской Федерации от 27 декабря 2010 г. № 190 «Об утверждении технических требований к взаимодействию информационных систем в единой системе межведомственного электронного взаимодействия».

Описание электронного сервиса

|  |  |
| --- | --- |
| **Наименование:** | Сервис информирования |
| **Назначение:** | Предоставление доступа к данным о посещении образовательных учреждений, меню поставщиков питания, оказанных услугах питания, информации о статусе лицевого счета держателя |
| **Поставщик:** | Региональный оператор проекта безналичной оплаты питания |
| **Область применения:** | Предоставление открытого доступа к данным |

Операции (методы) электронного сервиса

|  |  |
| --- | --- |
| **Метод** | **Назначение** |
| getSummary  Получение данных о лицевом счете | Предоставление информации о ЛС: ФИО, баланс, статус, параметры лицевого счета, информация о нахождении в ОУ |
| getSymmaryByTypeId  Получение данных о лицевом счете через идентификатор | Получение данных о лицевом счете через СНИЛС опекуна, GUID или внешний идентификатор |
| getPurchaseList  Получение данных о покупках с лицевого счета | Предоставление информации о покупках: дата, сумма, состав |
| getPaymentList  Получение данных о пополнениях лицевого счета | Предоставление информации о пополнениях: дата, сумма, источник |
| getMenuList  Получение информации о меню столовой | Получение информации о меню столовой: дата, наименование, цена |
| getEnterEventList  Получение данных посещениях образовательного учреждения | Получение информации о посещении образовательного учреждения: дата, направление входа |
| enableNotificationBySMS  Включить/отключить СМС-информирование | Изменение настроек СМС-информирования |
| changeMobilePhone  Изменить номер мобильного телефона для СМС-информирования | Изменение настроек СМС-информирования (номер телефона) |
| enableNotificationByEmail  Включить/отключить E-Mail-информирование | Изменение настроек Email-информирования лицевого счета |
| changeEmail  Изменить адрес e-mail для информирования | Изменение настроек Email-информирования (адрес) |
| activateLinkingToken  Применить код активации | Применить код активации для связки с лицевым счетом |
| generateLinkingToken  Сформировать код активации | Сформировать новый код активации для лицевого счета |
| sendLinkingTokenByContractId  Сформировать код активации и отправить по номеру лицевого счета | Сформировать код активации и отправить по e-mail и SMS для указанного лицевого счета |
| sendLinkingTokenByMobile  Сформировать код активации и отправить по номеру телефона | Сформировать код активации и отправить по e-mail и SMS для указанного номера телефона |
| changeExpenditureLimit()  Изменить размер дневного ограничения на затраты по номеру контракта ребенка | Изменить размер дневного ограничения на затраты по номеру контракта ребенка |
| attachGuardSan  Прикрепить СНИЛС опекуна по номеру ЛС | Прикрепить (присвоить) СНИЛС опекуна по номеру лицевого счета |
| attachGuardSanBySan  Прикрепить СНИЛС опекуна по СНИЛС клиента | Прикрепить СНИЛС опекуна по СНИЛС клиента |
| detachGuardSan  Открепить СНИЛС опекуна по номеру ЛС | Открепить СНИЛС опекуна по номеру лицевого счета |
| detachGuardSanBySan  Открепить СНИЛС опекуна по СНИЛС клиента | Открепить СНИЛС опекуна по СНИЛС клиента |
| transferBalance  Перевод средств клиента между субсчетами | Перевод средств клиента между субсчетами |
| getStudentsByCanNotConfirmPayment  Получение списка учащихся, которые после оплаты платного плана питания ушли в минус | Получение списка учащихся, которые после оплаты платного плана питания ушли в минус |
| getPurchaseSubscriptionFeedingListBySan  Получение данных о покупках с лицевого счета | Предоставление информации о покупках по абоненскому питанию: дата, сумма, состав |
| getPaymentSubscriptionFeedingListBySan  Получение данных о пополнениях лицевого счета | Предоставление информации о пополнениях по субсчету абоненского питания: дата, сумма, источник |
| transferBalanceBySan  Перевод средств клиента между субсчетами | Перевод средств клиента между субсчетами |
| createSubscriptionFeeding | Подключить подписку на АП по номеру ЛС |
| findSubscriptionFeeding | Получить данные об активной подписке АП по номеру ЛС |
| suspendSubscriptionFeeding | Приостановить действие подписки на АП по номеру ЛС |
| reopenSubscriptionFeeding | Возобновить подписку на АП по номеру ЛС |
| editSubscriptionFeedingPlan | Редактировать план питания по АП (т.е. создать новую циклограмму) по номеру ЛС |
| findClientCycleDiagram | Получить данные о текущем плане питания по АП (т.е. данные о текущей циклограмме) по номеру ЛС |
| findComplexesWithSubFeeding | Получить список комплексов, которые участвуют в АП, по номеру ЛС |
| createSubscriptionFeedingBySan | Подключить подписку на АП по СНИЛС клиента |
| findSubscriptionFeedingBySan | Получить данные об активной подписке АП по СНИЛС клиента |
| suspendSubscriptionFeedingBySan | Приостановить действие подписки на АП по СНИЛС клиента |
| reopenSubscriptionFeedingBySan | Возобновить подписку на АП по СНИЛС клиента |
| editSubscriptionFeedingPlanBySan | Редактировать план питания по АП (т.е. создать новую циклограмму) по СНИЛС клиента |
| findClientCycleDiagramBySan | Получить данные о текущем плане питания по АП (т.е. данные о текущей циклограмме) по СНИЛС клиента |
| findComplexesWithSubFeedingBySan | Получить список комплексов, которые участвуют в АП, по СНИЛС клиента |

Сценарии использования и схема взаимодействия

1. Информационная система вызывает сервис в синхронном режиме используя getSummaryByTypedId, передавая идентификатор клиента, которым он идентифицируется в ее базе данных, в ответ возвращаются данные о лицевом счете и номер лицевого счета
2. Информационная система получает данные, используя методы getPurchaseList, getPaymentList, getEnterEventList, getMenuList

Связи с другими электронными сервисами

Данный сервис не имеет связей с другими сервисами.

1. Руководство пользователя

## Операция «Получение данных о лицевом счете»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | getSummary |
| **Наименование операции:** | Получение данных о лицевом счете |
| **Назначение операции:** | Предоставление информации о ЛС: ФИО, баланс, статус, параметры лицевого счета, информация о нахождении в ОУ |

### Описание входных параметров

**Входные данные: getSummary**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |

### Описание выходных параметров

**Выходные данные: clientSummaryResult**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | clientSummary | Данные лицевого счета | + | ClientSummaryExt |  |
| 2 | resultCode | Код возврата | + | xs:long |  |
| 3 | description | Описание результата обработки | + | xs:string |  |

Параметры комплексного типа описаны в приложении «Описание общих структур данных».

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Лицевой счет не найден | Лицевой счет не найден |  |

### Контрольные примеры

**Запрос**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:getSummary>*  *<contractId>200485</contractId>*  *</soap:getSummary>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:getSummaryResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<clientSummary ContractId="200485" DateOfContract="2011-09-09" StateOfContract="Активен" Balance="566389" SubBalance1="166389" OverdraftLimit="100000" ExpenditureLimit="20000" FirstName="Петр" LastName="Петров" MiddleName="Иванович" OfficialName="Демонстрационная школа" LastEnterEventCode="5" LastEnterEventTime="2012-04-16T13:35:47" NotifyViaSMS="true" NotifyViaEmail="false" MobilePhone="79161111111" Email="email" DefaultMerchantId="52" DefaultMerchantInfo="{TSP.SBID=123123}"/>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *</return>*  *</ns2:getSummaryResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Операция «Получение данных о лицевом счете через идентификатор»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | getSymmaryByTypeId |
| **Наименование операции:** | Получение данных о лицевом счете через идентификатор |
| **Назначение операции:** | Предоставление информации о ЛС: ФИО, баланс, статус, параметры лицевого счета, информация о нахождении в ОУ |

### Описание входных параметров

**Входные данные: getSummaryByTypedId**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | id | Идентификатор | + | xs:string |  |
| 2 | idType | Тип идентификатора | + | xs:int | 0 – номер лицевого счета, 1 – СНИЛС представителя, 2 – внешний идентификатор, 3 – GUID |

### Описание выходных параметров

**Выходные данные: clientSummaryResult**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | clientSummary | Данные лицевого счета | + | ClientSummaryExt |  |
| 2 | resultCode | Код возврата | + | xs:long |  |
| 3 | description | Описание результата обработки | + | xs:string |  |

Параметры комплексного типа описаны в приложении «Описание общих структур данных».

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Лицевой счет не найден | Лицевой счет не найден |  |
| 4 | 120 | Найдено больше одного лицевого счета | Для данного идентификатора найдено больше одного лицевого счета |  |

### Контрольные примеры

**Запрос**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:getSummaryByTypedId>*  *<id>df7d827f-f425-4205-961e-6eb79231c58e</id>*  *<idType>3</idType>*  *</soap:getSummaryByTypedId>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:getSummaryByTypedIdResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<clientSummary ContractId="200485" DateOfContract="2011-09-09" StateOfContract="Активен" Balance="566389" SubBalance1="166389" OverdraftLimit="100000" ExpenditureLimit="20000" FirstName="Петр" LastName="Петров" MiddleName="Иванович" OfficialName="Демонстрационная школа" LastEnterEventCode="5" LastEnterEventTime="2012-04-16T13:35:47" NotifyViaSMS="true" NotifyViaEmail="false" MobilePhone="79161111111" Email="email" DefaultMerchantId="52" DefaultMerchantInfo="{TSP.SBID=123123}"/>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *</return>*  *</ns2:getSummaryByTypedIdResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Операция «Получение данных о покупках с лицевого счета»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | getPurchaseList |
| **Наименование операции:** | Получение данных о покупках с лицевого счета |
| **Назначение операции:** | Предоставление информации о покупках: дата, сумма, состав |

### Описание входных параметров

**Входные данные: getPurchaseList**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |
| 2 | startDate | Начальная дата выборки | + | xs:dateTime |  |
| 3 | endDate | Конечная дата выборки | + | xs:dateTime |  |

### Описание выходных параметров

**Выходные данные: purchaseListResult**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | purchaseList | Список покупок | + | PurchaseListExt |  |
| 2 | resultCode | Код возврата | + | xs:long |  |
| 3 | description | Описание результата обработки | + | xs:string |  |

Параметры комплексного типа описаны в приложении «Описание общих структур данных».

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Лицевой счет не найден | Лицевой счет не найден |  |

### Контрольные примеры

**Запрос**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:getPurchaseList>*  *<contractId>200485</contractId>*  *<startDate>2012-01-17T00:00:00Z</startDate>*  *<endDate>2012-02-17T00:00:00Z</endDate>*  *</soap:getPurchaseList>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:getPurchaseListResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<purchaseList>*  *<P Time="2012-02-14T21:37:36" Sum="0" SocDiscount="1500" TrdDiscount="0" Donation="0" ByCash="0" ByCard="0" IdOfCard="0">*  *<E Name="Хлеб Сельский" Sum="0" Amount="1" Type="2"/>*  *<E Name="Кофейный напиток с молоком" Sum="0" Amount="1" Type="2"/>*  *<E Name="Каша молочная пшенная с маслом" Sum="0" Amount="1" Type="2"/>*  *<E Name="Сыр Голландский" Sum="0" Amount="1" Type="2"/>*  *<E Name="ГПД завтрак" Sum="0" Amount="1" Type="1"/>*  *</P>*  *<P Time="2012-02-14T21:37:55" Sum="1425" SocDiscount="0" TrdDiscount="75" Donation="0" ByCash="0" ByCard="1425" IdOfCard="0">*  *<E Name="Хлеб Сельский" Sum="0" Amount="1" Type="2"/>*  *<E Name="Кофейный напиток с молоком" Sum="0" Amount="1" Type="2"/>*  *<E Name="Каша молочная пшенная с маслом" Sum="0" Amount="1" Type="2"/>*  *<E Name="Сыр Голландский" Sum="0" Amount="1" Type="2"/>*  *<E Name="ГПД завтрак" Sum="1425" Amount="1" Type="1"/>*  *</P>*  *</purchaseList>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *</return>*  *</ns2:getPurchaseListResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Операция «Получение данных о пополнениях лицевого счета»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | getPaymentList |
| **Наименование операции:** | Получение данных о пополнениях лицевого счета |
| **Назначение операции:** | Предоставление информации о пополнениях: дата, сумма, источник |

### Описание входных параметров

**Входные данные: getPaymentList**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |
| 2 | startDate | Начальная дата выборки | + | xs:dateTime |  |
| 3 | endDate | Конечная дата выборки | + | xs:dateTime |  |

### Описание выходных параметров

**Выходные данные: paymentListResult**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | paymentList | Список пополнений | + | PaymentList |  |
| 2 | resultCode | Код возврата | + | xs:long |  |
| 3 | description | Описание результата обработки | + | xs:string |  |

Параметры комплексного типа описаны в приложении «Описание общих структур данных».

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Лицевой счет не найден | Лицевой счет не найден |  |

### Контрольные примеры

**Запрос**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:getPaymentList>*  *<contractId>200485</contractId>*  *<startDate>2012-01-17T00:00:00Z</startDate>*  *<endDate>2012-02-17T00:00:00Z</endDate>*  *</soap:getPaymentList>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:getPaymentListResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<paymentList>*  *<P Time="2012-02-15T12:26:56" Sum="999" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-15T12:26:58" Sum="-999" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-15T12:48:27" Sum="999" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-15T12:48:27" Sum="-999" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-15T12:58:35" Sum="999" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-15T12:58:37" Sum="-999" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-15T13:55:59" Sum="2000" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-16T20:11:56" Sum="10000" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-16T20:13:25" Sum="10000" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-16T20:38:35" Sum="5000" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-16T20:39:18" Sum="100000" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-16T21:13:28" Sum="100000" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-16T21:19:53" Sum="100000" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-17T11:40:37" Sum="999" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-17T11:40:38" Sum="-999" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-17T11:41:46" Sum="999" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-17T11:41:47" Sum="-999" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-17T11:50:54" Sum="999" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-17T11:50:55" Sum="-999" Origin="Конкорд-ПА - Банкоматы"/>*  *</paymentList>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *</return>*  *</ns2:getPaymentListResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Операция «Получение информации о меню столовой»

Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | getMenuList |
| **Наименование операции:** | Получение информации о меню столовой |
| **Назначение операции:** | Получение информации о меню столовой: дата, наименование, цена |

### Описание входных параметров

**Входные данные: getMenuList**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |
| 2 | startDate | Начальная дата выборки | + | xs:dateTime |  |
| 3 | endDate | Конечная дата выборки | + | xs:dateTime |  |

### Описание выходных параметров

**Выходные данные: menuListResult**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | menuList | Меню | + | MenuListExt |  |
| 2 | resultCode | Код возврата | + | xs:long |  |
| 3 | description | Описание результата обработки | + | xs:string |  |

Параметры комплексного типа описаны в приложении «Описание общих структур данных».

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Лицевой счет не найден | Лицевой счет не найден |  |

### Контрольные примеры

**Запрос**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:getMenuList>*  *<contractId>7607195</contractId>*  *<startDate>2012-05-17T00:00:00Z</startDate>*  *<endDate>2012-05-18T00:00:00Z</endDate>*  *</soap:getMenuList>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:getMenuListResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<menuList>*  *<M Date="2012-05-17">*  *<E Group="Бутерброды" Name="Бутерброд горячий (колбаса, сыр)" Price="2300"/>*  *<E Group="Бутерброды" Name="Бутерброд горячий с сос." Price="2000"/>*  *<E Group="Бутерброды" Name="Бутерброд с колб." Price="1400"/>*  *<E Group="Бутерброды" Name="Бутерброд с колбас. с/к" Price="2000"/>*  *<E Group="Бутерброды" Name="Бутерброд с семгой сл.сол" Price="3200"/>*  *<E Group="Бутерброды" Name="Бутерброд с сыром" Price="1200"/>*  *<E Group="Бутерброды" Name="Сандвич на ржан.хлебе" Price="1100"/>*  *<E Group="Выпечка" Name="Булочка" Price="1000"/>*  *<E Group="Выпечка" Name="Ватрушка с абр. джемом" Price="1000"/>*  *<E Group="Выпечка" Name="Ватрушка с творогом" Price="1000"/>*  *<E Group="Выпечка" Name="Кекс весенний 500гр" Price="9000"/>*  *<E Group="Выпечка" Name="Кекс весенний 50гр" Price="1200"/>*  *<E Group="Выпечка" Name="Кекс столичный" Price="2000"/>*  *<E Group="Выпечка" Name="Коржик молочный" Price="1200"/>*  *<E Group="Выпечка" Name="Песочник с изюмом" Price="1400"/>*  *<E Group="Выпечка" Name="Печенье ракушка" Price="2000"/>*  *<E Group="Выпечка" Name="Пир. возд. с кремом" Price="1400"/>*  *<E Group="Выпечка" Name="Пир. корз. с белк.крем." Price="1400"/>*  *<E Group="Выпечка" Name="Пир. песоч.глазир." Price="1200"/>*  *<E Group="Выпечка" Name="Пир. труб. йогурт крем." Price="1200"/>*  *<E Group="Выпечка" Name="Пир.воз-орех" Price="1600"/>*  *<E Group="Выпечка" Name="Пир.песоч.ореховое" Price="1600"/>*  *<E Group="Выпечка" Name="Пир.слоен.с мясом" Price="3500"/>*  *<E Group="Выпечка" Name="Пир.труб. завар. с б/к" Price="1400"/>*  *<E Group="Выпечка" Name="Пир.труб. слоен. с б/к" Price="1400"/>*  *<E Group="Выпечка" Name="Пир.труб.зав.со слив.крем" Price="1400"/>*  *<E Group="Выпечка" Name="Пирог Домашний" Price="2000"/>*  *<E Group="Выпечка" Name="Пирожное картошка" Price="1600"/>*  *<E Group="Выпечка" Name="Пирожное ореховое" Price="1600"/>*  *<E Group="Выпечка" Name="Пирожное орешек" Price="1600"/>*  *<E Group="Выпечка" Name="Пирожок с вишн. джемом" Price="1000"/>*  *<E Group="Выпечка" Name="Пирожок с грибами и карт." Price="2000"/>*  *<E Group="Выпечка" Name="Пирожок с капус. и яйцом" Price="1000"/>*  *<E Group="Выпечка" Name="Пирожок с карт. и луком" Price="1000"/>*  *<E Group="Выпечка" Name="Пирожок с клубн. джемом" Price="1000"/>*  *<E Group="Выпечка" Name="Пирожок с курагой" Price="1000"/>*  *<E Group="Выпечка" Name="Пирожок с мясом" Price="2000"/>*  *<E Group="Выпечка" Name="Пирожок с печенью и рисом" Price="1000"/>*  *<E Group="Выпечка" Name="Пирожок с рисом и яйц." Price="1000"/>*  *<E Group="Выпечка" Name="Пирожок с рыбой и карт." Price="1000"/>*  *<E Group="Выпечка" Name="Пирожок с ябл. джемом" Price="1000"/>*  *<E Group="Выпечка" Name="Пол. пес. с повидлом" Price="1200"/>*  *<E Group="Выпечка" Name="Рожок слоеный с повидлом" Price="1600"/>*  *<E Group="Выпечка" Name="Ром-баба с шок. помад." Price="1200"/>*  *<E Group="Выпечка" Name="Рулет фруктовый" Price="1500"/>*  *<E Group="Выпечка" Name="Сосиска в тесте" Price="2300"/>*  *<E Group="Выпечка" Name="Шарлотка с ябл." Price="1000"/>*  *<E Group="Выпечка" Name="Язык слоеный" Price="1400"/>*  *<E Group="Гарниры" Name="Гречка отварная" Price="1200"/>*  *<E Group="Гарниры" Name="Капуста тушен." Price="1100"/>*  *<E Group="Гарниры" Name="Макаронные изд." Price="700"/>*  *<E Group="Гарниры" Name="Пюре картофель" Price="1600"/>*  *<E Group="Гарниры" Name="Рагу из овощей" Price="1200"/>*  *<E Group="Гарниры" Name="Рис отварной" Price="800"/>*  *<E Group="Гор. блюда" Name="Бефстроганов" Price="4900"/>*  *<E Group="Гор. блюда" Name="Гуляш" Price="4900"/>*  *<E Group="Гор. блюда" Name="Запек. карт с мясом" Price="5000"/>*  *<E Group="Гор. блюда" Name="Запек. карт.рыбн." Price="4100"/>*  *<E Group="Гор. блюда" Name="Зразы карт.с морк.и лук" Price="1300"/>*  *<E Group="Гор. блюда" Name="Зразы курин.рис/лук" Price="2700"/>*  *<E Group="Гор. блюда" Name="Зразы мясные" Price="3500"/>*  *<E Group="Гор. блюда" Name="Зразы рыбн.с карт." Price="2800"/>*  *<E Group="Гор. блюда" Name="Картофель жареный" Price="2000"/>*  *<E Group="Гор. блюда" Name="Котл. карт с мясом" Price="7700"/>*  *<E Group="Гор. блюда" Name="Котлета из птицы" Price="1500"/>*  *<E Group="Гор. блюда" Name="Котлета капустная" Price="1000"/>*  *<E Group="Гор. блюда" Name="Котлета картофельная" Price="1000"/>*  *<E Group="Гор. блюда" Name="Котлета морковная" Price="1000"/>*  *<E Group="Гор. блюда" Name="Котлета мясная руб." Price="2000"/>*  *<E Group="Гор. блюда" Name="Котлета свекольная" Price="1000"/>*  *<E Group="Гор. блюда" Name="Котлеты кур. с морк." Price="3300"/>*  *<E Group="Гор. блюда" Name="Котлеты куриные" Price="3200"/>*  *<E Group="Гор. блюда" Name="Котлеты рубл.гов." Price="2800"/>*  *<E Group="Гор. блюда" Name="Котлеты рубл.кур." Price="3300"/>*  *<E Group="Гор. блюда" Name="Котлеты рыбные" Price="2600"/>*  *<E Group="Гор. блюда" Name="Курица отварная" Price="2500"/>*  *<E Group="Гор. блюда" Name="Медальон из свинины" Price="7200"/>*  *<E Group="Гор. блюда" Name="Медальон из филе птицы" Price="4800"/>*  *<E Group="Гор. блюда" Name="Мясо отварное гов." Price="2600"/>*  *<E Group="Гор. блюда" Name="Печень тушон. гов." Price="2900"/>*  *<E Group="Гор. блюда" Name="Плов курин." Price="3900"/>*  *<E Group="Гор. блюда" Name="Рагу куринное" Price="2700"/>*  *<E Group="Гор. блюда" Name="Рулет карт.с мясом" Price="4000"/>*  *<E Group="Гор. блюда" Name="Рулет мясной с яйцом" Price="4000"/>*  *<E Group="Гор. блюда" Name="Рыба в кляре" Price="3000"/>*  *<E Group="Гор. блюда" Name="Рыба тушен. к карт." Price="4300"/>*  *<E Group="Гор. блюда" Name="Свин. туш. с картоф." Price="5000"/>*  *<E Group="Гор. блюда" Name="Суфле из печени" Price="7700"/>*  *<E Group="Гор. блюда" Name="Тефтели гов." Price="3100"/>*  *<E Group="Гор. блюда" Name="Тефтели рыбные" Price="2400"/>*  *<E Group="Гор. блюда" Name="Тефтели с рис." Price="2700"/>*  *<E Group="Гор. блюда" Name="Филе птицы в кляре" Price="3800"/>*  *<E Group="Гор. блюда" Name="Эскалоп из свинины" Price="7800"/>*  *<E Group="Завтраки" Name="Вареники лен. с твор." Price="2700"/>*  *<E Group="Завтраки" Name="Греча отварная" Price="800"/>*  *<E Group="Завтраки" Name="Запеканка из творога" Price="2700"/>*  *<E Group="Завтраки" Name="Запеканка макар." Price="1500"/>*  *<E Group="Завтраки" Name="Запеканка твор. с изюмом" Price="2600"/>*  *<E Group="Завтраки" Name="Котлеты рубл.гов." Price="1400"/>*  *<E Group="Завтраки" Name="Крупеник рис/творог" Price="2600"/>*  *<E Group="Завтраки" Name="Курица отварная" Price="1600"/>*  *<E Group="Завтраки" Name="Омлет" Price="8800"/>*  *<E Group="Завтраки" Name="Плов курин." Price="2900"/>*  *<E Group="Завтраки" Name="Пюре картофель." Price="1100"/>*  *<E Group="Завтраки" Name="Рагу из овощей" Price="800"/>*  *<E Group="Завтраки" Name="Рагу куринное" Price="2000"/>*  *<E Group="Завтраки" Name="Рис отварной" Price="500"/>*  *<E Group="Завтраки" Name="Сырники из твор." Price="2700"/>*  *<E Group="Завтраки" Name="Тефтели гов." Price="2100"/>*  *<E Group="Завтраки" Name="Тефтели рыбные" Price="1600"/>*  *<E Group="Завтраки" Name="Тефтели с рис." Price="1400"/>*  *<E Group="Завтраки" Name="Хлеб" Price="200"/>*  *<E Group="Каши" Name="Каша гречневая" Price="1100"/>*  *<E Group="Каши" Name="Каша манная" Price="600"/>*  *<E Group="Каши" Name="Каша овсяная" Price="1000"/>*  *<E Group="Каши" Name="Каша пшен. с изум." Price="1300"/>*  *<E Group="Каши" Name="Каша пшенная" Price="1300"/>*  *<E Group="Каши" Name="Каша рис с изум." Price="1300"/>*  *<E Group="Кондитер" Name="Батончик &quot;Кит-кат&quot;" Price="2700"/>*  *<E Group="Кондитер" Name="Батончик &quot;Милки Вей&quot;" Price="2000"/>*  *<E Group="Кондитер" Name="Батончик &quot;Несквик&quot;" Price="1700"/>*  *<E Group="Кондитер" Name="Батончик Гематог." Price="700"/>*  *<E Group="Кондитер" Name="Батончик кокосовый &quot;Гемо-хелпер&quot;" Price="1400"/>*  *<E Group="Кондитер" Name="Батончик Натс" Price="2800"/>*  *<E Group="Кондитер" Name="Батончик Пульсар" Price="1800"/>*  *<E Group="Кондитер" Name="Батончик СИЛЬВЕР" Price="2000"/>*  *<E Group="Кондитер" Name="Батончик Темпо" Price="1200"/>*  *<E Group="Кондитер" Name="Вафли &quot;Лимонные&quot;" Price="900"/>*  *<E Group="Кондитер" Name="Вафли без начинки" Price="1900"/>*  *<E Group="Кондитер" Name="Вафли глазированные" Price="2800"/>*  *<E Group="Кондитер" Name="Вафли Коломенск." Price="500"/>*  *<E Group="Кондитер" Name="Джем (ассортимент)" Price="500"/>*  *<E Group="Кондитер" Name="Зефир &quot;Ванильный&quot;" Price="600"/>*  *<E Group="Кондитер" Name="Конфета Курьёз" Price="900"/>*  *<E Group="Кондитер" Name="Конфета Сласть" Price="1200"/>*  *<E Group="Кондитер" Name="Конфета Фарс" Price="1500"/>*  *<E Group="Кондитер" Name="Конфета Экер" Price="800"/>*  *<E Group="Кондитер" Name="Крекер Соломка" Price="1000"/>*  *<E Group="Кондитер" Name="Мармелад Джу-Джав" Price="1000"/>*  *<E Group="Кондитер" Name="Мармелад Зол.Мишка" Price="800"/>*  *<E Group="Кондитер" Name="Мармелад Медвеж." Price="2300"/>*  *<E Group="Кондитер" Name="Мармелад Фрутелла" Price="2500"/>*  *<E Group="Кондитер" Name="Мёд" Price="700"/>*  *<E Group="Кондитер" Name="Мини-кекс" Price="900"/>*  *<E Group="Кондитер" Name="Ореховый батончик &quot;Пульсар&quot;" Price="1800"/>*  *<E Group="Кондитер" Name="Печенье &quot;Мишка-топтышка&quot;" Price="1000"/>*  *<E Group="Кондитер" Name="Печенье Вагон Виллз" Price="1600"/>*  *<E Group="Кондитер" Name="Печенье Нежное" Price="500"/>*  *<E Group="Кондитер" Name="Пироженное биск.шок.начинка &quot;Барни&quot;" Price="2400"/>*  *<E Group="Кондитер" Name="Пироженое Бонди" Price="1200"/>*  *<E Group="Кондитер" Name="Пироженое Чокопай" Price="900"/>*  *<E Group="Кондитер" Name="Снэки Кук. палоч. Кузя" Price="1500"/>*  *<E Group="Кондитер" Name="Успевай-ка с кальцием и жел." Price="1300"/>*  *<E Group="Кондитер" Name="Хрустящие тр. Читос" Price="2400"/>*  *<E Group="Кондитер" Name="Шок. Киндер CiO" Price="3600"/>*  *<E Group="Кондитер" Name="Шок. Киндер МАХI" Price="1600"/>*  *<E Group="Кондитер" Name="Шоколад &quot;Алёнка&quot;" Price="1200"/>*  *<E Group="Молочная прод." Name="Биойогурт питьевой" Price="3100"/>*  *<E Group="Молочная прод." Name="Брикет &quot;Филёвский&quot;" Price="1900"/>*  *<E Group="Молочная прод." Name="Ваф.стак. &quot;Фил.&quot;" Price="1900"/>*  *<E Group="Молочная прод." Name="Йогурт" Price="1200"/>*  *<E Group="Молочная прод." Name="Кокт. Мол. кар. банан" Price="2000"/>*  *<E Group="Молочная прод." Name="Кокт. Мол. клубника" Price="2000"/>*  *<E Group="Молочная прод." Name="Кокт. Мол. шоколад" Price="2200"/>*  *<E Group="Молочная прод." Name="Кокт. Мол.Ваниль" Price="2000"/>*  *<E Group="Молочная прод." Name="Коктейль молоч. &quot;Здрайверы&quot; в асс." Price="2400"/>*  *<E Group="Молочная прод." Name="Молоко &quot;Большая перемена&quot; 3,2%" Price="1900"/>*  *<E Group="Молочная прод." Name="Молоко &quot;Здрайверы&quot; 3,2%" Price="1700"/>*  *<E Group="Молочная прод." Name="Мор. &quot;Дед Мороз&quot;Эскимо" Price="1300"/>*  *<E Group="Молочная прод." Name="Морож. &quot; Бородино&quot;" Price="2400"/>*  *<E Group="Молочная прод." Name="Морож. &quot;Вол.фонарь&quot;" Price="2600"/>*  *<E Group="Молочная прод." Name="Морож. &quot;Елочка&quot;" Price="2300"/>*  *<E Group="Молочная прод." Name="Морож. &quot;Ленин&quot; батон" Price="2300"/>*  *<E Group="Молочная прод." Name="Морож. &quot;Ленин.&quot; эскимо" Price="1900"/>*  *<E Group="Молочная прод." Name="Рожок &quot;Фил.&quot; пломбир" Price="2000"/>*  *<E Group="Напитки уп." Name="Вода &quot;Бонаква&quot;" Price="2700"/>*  *<E Group="Напитки уп." Name="Вода &quot;Святой Источник&quot;" Price="1900"/>*  *<E Group="Напитки уп." Name="Вода Дана" Price="1600"/>*  *<E Group="Напитки уп." Name="Вода Демидовск." Price="1500"/>*  *<E Group="Напитки уп." Name="Голд" Price="1400"/>*  *<E Group="Напитки уп." Name="Какао с молоком" Price="1100"/>*  *<E Group="Напитки уп." Name="Кофе" Price="1000"/>*  *<E Group="Напитки уп." Name="Лимон (долька)" Price="100"/>*  *<E Group="Напитки уп." Name="Морс &quot;ФрутоНяня&quot; клюква/малина" Price="2900"/>*  *<E Group="Напитки уп." Name="Морс Здрайверы" Price="2000"/>*  *<E Group="Напитки уп." Name="Напиток &quot;Фрустайл&quot; в асс." Price="3500"/>*  *<E Group="Напитки уп." Name="Напиток из св.фруктов" Price="1000"/>*  *<E Group="Напитки уп." Name="Напиток из суш/фр" Price="1200"/>*  *<E Group="Напитки уп." Name="Напиток цитрусовый" Price="700"/>*  *<E Group="Напитки уп." Name="Напиток Шиповник" Price="1100"/>*  *<E Group="Напитки уп." Name="Нектар &quot;Любимый&quot; сад" Price="1100"/>*  *<E Group="Напитки уп." Name="Нектар J7" Price="1700"/>*  *<E Group="Напитки уп." Name="Нектар Туса-ждуса" Price="2200"/>*  *<E Group="Напитки уп." Name="Сахар" Price="100"/>*  *<E Group="Напитки уп." Name="Сок &quot;Здрайверы&quot; мультифр." Price="1400"/>*  *<E Group="Напитки уп." Name="Сок &quot;Малышам&quot;" Price="700"/>*  *<E Group="Напитки уп." Name="Сок в асс." Price="1300"/>*  *<E Group="Напитки уп." Name="Сок Каприз" Price="2400"/>*  *<E Group="Напитки уп." Name="Фрукт. десерт &quot;ФрутоНяня&quot;" Price="1800"/>*  *<E Group="Напитки уп." Name="Хол. чай Нести" Price="4500"/>*  *<E Group="Напитки уп." Name="Чай" Price="300"/>*  *<E Group="Салаты" Name="Морковь по корейски" Price="1500"/>*  *<E Group="Салаты" Name="Рыба под маринадом" Price="2400"/>*  *<E Group="Салаты" Name="Салат винегрет" Price="1300"/>*  *<E Group="Салаты" Name="Салат витаминный" Price="1000"/>*  *<E Group="Салаты" Name="Салат из белок.капусты" Price="1200"/>*  *<E Group="Салаты" Name="Салат из капус.с овощ." Price="1500"/>*  *<E Group="Салаты" Name="Салат из морк.с сахаром" Price="1000"/>*  *<E Group="Салаты" Name="Салат из огурц. и пом." Price="2000"/>*  *<E Group="Салаты" Name="Салат из помид с рас. маслом" Price="2300"/>*  *<E Group="Салаты" Name="Салат из свеж.огур." Price="2000"/>*  *<E Group="Салаты" Name="Салат из свекл. яблок" Price="1300"/>*  *<E Group="Салаты" Name="Салат из свеклы и сыра" Price="1500"/>*  *<E Group="Салаты" Name="Салат из свеклы с зел.гор" Price="1000"/>*  *<E Group="Салаты" Name="Салат крабовый" Price="1700"/>*  *<E Group="Салаты" Name="Салат Мимоза" Price="2200"/>*  *<E Group="Салаты" Name="Салат мясной" Price="3500"/>*  *<E Group="Салаты" Name="Салат рыбный" Price="2200"/>*  *<E Group="Салаты" Name="Салат столичный" Price="2700"/>*  *<E Group="Салаты" Name="Салат Цезарь" Price="3400"/>*  *<E Group="Соб. Произв." Name="Блинчики 2 шт." Price="1000"/>*  *<E Group="Соб. Произв." Name="Блинчики с джемом" Price="1200"/>*  *<E Group="Соб. Произв." Name="Блинчики с капустой" Price="1000"/>*  *<E Group="Соб. Произв." Name="Блинчики с мас/слив" Price="800"/>*  *<E Group="Соб. Произв." Name="Блинчики с медом" Price="1500"/>*  *<E Group="Соб. Произв." Name="Блинчики с мясом" Price="2800"/>*  *<E Group="Соб. Произв." Name="Блинчики с сыром 2шт." Price="3100"/>*  *<E Group="Соб. Произв." Name="Блинчики смет. и сах. 2шт." Price="1600"/>*  *<E Group="Соб. Произв." Name="Блинчики со смет. 2шт." Price="1500"/>*  *<E Group="Соб. Произв." Name="Ватрушка с повидлом" Price="800"/>*  *<E Group="Соб. Произв." Name="Колбаска сос. в тесте" Price="2200"/>*  *<E Group="Соб. Произв." Name="Оладьи" Price="1000"/>*  *<E Group="Соб. Произв." Name="Пакет" Price="20"/>*  *<E Group="Соб. Произв." Name="Пельмени отварные" Price="2100"/>*  *<E Group="Соб. Произв." Name="Пирожок с кап. и яйцом" Price="1000"/>*  *<E Group="Соб. Произв." Name="Пирожок с капустой" Price="1000"/>*  *<E Group="Соб. Произв." Name="Пирожок с карт.и лук." Price="1000"/>*  *<E Group="Соб. Произв." Name="Пирожок с мясом" Price="2200"/>*  *<E Group="Соб. Произв." Name="Пирожок с яблоками" Price="1000"/>*  *<E Group="Соб. Произв." Name="Пицца &quot;Школьная&quot;" Price="2700"/>*  *<E Group="Соб. Произв." Name="Пицца с сосисками" Price="2700"/>*  *<E Group="Соб. Произв." Name="Сарделька отварная" Price="1600"/>*  *<E Group="Соб. Произв." Name="Сосиска отварная" Price="1600"/>*  *<E Group="Соб. Произв." Name="Яблоко печ.с сахаром" Price="2200"/>*  *<E Group="Соус" Name="Маринад овощной" Price="1000"/>*  *<E Group="Соус" Name="Соус мол.сладкий" Price="300"/>*  *<E Group="Соус" Name="Соус молочный" Price="300"/>*  *<E Group="Соус" Name="Соус молочный" Price="300"/>*  *<E Group="Соус" Name="Соус сливочный" Price="300"/>*  *<E Group="Соус" Name="Соус сметанный" Price="300"/>*  *<E Group="Супы" Name="Борщ с капус. и карт." Price="1100"/>*  *<E Group="Супы" Name="Бульон куринный" Price="1500"/>*  *<E Group="Супы" Name="Рассольник ленинград." Price="1100"/>*  *<E Group="Супы" Name="Суп-пюре картоф." Price="1500"/>*  *<E Group="Супы" Name="Суп из овощей" Price="1300"/>*  *<E Group="Супы" Name="Суп из рыбы" Price="2600"/>*  *<E Group="Супы" Name="Суп картоф. с рыбой" Price="2500"/>*  *<E Group="Супы" Name="Суп крестьян. с круп." Price="1000"/>*  *<E Group="Супы" Name="Суп куринный" Price="1500"/>*  *<E Group="Супы" Name="Суп с макар. издел." Price="900"/>*  *<E Group="Супы" Name="Щи из свежей капусты" Price="1300"/>*  *<E Group="Фрукты" Name="Апельсин" Price="2000"/>*  *<E Group="Фрукты" Name="Груша" Price="2000"/>*  *<E Group="Фрукты" Name="Мандарин" Price="1200"/>*  *<E Group="Фрукты" Name="Яблоко зеленое" Price="2000"/>*  *<E Group="Фрукты" Name="Яблоко печ.с сах." Price="1500"/>*  *</M>*  *</menuList>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *</return>*  *</ns2:getMenuListResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Операция «Получение данных посещениях образовательного учреждения»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | getEnterEventList |
| **Наименование операции:** | Получение данных посещениях образовательного учреждения |
| **Назначение операции:** | Получение информации о посещении образовательного учреждения: дата, направление входа |

### Описание входных параметров

**Входные данные: getEnterEventList**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |
| 2 | startDate | Начальная дата выборки | + | xs:dateTime |  |
| 3 | endDate | Конечная дата выборки | + | xs:dateTime |  |

### Описание выходных параметров

**Выходные данные: enterEventListResult**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | enterEventList | События посещения | + | EnterEventList |  |
| 2 | resultCode | Код возврата | + | xs:long |  |
| 3 | description | Описание результата обработки | + | xs:string |  |

Параметры комплексного типа описаны в приложении «Описание общих структур данных».

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Лицевой счет не найден | Лицевой счет не найден |  |

### Контрольные примеры

**Запрос**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:getEnterEventList>*  *<contractId>07613912</contractId>*  *<startDate>2012-05-01T00:00:00Z</startDate>*  *<endDate>2012-05-20T00:00:00Z</endDate>*  *</soap:getEnterEventList>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:getEnterEventListResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<enterEventList>*  *<E DateTime="2012-05-18T12:25:02" Day="5" EnterName="Вход №1" Direction="0" TemporaryCard="1"/>*  *<E DateTime="2012-05-18T12:26:28" Day="5" EnterName="Вход №1" Direction="1" TemporaryCard="1"/>*  *<E DateTime="2012-05-18T12:38:35" Day="5" EnterName="Вход №1" Direction="7" TemporaryCard="1"/>*  *<E DateTime="2012-05-18T12:44:29" Day="5" EnterName="Вход №1" Direction="7" TemporaryCard="1"/>*  *</enterEventList>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *</return>*  *</ns2:getEnterEventListResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Операция «Включить/отключить СМС-информирование»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | enableNotificationBySMS |
| **Наименование операции:** | Включить/отключить СМС-информирование |
| **Назначение операции:** | Изменение настроек СМС-информирования лицевого счета |

### Описание входных параметров

**Входные данные: enableNotificationBySMS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |
| 2 | state | Статус | + | xs:boolean |  |

### Описание выходных параметров

**Выходные данные: result**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 2 | resultCode | Код возврата | + | xs:long |  |
| 3 | description | Описание результата обработки | + | xs:string |  |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Лицевой счет не найден | Лицевой счет не найден |  |

### Контрольные примеры

**Запрос**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:enableNotificationBySMS>*  *<contractId>200485</contractId>*  *<state>false</state>*  *</soap:enableNotificationBySMS>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:enableNotificationBySMSResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *</return>*  *</ns2:enableNotificationBySMSResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Операция «Изменить номер мобильного телефона для СМС-информирования»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | changeMobilePhone |
| **Наименование операции:** | Изменить номер мобильного телефона для СМС-информирования |
| **Назначение операции:** | Изменение настроек СМС-информирования (номер телефона) |

### Описание входных параметров

**Входные данные: changeMobilePhone**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |
| 2 | mobilePhone | Номер мобильного телефона | + | xs:string |  |

### Описание выходных параметров

**Выходные данные: result**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 2 | resultCode | Код возврата | + | xs:long |  |
| 3 | description | Описание результата обработки | + | xs:string |  |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Лицевой счет не найден | Лицевой счет не найден |  |
| 4 | 150 | Неверные данные | Неправильный формат номера |  |

### Контрольные примеры

**Запрос**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:changeMobilePhone>*  *<contractId>200485</contractId>*  *<mobilePhone>79161111111</mobilePhone>*  *</soap: changeMobilePhone>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:changeMobilePhoneResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *</return>*  *</ns2:changeMobilePhoneResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Операция «Включить/отключить E-Mail-информирование»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | enableNotificationByEmail |
| **Наименование операции:** | Включить/отключить E-Mail-информирование |
| **Назначение операции:** | Изменение настроек Email-информирования лицевого счета |

### Описание входных параметров

**Входные данные: enableNotificationByEmail**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |
| 2 | state | Статус | + | xs:boolean |  |

### Описание выходных параметров

**Выходные данные: result**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 2 | resultCode | Код возврата | + | xs:long |  |
| 3 | description | Описание результата обработки | + | xs:string |  |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Лицевой счет не найден | Лицевой счет не найден |  |

### Контрольные примеры

**Запрос**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:enableNotificationByEmail>*  *<contractId>200485</contractId>*  *<state>false</state>*  *</soap:enableNotificationByEmail>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:enableNotificationByEmailResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *</return>*  *</ns2:enableNotificationByEmailResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Операция «Изменить адрес e-mail для информирования»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | changeEmail |
| **Наименование операции:** | Изменить адрес e-mail для информирования |
| **Назначение операции:** | Изменение настроек Email-информирования (адрес) |

### Описание входных параметров

**Входные данные:** changeEmail

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |
| 2 | email | Адрес e-mail | + | xs:string |  |

### Описание выходных параметров

**Выходные данные: result**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string |  |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Лицевой счет не найден | Лицевой счет не найден |  |
| 4 | 150 | Неверные данные | Неправильный формат адреса |  |

### Контрольные примеры

**Запрос**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:changeEmail>*  *<contractId>200485</contractId>*  *<email>test@mail.ru</email>*  *</soap:changeEmail>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:changeEmailResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *</return>*  *</ns2:changeEmailResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Операция «Применить код активации»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | activateLinkingToken |
| **Наименование операции:** | Применить код активации |
| **Назначение операции:** | Применить код активации для связки с лицевым счетом |

### Описание входных параметров

**Входные данные: activateLinkingToken**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | linkingToken | Код активации | + | xs:string |  |

### Описание выходных параметров

**Выходные данные: activateLinkingTokenResult**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета |  | xs:long | Заполняется только в случае если код активации найден |
| 2 | resultCode | Код возврата | + | xs:long |  |
| 3 | description | Описание результата обработки | + | xs:string |  |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 150 | Код активации не найден | Код активации не найден |  |

### Контрольные примеры

**Запрос**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:activateLinkingToken>*  *<linkingToken>f9bcslbc5</linkingToken>*  *</soap:activateLinkingToken>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:activateLinkingTokenResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *<contractId>200485</contractId>*  *</return>*  *</ns2:activateLinkingTokenResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Операция «Сформировать код активации»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | generateLinkingToken |
| **Наименование операции:** | Сформировать код активации |
| **Назначение операции:** | Сформировать новый код активации для лицевого счета |

### Описание входных параметров

**Входные данные: activateLinkingToken**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |

### Описание выходных параметров

**Выходные данные: generateLinkingTokenResult**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |
| 2 | linkingToken | Код активации | + | xs:string |  |
| 3 | resultCode | Код возврата | + | xs:long |  |
| 4 | description | Описание результата обработки | + | xs:string |  |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Лицевой счет не найден | Лицевой счет не найден |  |

### Контрольные примеры

**Запрос**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:generateLinkingToken>*  *<contractId>200485</contractId>*  *</soap:generateLinkingToken>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:generateLinkingTokenResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *<contractId>200485</contractId>*  *<linkingToken>f9bcslbc5</linkingToken>*  *</return>*  *</ns2:generateLinkingTokenResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Операция «Сформировать код активации и отправить по номеру лицевого счета»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | sendLinkingTokenByContractId |
| **Наименование операции:** | Сформировать код активации и отправить по номеру лицевого счета |
| **Назначение операции:** | Сформировать код активации и отправить по e-mail и SMS для указанного лицевого счета |

### Описание входных параметров

**Входные данные: sendLinkingTokenByContractId**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |

### Описание выходных параметров

**Выходные данные: result**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string | В случае успешной обработки заполняется по каким каналам быт отправлен код |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 160 | У лицевого счета нет контактных данных | У лицевого счета нет контактных данных |  |

### Контрольные примеры

**Запрос**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:sendLinkingTokenByContractId>*  *<contractId>200485</contractId>*  *</soap:sendLinkingTokenByContractId>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:sendLinkingTokenByContractIdResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<resultCode>0</resultCode>*  *<description>Код активации отправлен по SMS</description>*  *</return>*  *</ns2:sendLinkingTokenByContractIdResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Операция «Сформировать код активации и отправить по e-mail и SMS»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | sendLinkingTokenByMobile |
| **Наименование операции:** | Сформировать код активации и отправить по номеру телефона |
| **Назначение операции:** | Сформировать код активации и отправить по e-mail и SMS для указанного номера телефона |

### Описание входных параметров

**Входные данные: sendLinkingTokenByMobile**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | mobilePhone | Номер телефона | + | xs:string |  |

### Описание выходных параметров

**Выходные данные: result**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string | В случае успешной обработки заполняется сколько л/с привязаны к данному номеру |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Клиент не найден | Нет лицевых счетов с данным контактным номером мобильного телефона |  |
| 4 | 150 | Неверный формат | Неверный формат номера мобильного телефона |  |

### Контрольные примеры

**Запрос**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:sendLinkingTokenByMobile>*  *<mobilePhone>79161111111</mobilePhone>*  *</soap:sendLinkingTokenByMobile>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:sendLinkingTokenByMobileResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<resultCode>0</resultCode>*  *<description>Код активации отправлен по SMS для 2 л/с</description>*  *</return>*  *</ns2:sendLinkingTokenByMobileResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Операция «Изменить размер дневного ограничения на затраты»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | changeExpenditureLimit |
| **Наименование операции:** | Установить размер дневного ограничения на затраты для указанного лицевого счета |
| **Назначение операции:** | Установить размер дневного ограничения на затраты для указанного лицевого счета |

### Описание входных параметров

**Входные данные:** changeExpenditureLimit

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |
| 2 | limit | Размер дневного ограничения | + | xs:long | Размер дневного ограничения указывается в копейках. Если 0 – то лимит не установлен. |

### Описание выходных параметров

**Выходные данные: result**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string |  |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Лицевой счет не найден | Лицевой счет не найден |  |
| 4 | 150 | Неверные данные | Лимит не может быть меньше нуля |  |

### Контрольные примеры

**Запрос**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:changeExpenditureLimit>*  *<contractId>200485</contractId>*  *<limit>5000</limit>*  *</soap:changeExpenditureLimit>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:changeExpenditureLimitResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *</return>*  *</ns2:changeExpenditureLimitResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Операция «Прикрепить СНИЛС опекуна по номеру ЛС»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | attachGuardSan |
| **Наименование операции:** | Прикрепить СНИЛС опекуна по номеру ЛС |
| **Назначение операции:** | Прикрепить СНИЛС опекуна по номеру лицевого счета |

### Описание входных параметров

**Входные данные:** attachGuardSan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |
| 2 | guardSan | СНИЛС представителя | + | xs:string |  |

### Описание выходных параметров

**Выходные данные: result**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string |  |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |

## Операция «Прикрепить СНИЛС опекуна по СНИЛС клиента»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | attachGuardSanBySan |
| **Наименование операции:** | Прикрепить СНИЛС опекуна по СНИЛС клиента |
| **Назначение операции:** | Прикрепить СНИЛС опекуна по СНИЛС клиента |

### Описание входных параметров

**Входные данные:** attachGuardSanBySan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | San | СНИЛС клиента | + | xs:string |  |
| 2 | guardSan | СНИЛС представителя | + | xs:string |  |

### Описание выходных параметров

**Выходные данные: result**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string |  |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |

## Операция «Открепить СНИЛС опекуна по номеру ЛС»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | detachGuardSan |
| **Наименование операции:** | Открепить СНИЛС опекуна по номеру ЛС |
| **Назначение операции:** | Открепить СНИЛС опекуна по номеру лицевого счета |

### Описание входных параметров

**Входные данные:** detachGuardSan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |
| 2 | guardSan | СНИЛС представителя | + | xs:string |  |

### Описание выходных параметров

**Выходные данные: result**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string |  |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |

## Операция «Открепить СНИЛС опекуна по СНИЛС клиента»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | detachGuardSanBySan |
| **Наименование операции:** | Открепить СНИЛС опекуна по СНИЛС клиента |
| **Назначение операции:** | Открепить СНИЛС опекуна по СНИЛС клиента |

### Описание входных параметров

**Входные данные:** detachGuardSanBySan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | San | СНИЛС клиента | + | xs:string |  |
| 2 | guardSan | СНИЛС представителя | + | xs:string |  |

### Описание выходных параметров

**Выходные данные: result**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string |  |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |

## Операция «Получить список прикрепленных лицевых счетов по СНИЛС опекуна»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | getClientsByGuardSan |
| **Наименование операции:** | Получить список прикрепленных лицевых счетов по СНИЛС опекуна |
| **Назначение операции:** | Используется внешними порталами для получения списка лицевых счетов в личном кабинете опекуна |

### Описание входных параметров

**Входные данные:** getClientsByGuardSan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | guardSan | СНИЛС опекуна | + | xs:string |  |

### Описание выходных параметров

**Выходные данные: ClientsData**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string |  |
| 3 | clientList | Данные прикрепленных лицевых счетов | - | ClientItem |  |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |

### Контрольные примеры

**Запрос**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:getClientsByGuardSan>*  *<san>16721468173</san>*  *</soap:getClientsByGuardSan>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:getClientsByGuardSanResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<clientList>*  *<Client ContractId="10500015" San=""/>*  *<Client ContractId="200485" San=""/>*  *</clientList>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *</return>*  *</ns2:getClientsByGuardSanResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Операция «Получить типы уведомлений»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | getClientNotificationTypes |
| **Наименование операции:** | Получить доступные типы уведомлений клиентов |
| **Назначение операции:** | Получить доступные типы уведомлений клиентов для последующей настройки |

### Описание входных параметров

**Входные данные:** отсутствуют

### Описание выходных параметров

**Выходные данные: ClientNotificationSettingsResult**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string |  |
| 3 | settings | Список доступных типов уведомлений | + | ClientNotificationSettingsItem |  |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |

## Операция «Получить настройки уведомлений клиента по номеру ЛС»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | getClientNotificationSettings |
| **Наименование операции:** | Получить настройки уведомлений клиента по номеру ЛС |
| **Назначение операции:** | Получить настройки уведомлений клиента по номеру ЛС |

### Описание входных параметров

**Входные данные:** getClientNotificationSettings

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:string |  |

### Описание выходных параметров

**Выходные данные: result**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string |  |
| 3 | settings | Список установленных типов уведомлений | + | ClientNotificationSettingsItem |  |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Лицевой счет не найден | Лицевой счет не найден |  |

## Операция «Установить настройки уведомлений клиента по номеру ЛС»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | setClientNotificationSettings |
| **Наименование операции:** | Установить настройки уведомлений клиента по номеру ЛС |
| **Назначение операции:** | Установить настройки уведомлений клиента по номеру ЛС |

### Описание входных параметров

**Входные данные:** setClientNotificationSettings

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:string |  |
| 2 | notificationType | Список кодов типов уведомлений | + | xs:long | Коды типов уведомлений должны быть получены из getClientNotificationTypes |

### Описание выходных параметров

**Выходные данные: result**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string |  |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Лицевой счет не найден | Лицевой счет не найден |  |

## Операция «Получить статистику по ЛС»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | getClientStats |
| **Наименование операции:** | Получить статистику по ЛС |
| **Назначение операции:** | Получить статистику операций по ЛС |

### Описание входных параметров

**Входные данные:** setClientNotificationSettings

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |
| 2 | startDate | Начальная дата | + | xs:date |  |
| 3 | endDate | Конечная дата | + | xs:date | Конечная дата не включается в выборку |
| 4 | type | Тип статистики | + | xs:int | 0 – общая статистика |

### Описание выходных параметров

**Выходные данные: result**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string |  |
| 3 | stats | Статитика | - | xs:string | Строка в формате JSON:  {"stat":[{"id":"<идентификатор параметра>","name":"наименование параметра","value":"значение"}]} |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Лицевой счет не найден | Лицевой счет не найден |  |

## Перевод средств клиента между субсчетами

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | transferBalance |
| **Наименование операции:** | Перевод средств клиента между субсчетами |
| **Назначение операции:** | Перевод средств клиента между субсчетами |

### Описание входных параметров

**Входные данные:** transferBalance

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |
| 2 | fromSub | Порядковый номер лицевого счета с которого произвести перевод | + | xs:int | Вводятся последние 2 цифры лицевого счета если счет основной то пишем 0 |
| 3 | toSub | Порядковый номер лицевого счета в которого произвести перевод | + | xs:int |  |
| 4 | amount | Сумма перевода | + | xs: long | Сумма пишется в копейках |

### Описание выходных параметров

**Выходные данные: result**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string |  |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Лицевой счет не найден | Лицевой счет не найден |  |
| 4 | 170 | Недостаточно средств на лицевом счете | При не хвадке стредств для перевода |  |

## Получение списка учащихся, которые после оплаты платного плана питания ушли в минус

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | getStudentsByCanNotConfirmPayment |
| **Наименование операции:** | Получение списка учащихся, которые после оплаты платного плана питания ушли в минус |
| **Назначение операции:** | Получение списка учащихся, которые после оплаты платного плана питания ушли в минус |

### Описание входных параметров

**Входные данные:** getStudentsByCanNotConfirmPayment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |

### Описание выходных параметров

**Выходные данные: result**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string |  |
| 3 | clientConfirmPaymentData | Спиок учеников | + | StudentsConfirmPaymentList |  |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Лицевой счет не найден | Лицевой счет не найден |  |

## Операция «Получение данных о покупках с субчета по абоненскому питанию лицевого счета»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | getPurchaseSubscriptionFeedingList |
| **Наименование операции:** | Получение данных о покупках с субсчета по абоненскому питанию |
| **Назначение операции:** | Предоставление информации о покупках: дата, сумма, состав |

### Описание входных параметров

**Входные данные: getPurchaseSubscriptionFeedingList**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | san | Снилс клиента | + | xs:string |  |
| 2 | startDate | Начальная дата выборки | + | xs:dateTime |  |
| 3 | endDate | Конечная дата выборки | + | xs:dateTime |  |

### Описание выходных параметров

**Выходные данные: purchaseListResult**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | purchaseList | Список покупок | + | PurchaseListExt |  |
| 2 | resultCode | Код возврата | + | xs:long |  |
| 3 | description | Описание результата обработки | + | xs:string |  |

Параметры комплексного типа описаны в приложении «Описание общих структур данных».

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Лицевой счет не найден | Лицевой счет не найден |  |

### Контрольные примеры

**Запрос**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:getPurchaseList>*  *<*san*>12312399</*san*>*  *<startDate>2014-01-17T00:00:00Z</startDate>*  *<endDate>2014-02-17T00:00:00Z</endDate>*  *</soap:getPurchaseList>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:getPurchaseListResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<purchaseList>*  *<P Time="2012-02-14T21:37:36" Sum="3000" SocDiscount="0" TrdDiscount="0" Donation="0" ByCash="0" ByCard="3000" IdOfCard="123456789">*  *<E Name="Завтрак АП " Sum="500" Amount="1" Type="2"/>*  *<E Name="Обед АП " Sum="1500" Amount="1" Type="2"/>*  *<E Name="Полдник АП " Sum="1000" Amount="1" Type="2"/>*  *</P>*  *</purchaseList>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *</return>*  *</ns2:getPurchaseListResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Операция «Получение данных о пополнениях субсчета по абоненскому питанию»

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | getPaymentSubscriptionFeedingList |
| **Наименование операции:** | Получение данных о пополнениях субчета по абоненскому питанию |
| **Назначение операции:** | Предоставление информации о пополнениях: дата, сумма, источник |

### Описание входных параметров

**Входные данные: getPaymentList**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | san | Снилс клиента | + | xs:string |  |
| 2 | startDate | Начальная дата выборки | + | xs:dateTime |  |
| 3 | endDate | Конечная дата выборки | + | xs:dateTime |  |

### Описание выходных параметров

**Выходные данные: paymentListResult**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | paymentList | Список пополнений | + | PaymentList |  |
| 2 | resultCode | Код возврата | + | xs:long |  |
| 3 | description | Описание результата обработки | + | xs:string |  |

Параметры комплексного типа описаны в приложении «Описание общих структур данных».

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Лицевой счет не найден | Лицевой счет не найден |  |

### Контрольные примеры

**Запрос**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:getPaymentList>*  *<*san*>12312399</*san*>*  *<startDate>2014-01-01T00:00:00Z</startDate>*  *<endDate>2014-02-01T00:00:00Z</endDate>*  *</soap:getPaymentList>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:getPaymentListResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<paymentList>*  *<P Time="2012-02-15T12:26:56" Sum="999" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-15T12:26:58" Sum="-999" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-15T12:48:27" Sum="999" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-15T12:48:27" Sum="-999" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-15T12:58:35" Sum="999" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-15T12:58:37" Sum="-999" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-15T13:55:59" Sum="2000" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-16T20:11:56" Sum="10000" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-16T20:13:25" Sum="10000" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-16T20:38:35" Sum="5000" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-16T20:39:18" Sum="100000" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-16T21:13:28" Sum="100000" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-16T21:19:53" Sum="100000" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-17T11:40:37" Sum="999" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-17T11:40:38" Sum="-999" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-17T11:41:46" Sum="999" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-17T11:41:47" Sum="-999" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-17T11:50:54" Sum="999" Origin="Конкорд-ПА - Банкоматы"/>*  *<P Time="2012-02-17T11:50:55" Sum="-999" Origin="Конкорд-ПА - Банкоматы"/>*  *</paymentList>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *</return>*  *</ns2:getPaymentListResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Перевод средств клиента между субсчетами

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | transferBalanceBySan |
| **Наименование операции:** | Перевод средств клиента между субсчетами |
| **Назначение операции:** | Перевод средств клиента между субсчетами |

### Описание входных параметров

**Входные данные:** transferBalance

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | san | Снилс клиента | + | xs:string |  |
| 2 | fromSub | Порядковый номер лицевого счета с которого произвести перевод | + | xs:int | Вводятся последние 2 цифры лицевого счета если счет основной то пишем 0 |
| 3 | toSub | Порядковый номер лицевого счета в которого произвести перевод | + | xs:int |  |
| 4 | amount | Сумма перевода | + | xs: long | Сумма пишется в копейках |

### Описание выходных параметров

**Выходные данные: result**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string |  |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Лицевой счет не найден | Лицевой счет не найден |  |
| 4 | 170 | Недостаточно средств на лицевом счете | При не хвадке стредств для перевода |  |

### Контрольные примеры

**Запрос**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:getPaymentList>*  *<san>12312399</san>*  *<fromSub>20048500</fromSub>*  *<toSub>20048501</toSub>*  *<amount>3000</amount>*  *</soap:getPaymentList>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:transferBalanceResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *</return>*  *</ns2:transferBalanceResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Подключение подписки на АП клиенту

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | createSubscriptionFeeding |
| **Наименование операции:** | Подключение подписки на АП по номеру ЛС клиента |
| **Назначение операции:** | Подключение подписки на АП по номеру ЛС клиента |

### Описание входных параметров

**Входные данные:** createSubscriptionFeeding

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |
| 2 | cycleDiagram | Список идентификаторов комплексов питания по дням недели | + | CycleDiagramIn | Идентификаторы комплексов перечисляются через «;» |

Параметры комплексного типа описаны в приложении «Описание общих структур данных».

### Описание выходных параметров

**Выходные данные:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string |  |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Клиент не найден по данному ЛС | Клиент не найден по данному ЛС |  |
| 4 | 190 | У клиента уже есть активная подписка на АП. | Клиенту уже ранее была подключена подписка на АП |  |
| 5 | 180 | Отсутствуют настройки абонементного питания для организации клиента | Отсутствуют настройки абонементного питания для организации клиента. Без них нельзя активировать подписку на АП. |  |

### Контрольные примеры

**Запрос:**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:createSubscriptionFeeding>*  *<contractId>600064</contractId>*  *<cycleDiagram>*  *<Monday>2</Monday>*  *<Tuesday>2</Tuesday>*  *<Wednesday>2</Wednesday>*  *<Thursday>2</Thursday>*  *<Friday>2</Friday>*  *<Saturday>2</Saturday>*  *<Sunday>2</Sunday>*  *</cycleDiagram>*  *</soap:createSubscriptionFeeding>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения:**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:createSubscriptionFeedingResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<resultCode>0</resultCode>*  *<description>OK </description>*  *</return>*  *</ns2:createSubscriptionFeedingResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Получение данных об активной подписке АП по номеру ЛС

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | findSubscriptionFeeding |
| **Наименование операции:** | Получение данных об активной подписке АП по номеру ЛС |
| **Назначение операции:** | Получение данных об активной подписке АП по номеру ЛС |

### Описание входных параметров

**Входные данные:** findSubscriptionFeeding

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |

### Описание выходных параметров

**Выходные данные:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string |  |
| 3 | IdOfSubscriptionFeeding | Идентификатор подписки |  | xs:long |  |
| 4 | DateActivate | Дата активации подписки |  | xs:dateTime |  |
| 5 | LastDatePause | Дата последней приостановки |  | xs:dateTime |  |
| 6 | Suspended | Флаг приостановки подписки |  | xs:boolean | true – если подписка приостановлена. |

Если у клиента нет подключенной подписки, параметры **IdOfSubscriptionFeeding**, **DateActivate**, **LastDatePause**, **Suspended** отсутствуют в ответе.

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Клиент не найден по данному ЛС | Клиент не найден по данному ЛС |  |

### Контрольные примеры

**Запрос:**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:findSubscriptionFeeding>*  *<contractId>600049</contractId>*  *</soap:findSubscriptionFeeding>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения:**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:findSubscriptionFeedingResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *<IdOfSubscriptionFeeding>15</IdOfSubscriptionFeeding>*  *<DateActivate>2013-12-02T18:21:11.694+04:00</DateActivate>*  *<LastDatePause>2013-12-03T15:48:24.166+04:00</LastDatePause>*  *<Suspended>true</Suspended>*  *</return>*  *</ns2:findSubscriptionFeedingResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Приостановка действия подписки на АП по номеру ЛС

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | suspendSubscriptionFeeding |
| **Наименование операции:** | Приостановить действие подписки на АП по номеру ЛС клиента |
| **Назначение операции:** | Приостановить действие подписки на АП по номеру ЛС клиента |

### Описание входных параметров

**Входные данные:** suspendSubscriptionFeeding

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |

### Описание выходных параметров

**Выходные данные:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string |  |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Клиент не найден по данному ЛС | Клиент не найден по данному ЛС |  |

### Контрольные примеры

**Запрос:**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:suspendSubscriptionFeeding>*  *<contractId>600023</contractId>*  *</soap:suspendSubscriptionFeeding>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения:**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:suspendSubscriptionFeedingResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *</return>*  *</ns2:suspendSubscriptionFeedingResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Возобновление подписки на АП по номеру ЛС

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | reopenSubscriptionFeeding |
| **Наименование операции:** | Возобновить подписку на АП по номеру ЛС клиента |
| **Назначение операции:** | Возобновить подписку на АП по номеру ЛС клиента |

### Описание входных параметров

**Входные данные:** reopenSubscriptionFeeding

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |

### Описание выходных параметров

**Выходные данные:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string |  |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Клиент не найден по данному ЛС | Клиент не найден по данному ЛС |  |

### Контрольные примеры

**Запрос:**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:reopenSubscriptionFeeding>*  *<contractId>600023</contractId>*  *</soap:reopenSubscriptionFeeding>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения:**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:reopenSubscriptionFeedingResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *</return>*  *</ns2:reopenSubscriptionFeedingResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Редактирование плана питания по АП (создание новой циклограммы) по номеру ЛС

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | editSubscriptionFeedingPlan |
| **Наименование операции:** | Редактирование плана питания по АП (создание новой циклограммы) по номеру ЛС клиента |
| **Назначение операции:** | Редактирование плана питания по АП (создание новой циклограммы) по номеру ЛС клиента |

### Описание входных параметров

**Входные данные:** editSubscriptionFeedingPlan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |
| 2 | cycleDiagram | Список идентификаторов комплексов питания по дням недели | + | CycleDiagramIn | Идентификаторы комплексов перечисляются через «;» |

Параметры комплексного типа описаны в приложении «Описание общих структур данных».

### Описание выходных параметров

**Выходные данные:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string |  |
| 3 | GlobalId | Идентификатор созданной циклограммы |  | xs:long |  |
| 4 | Monday | Список идентификаторов комплексов питания на понедельник |  | xs:string |  |
| 5 | MondayPrice | Общая цена понедельника |  | xs:long | Цена в копейках |
| 6 | Tuesday | Список идентификаторов комплексов питания на вторник |  | xs:string |  |
| 7 | TuesdayPrice | Общая цена вторника |  | xs:long | Цена в копейках |
| 8 | Wednesday | Список идентификаторов комплексов питания на среду |  | xs:string |  |
| 9 | WednesdayPrice | Общая цена среды |  | xs:long | Цена в копейках |
| 10 | Thursday | Список идентификаторов комплексов питания на четверг |  | xs:string |  |
| 11 | ThursdayPrice | Общая цена четверга |  | xs:long | Цена в копейках |
| 12 | Friday | Список идентификаторов комплексов питания на пятницу |  | xs:string |  |
| 13 | FridayPrice | Общая цена пятницы |  | xs:long | Цена в копейках |
| 14 | Saturday | Список идентификаторов комплексов питания на субботу |  | xs:string |  |
| 15 | SaturdayPrice | Общая цена субботы |  | xs:long | Цена в копейках |
| 16 | DateActivationDiagram | Дата активации циклограммы |  | xs:dateTime |  |
| 17 | StateDiagram | Состояние циклограммы |  | xs:int | Возможные значения:  0 – активная,  1 – в ожидании активации,  2 - блокированная |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Клиент не найден по данному ЛС | Клиент не найден по данному ЛС |  |
| 4 | 180 | Отсутствуют настройки абонементного питания для организации клиента | Отсутствуют настройки абонементного питания для организации клиента. Без них нельзя создать циклограмму. |  |

### Контрольные примеры

**Запрос:**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:editSubscriptionFeedingPlan>*  *<contractId>600023</contractId>*  *<cycleDiagram>*  *<Monday>1;3</Monday>*  *<Tuesday>1</Tuesday>*  *<Wednesday>0;1;4</Wednesday>*  *<Thursday>1</Thursday>*  *<Friday>1;4</Friday>*  *<Saturday>1</Saturday>*  *<Sunday>1</Sunday>*  *</cycleDiagram>*  *</soap:editSubscriptionFeedingPlan>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения:**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:editSubscriptionFeedingPlanResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *<GlobalId>23</GlobalId>*  *<Monday>1;3</Monday>*  *<Tuesday>1</Tuesday>*  *<Wednesday>0;1;4</Wednesday>*  *<Thursday>1</Thursday>*  *<Friday>1;4</Friday>*  *<Saturday>1</Saturday>*  *<DateActivationDiagram>2014-01-20T00:00:00+04:00</DateActivationDiagram>*  *<StateDiagram>1</StateDiagram>*  *<MondayPrice>6000</MondayPrice>*  *<TuesdayPrice>3000</TuesdayPrice>*  *<WednesdayPrice>9000</WednesdayPrice>*  *<ThursdayPrice>3000</ThursdayPrice>*  *<FridayPrice>6000</FridayPrice>*  *<SaturdayPrice>3000</SaturdayPrice>*  *</return>*  *</ns2:editSubscriptionFeedingPlanResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Получение данных о текущем плане питания по АП по номеру ЛС

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | findClientCycleDiagram |
| **Наименование операции:** | Получение данных о текущем плане питания по АП (т.е. данных об активной циклограмме питания) по номеру ЛС клиента |
| **Назначение операции:** | Получение данных о текущем плане питания по АП (т.е. данных об активной циклограмме питания) по номеру ЛС клиента |

### Описание входных параметров

**Входные данные:** findClientCycleDiagram

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |

### Описание выходных параметров

**Выходные данные:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string |  |
| 3 | GlobalId | Идентификатор циклограммы |  | xs:long |  |
| 4 | Monday | Список идентификаторов комплексов питания на понедельник |  | xs:string |  |
| 5 | MondayPrice | Общая цена понедельника |  | xs:long | Цена в копейках |
| 6 | Tuesday | Список идентификаторов комплексов питания на вторник |  | xs:string |  |
| 7 | TuesdayPrice | Общая цена вторника |  | xs:long | Цена в копейках |
| 8 | Wednesday | Список идентификаторов комплексов питания на среду |  | xs:string |  |
| 9 | WednesdayPrice | Общая цена среды |  | xs:long | Цена в копейках |
| 10 | Thursday | Список идентификаторов комплексов питания на четверг |  | xs:string |  |
| 11 | ThursdayPrice | Общая цена четверга |  | xs:long | Цена в копейках |
| 12 | Friday | Список идентификаторов комплексов питания на пятницу |  | xs:string |  |
| 13 | FridayPrice | Общая цена пятницы |  | xs:long | Цена в копейках |
| 14 | Saturday | Список идентификаторов комплексов питания на субботу |  | xs:string |  |
| 15 | SaturdayPrice | Общая цена субботы |  | xs:long | Цена в копейках |
| 16 | DateActivationDiagram | Дата активации циклограммы |  | xs:dateTime |  |
| 17 | StateDiagram | Состояние циклограммы |  | xs:int | Всегда будет равно 0 (активная) |

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Клиент не найден по данному ЛС | Клиент не найден по данному ЛС |  |

### Контрольные примеры

**Запрос:**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:findClientCycleDiagram>*  *<contractId>600023</contractId>*  *</soap:findClientCycleDiagram>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения:**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:findClientCycleDiagramResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *<GlobalId>10</GlobalId>*  *<Monday>2;9;1</Monday>*  *<Tuesday>3</Tuesday>*  *<Wednesday>6;4</Wednesday>*  *<Thursday>6;4</Thursday>*  *<Friday>4</Friday>*  *<DateActivationDiagram>2013-12-07T16:19:52.045+04:00</DateActivationDiagram>*  *<StateDiagram>0</StateDiagram>*  *<MondayPrice>9000</MondayPrice>*  *<TuesdayPrice>3000</TuesdayPrice>*  *<WednesdayPrice>6000</WednesdayPrice>*  *<ThursdayPrice>6000</ThursdayPrice>*  *<FridayPrice>3000</FridayPrice>*  *<SaturdayPrice>0</SaturdayPrice>*  *<SundayPrice>0</SundayPrice>*  *</return>*  *</ns2:findClientCycleDiagramResponse>*  *</soap:Body>*  *</soap:Envelope>* |

## Получить список комплексов, которые участвуют в АП, по номеру ЛС

### Общие сведения

|  |  |
| --- | --- |
| **Код операции:** | findComplexesWithSubFeeding |
| **Наименование операции:** | Получить список комплексов, которые участвуют в АП, по номеру ЛС клиента |
| **Назначение операции:** | Получить список комплексов, которые участвуют в АП, по номеру ЛС клиента |

### Описание входных параметров

**Входные данные:** findComplexesWithSubFeeding

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | contractId | Номер лицевого счета | + | xs:long |  |

### Описание выходных параметров

**Выходные данные:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | resultCode | Код возврата | + | xs:long |  |
| 2 | description | Описание результата обработки | + | xs:string |  |
| 3 | ComplexInfoList | Список комплексов (элементов ComplexInfo) | + | ComplexInfoList | Может быть пустым, если на текущий день у ОУ нет комплексов. |
| 4 | ComplexInfo | Описание одного комплекса |  | ComplexInfoExt |  |

Параметры комплексного типа описаны в приложении «Описание общих структур данных».

### Ошибки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Код возврата | Описание кода возврата | Условия возникновения | Комментарий |
| 1 | 0 | Успешно | Успешная обработка |  |
| 2 | 100 | Внутренняя ошибка | Внутренняя ошибка сервера при обработке |  |
| 3 | 110 | Клиент не найден по данному ЛС | Клиент не найден по данному ЛС |  |

### Контрольные примеры

**Запрос:**

|  |
| --- |
| *<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:soap="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<soapenv:Header/>*  *<soapenv:Body>*  *<soap:findComplexesWithSubFeeding>*  *<contractId>600023</contractId>*  *</soap:findComplexesWithSubFeeding>*  *</soapenv:Body>*  *</soapenv:Envelope>* |

**Ответ на запрос в случае успешного исполнения:**

|  |
| --- |
| *<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">*  *<soap:Body>*  *<ns2:findComplexesWithSubFeedingResponse xmlns:ns2="http://soap.integra.partner.web.processor.ecafe.axetta.ru/">*  *<return>*  *<resultCode>0</resultCode>*  *<description>OK</description>*  *<ComplexInfoList>*  *<ComplexInfo idOfComplexInfo="807" idOfComplex="3" complexName="Комплекс №4" menuDate="2014-01-15T00:00:00+04:00" currentPrice="3000" usedSubscriptionFeeding="1"/>*  *<ComplexInfo idOfComplexInfo="808" idOfComplex="4" complexName="Комплекс №5" menuDate="2014-01-15T00:00:00+04:00" currentPrice="3000" usedSubscriptionFeeding="1"/>*  *<ComplexInfo idOfComplexInfo="810" idOfComplex="6" complexName="Комплекс №7" menuDate="2014-01-15T00:00:00+04:00" currentPrice="3000" usedSubscriptionFeeding="1"/>*  *<ComplexInfo idOfComplexInfo="812" idOfComplex="8" complexName="Комплекс №9" menuDate="2014-01-15T00:00:00+04:00" currentPrice="3000" usedSubscriptionFeeding="1"/>*  *<ComplexInfo idOfComplexInfo="805" idOfComplex="1" complexName="Комплекс №2" menuDate="2014-01-15T00:00:00+04:00" currentPrice="3000" usedSubscriptionFeeding="1"/>*  *<ComplexInfo idOfComplexInfo="806" idOfComplex="2" complexName="Комплекс №3" menuDate="2014-01-15T00:00:00+04:00" currentPrice="3000" usedSubscriptionFeeding="1"/>*  *<ComplexInfo idOfComplexInfo="809" idOfComplex="5" complexName="Комплекс №6" menuDate="2014-01-15T00:00:00+04:00" currentPrice="3000" usedSubscriptionFeeding="1"/>*  *<ComplexInfo idOfComplexInfo="811" idOfComplex="7" complexName="Комплекс №8" menuDate="2014-01-15T00:00:00+04:00" currentPrice="3000" usedSubscriptionFeeding="1"/>*  *<ComplexInfo idOfComplexInfo="813" idOfComplex="9" complexName="Комплекс №10" menuDate="2014-01-15T00:00:00+04:00" currentPrice="3000" usedSubscriptionFeeding="1"/>*  *</ComplexInfoList>*  *</return>*  *</ns2:findComplexesWithSubFeedingResponse>*  *</soap:Body>*  *</soap:Envelope>* |

1. Приложения

Описание сервиса (WSDL)

|  |
| --- |
| <?xml version='1.0' encoding='UTF-8'?><wsdl:definitions name="ClientRoomControllerWSService" targetNamespace="http://soap.integra.partner.web.processor.ecafe.axetta.ru/" xmlns:ns1="http://schemas.xmlsoap.org/soap/http" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://soap.integra.partner.web.processor.ecafe.axetta.ru/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">  <wsdl:types>  <xs:schema elementFormDefault="unqualified" targetNamespace="http://soap.integra.partner.web.processor.ecafe.axetta.ru/" version="1.0" xmlns:tns="http://soap.integra.partner.web.processor.ecafe.axetta.ru/" xmlns:xs="http://www.w3.org/2001/XMLSchema">  <xs:element name="Data">  <xs:complexType>  <xs:sequence>  <xs:element minOccurs="0" name="ClientSummary" type="tns:ClientSummary"/>  <xs:element minOccurs="0" name="ClientSummaryExt" type="tns:ClientSummaryExt"/>  <xs:element minOccurs="0" name="PurchaseList" type="tns:PurchaseList"/>  <xs:element minOccurs="0" name="PurchaseListExt" type="tns:PurchaseListExt"/>  <xs:element minOccurs="0" name="PaymentList" type="tns:PaymentList"/>  <xs:element minOccurs="0" name="MenuList" type="tns:MenuList"/>  <xs:element minOccurs="0" name="MenuListExt" type="tns:MenuListExt"/>  <xs:element minOccurs="0" name="CardList" type="tns:CardList"/>  <xs:element minOccurs="0" name="EnterEventList" type="tns:EnterEventList"/>  <xs:element minOccurs="0" name="CirculationItemList" type="tns:CirculationItemList"/>  <xs:element minOccurs="0" name="ComplexDateList" type="tns:ComplexDateList"/>  <xs:element minOccurs="0" name="QuestionaryList" type="tns:QuestionaryList"/>  </xs:sequence>  <xs:attribute name="IdOfContract" type="xs:long"/>  <xs:attribute name="ResultCode" type="xs:long"/>  <xs:attribute name="Description" type="xs:string"/>  </xs:complexType>  </xs:element>  <xs:element name="activateLinkingToken" type="tns:activateLinkingToken"/>  <xs:element name="activateLinkingTokenResponse" type="tns:activateLinkingTokenResponse"/>  <xs:element name="attachGuardSan" type="tns:attachGuardSan"/>  <xs:element name="attachGuardSanBySan" type="tns:attachGuardSanBySan"/>  <xs:element name="attachGuardSanBySanResponse" type="tns:attachGuardSanBySanResponse"/>  <xs:element name="attachGuardSanResponse" type="tns:attachGuardSanResponse"/>  <xs:element name="authorizeClient" type="tns:authorizeClient"/>  <xs:element name="authorizeClientResponse" type="tns:authorizeClientResponse"/>  <xs:element name="changeComplaintStatusToConsideration" type="tns:changeComplaintStatusToConsideration"/>  <xs:element name="changeComplaintStatusToConsiderationResponse" type="tns:changeComplaintStatusToConsiderationResponse"/>  <xs:element name="changeComplaintStatusToInvestigation" type="tns:changeComplaintStatusToInvestigation"/>  <xs:element name="changeComplaintStatusToInvestigationResponse" type="tns:changeComplaintStatusToInvestigationResponse"/>  <xs:element name="changeEmail" type="tns:changeEmail"/>  <xs:element name="changeEmailResponse" type="tns:changeEmailResponse"/>  <xs:element name="changeExpenditureLimit" type="tns:changeExpenditureLimit"/>  <xs:element name="changeExpenditureLimitResponse" type="tns:changeExpenditureLimitResponse"/>  <xs:element name="changeMobilePhone" type="tns:changeMobilePhone"/>  <xs:element name="changeMobilePhoneResponse" type="tns:changeMobilePhoneResponse"/>  <xs:element name="changePassword" type="tns:changePassword"/>  <xs:element name="changePasswordResponse" type="tns:changePasswordResponse"/>  <xs:element name="changePaymentOrderStatus" type="tns:changePaymentOrderStatus"/>  <xs:element name="changePaymentOrderStatusResponse" type="tns:changePaymentOrderStatusResponse"/>  <xs:element name="changePersonalInfo" type="tns:changePersonalInfo"/>  <xs:element name="changePersonalInfoResponse" type="tns:changePersonalInfoResponse"/>  <xs:element name="checkPasswordRestoreRequest" type="tns:checkPasswordRestoreRequest"/>  <xs:element name="checkPasswordRestoreRequestResponse" type="tns:checkPasswordRestoreRequestResponse"/>  <xs:element name="createPaymentOrder" type="tns:createPaymentOrder"/>  <xs:element name="createPaymentOrderResponse" type="tns:createPaymentOrderResponse"/>  <xs:element name="detachGuardSan" type="tns:detachGuardSan"/>  <xs:element name="detachGuardSanBySan" type="tns:detachGuardSanBySan"/>  <xs:element name="detachGuardSanBySanResponse" type="tns:detachGuardSanBySanResponse"/>  <xs:element name="detachGuardSanResponse" type="tns:detachGuardSanResponse"/>  <xs:element name="enableNotificationByEmail" type="tns:enableNotificationByEmail"/>  <xs:element name="enableNotificationByEmailResponse" type="tns:enableNotificationByEmailResponse"/>  <xs:element name="enableNotificationBySMS" type="tns:enableNotificationBySMS"/>  <xs:element name="enableNotificationBySMSResponse" type="tns:enableNotificationBySMSResponse"/>  <xs:element name="excludeGoodFromProhibition" type="tns:excludeGoodFromProhibition"/>  <xs:element name="excludeGoodFromProhibitionResponse" type="tns:excludeGoodFromProhibitionResponse"/>  <xs:element name="excludeGoodGroupFromProhibition" type="tns:excludeGoodGroupFromProhibition"/>  <xs:element name="excludeGoodGroupFromProhibitionResponse" type="tns:excludeGoodGroupFromProhibitionResponse"/>  <xs:element name="generateLinkingToken" type="tns:generateLinkingToken"/>  <xs:element name="generateLinkingTokenResponse" type="tns:generateLinkingTokenResponse"/>  <xs:element name="getActiveMenuQuestions" type="tns:getActiveMenuQuestions"/>  <xs:element name="getActiveMenuQuestionsResponse" type="tns:getActiveMenuQuestionsResponse"/>  <xs:element name="getBanks" type="tns:getBanks"/>  <xs:element name="getBanksResponse" type="tns:getBanksResponse"/>  <xs:element name="getCardList" type="tns:getCardList"/>  <xs:element name="getCardListBySan" type="tns:getCardListBySan"/>  <xs:element name="getCardListBySanResponse" type="tns:getCardListBySanResponse"/>  <xs:element name="getCardListResponse" type="tns:getCardListResponse"/>  <xs:element name="getChronopayConfig" type="tns:getChronopayConfig"/>  <xs:element name="getChronopayConfigResponse" type="tns:getChronopayConfigResponse"/>  <xs:element name="getCirculationList" type="tns:getCirculationList"/>  <xs:element name="getCirculationListResponse" type="tns:getCirculationListResponse"/>  <xs:element name="getClientSmsList" type="tns:getClientSmsList"/>  <xs:element name="getClientSmsListResponse" type="tns:getClientSmsListResponse"/>  <xs:element name="getClientStats" type="tns:getClientStats"/>  <xs:element name="getClientStatsResponse" type="tns:getClientStatsResponse"/>  <xs:element name="getClientsByGuardSan" type="tns:getClientsByGuardSan"/>  <xs:element name="getClientsByGuardSanResponse" type="tns:getClientsByGuardSanResponse"/>  <xs:element name="getComplexList" type="tns:getComplexList"/>  <xs:element name="getComplexListResponse" type="tns:getComplexListResponse"/>  <xs:element name="getContractIdByCardNo" type="tns:getContractIdByCardNo"/>  <xs:element name="getContractIdByCardNoResponse" type="tns:getContractIdByCardNoResponse"/>  <xs:element name="getDishProhibitionsList" type="tns:getDishProhibitionsList"/>  <xs:element name="getDishProhibitionsListResponse" type="tns:getDishProhibitionsListResponse"/>  <xs:element name="getEnterEventList" type="tns:getEnterEventList"/>  <xs:element name="getEnterEventListBySan" type="tns:getEnterEventListBySan"/>  <xs:element name="getEnterEventListBySanResponse" type="tns:getEnterEventListBySanResponse"/>  <xs:element name="getEnterEventListResponse" type="tns:getEnterEventListResponse"/>  <xs:element name="getGroupListByOrg" type="tns:getGroupListByOrg"/>  <xs:element name="getGroupListByOrgResponse" type="tns:getGroupListByOrgResponse"/>  <xs:element name="getHiddenPages" type="tns:getHiddenPages"/>  <xs:element name="getHiddenPagesResponse" type="tns:getHiddenPagesResponse"/>  <xs:element name="getIdOfClient" type="tns:getIdOfClient"/>  <xs:element name="getIdOfClientResponse" type="tns:getIdOfClientResponse"/>  <xs:element name="getIdOfContragent" type="tns:getIdOfContragent"/>  <xs:element name="getIdOfContragentResponse" type="tns:getIdOfContragentResponse"/>  <xs:element name="getListOfComplaintBookEntriesByClient" type="tns:getListOfComplaintBookEntriesByClient"/>  <xs:element name="getListOfComplaintBookEntriesByClientResponse" type="tns:getListOfComplaintBookEntriesByClientResponse"/>  <xs:element name="getListOfComplaintBookEntriesByOrg" type="tns:getListOfComplaintBookEntriesByOrg"/>  <xs:element name="getListOfComplaintBookEntriesByOrgResponse" type="tns:getListOfComplaintBookEntriesByOrgResponse"/>  <xs:element name="getListOfGoods" type="tns:getListOfGoods"/>  <xs:element name="getListOfGoodsResponse" type="tns:getListOfGoodsResponse"/>  <xs:element name="getListOfProducts" type="tns:getListOfProducts"/>  <xs:element name="getListOfProductsResponse" type="tns:getListOfProductsResponse"/>  <xs:element name="getMenuList" type="tns:getMenuList"/>  <xs:element name="getMenuListByOrg" type="tns:getMenuListByOrg"/>  <xs:element name="getMenuListByOrgResponse" type="tns:getMenuListByOrgResponse"/>  <xs:element name="getMenuListBySan" type="tns:getMenuListBySan"/>  <xs:element name="getMenuListBySanResponse" type="tns:getMenuListBySanResponse"/>  <xs:element name="getMenuListResponse" type="tns:getMenuListResponse"/>  <xs:element name="getNotificationSettings" type="tns:getNotificationSettings"/>  <xs:element name="getNotificationSettingsResponse" type="tns:getNotificationSettingsResponse"/>  <xs:element name="getNotificationTypes" type="tns:getNotificationTypes"/>  <xs:element name="getNotificationTypesResponse" type="tns:getNotificationTypesResponse"/>  <xs:element name="getPaymentList" type="tns:getPaymentList"/>  <xs:element name="getPaymentListBySan" type="tns:getPaymentListBySan"/>  <xs:element name="getPaymentListBySanResponse" type="tns:getPaymentListBySanResponse"/>  <xs:element name="getPaymentListResponse" type="tns:getPaymentListResponse"/>  <xs:element name="getPurchaseList" type="tns:getPurchaseList"/>  <xs:element name="getPurchaseListBySan" type="tns:getPurchaseListBySan"/>  <xs:element name="getPurchaseListBySanResponse" type="tns:getPurchaseListBySanResponse"/>  <xs:element name="getPurchaseListResponse" type="tns:getPurchaseListResponse"/>  <xs:element name="getRBKMoneyConfig" type="tns:getRBKMoneyConfig"/>  <xs:element name="getRBKMoneyConfigResponse" type="tns:getRBKMoneyConfigResponse"/>  <xs:element name="getStudentListByIdOfClientGroup" type="tns:getStudentListByIdOfClientGroup"/>  <xs:element name="getStudentListByIdOfClientGroupResponse" type="tns:getStudentListByIdOfClientGroupResponse"/>  <xs:element name="getStudentsByCanNotConfirmPayment" type="tns:getStudentsByCanNotConfirmPayment"/>  <xs:element name="getStudentsByCanNotConfirmPaymentResponse" type="tns:getStudentsByCanNotConfirmPaymentResponse"/>  <xs:element name="getSummary" type="tns:getSummary"/>  <xs:element name="getSummaryByGuardSan" type="tns:getSummaryByGuardSan"/>  <xs:element name="getSummaryByGuardSanResponse" type="tns:getSummaryByGuardSanResponse"/>  <xs:element name="getSummaryBySan" type="tns:getSummaryBySan"/>  <xs:element name="getSummaryBySanResponse" type="tns:getSummaryBySanResponse"/>  <xs:element name="getSummaryByTypedId" type="tns:getSummaryByTypedId"/>  <xs:element name="getSummaryByTypedIdResponse" type="tns:getSummaryByTypedIdResponse"/>  <xs:element name="getSummaryResponse" type="tns:getSummaryResponse"/>  <xs:element name="giveConclusionOnComplaint" type="tns:giveConclusionOnComplaint"/>  <xs:element name="giveConclusionOnComplaintResponse" type="tns:giveConclusionOnComplaintResponse"/>  <xs:element name="openComplaint" type="tns:openComplaint"/>  <xs:element name="openComplaintResponse" type="tns:openComplaintResponse"/>  <xs:element name="sendLinkingTokenByContractId" type="tns:sendLinkingTokenByContractId"/>  <xs:element name="sendLinkingTokenByContractIdResponse" type="tns:sendLinkingTokenByContractIdResponse"/>  <xs:element name="sendLinkingTokenByMobile" type="tns:sendLinkingTokenByMobile"/>  <xs:element name="sendLinkingTokenByMobileResponse" type="tns:sendLinkingTokenByMobileResponse"/>  <xs:element name="sendPasswordRecoverURLFromEmail" type="tns:sendPasswordRecoverURLFromEmail"/>  <xs:element name="sendPasswordRecoverURLFromEmailResponse" type="tns:sendPasswordRecoverURLFromEmailResponse"/>  <xs:element name="setAnswerFromQuestion" type="tns:setAnswerFromQuestion"/>  <xs:element name="setAnswerFromQuestionResponse" type="tns:setAnswerFromQuestionResponse"/>  <xs:element name="setNotificationSettings" type="tns:setNotificationSettings"/>  <xs:element name="setNotificationSettingsResponse" type="tns:setNotificationSettingsResponse"/>  <xs:element name="setProhibitionOnGood" type="tns:setProhibitionOnGood"/>  <xs:element name="setProhibitionOnGoodGroup" type="tns:setProhibitionOnGoodGroup"/>  <xs:element name="setProhibitionOnGoodGroupResponse" type="tns:setProhibitionOnGoodGroupResponse"/>  <xs:element name="setProhibitionOnGoodResponse" type="tns:setProhibitionOnGoodResponse"/>  <xs:element name="setProhibitionOnProduct" type="tns:setProhibitionOnProduct"/>  <xs:element name="setProhibitionOnProductGroup" type="tns:setProhibitionOnProductGroup"/>  <xs:element name="setProhibitionOnProductGroupResponse" type="tns:setProhibitionOnProductGroupResponse"/>  <xs:element name="setProhibitionOnProductResponse" type="tns:setProhibitionOnProductResponse"/>  <xs:element name="transferBalance" type="tns: transferBalance"/>  <xs:element name="transferBalanceResponse" type="tns: transferBalanceResponse"/>  <xs:complexType name="changeComplaintStatusToInvestigation">  <xs:sequence>  <xs:element minOccurs="0" name="complaintId" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="changeComplaintStatusToInvestigationResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:result"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="result">  <xs:sequence>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="activateLinkingToken">  <xs:sequence>  <xs:element minOccurs="0" name="linkingToken" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="activateLinkingTokenResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:activateLinkingTokenResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="activateLinkingTokenResult">  <xs:complexContent>  <xs:extension base="tns:result">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  </xs:sequence>  </xs:extension>  </xs:complexContent>  </xs:complexType>  <xs:complexType name="setNotificationSettings">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element maxOccurs="unbounded" minOccurs="0" name="notificationType" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="setNotificationSettingsResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:clientNotificationChangeResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="clientNotificationChangeResult">  <xs:sequence>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="excludeGoodFromProhibition">  <xs:sequence>  <xs:element minOccurs="0" name="orgId" type="xs:long"/>  <xs:element minOccurs="0" name="idOfProhibition" type="xs:long"/>  <xs:element minOccurs="0" name="idOfGood" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="excludeGoodFromProhibitionResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:idResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="idResult">  <xs:sequence>  <xs:element minOccurs="0" name="id" type="xs:long"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="sendLinkingTokenByContractId">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="sendLinkingTokenByContractIdResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:result"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="createPaymentOrder">  <xs:sequence>  <xs:element minOccurs="0" name="idOfClient" type="xs:long"/>  <xs:element minOccurs="0" name="idOfContragent" type="xs:long"/>  <xs:element name="paymentMethod" type="xs:int"/>  <xs:element minOccurs="0" name="copecksAmount" type="xs:long"/>  <xs:element minOccurs="0" name="contragentSum" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="createPaymentOrderResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:idResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getListOfComplaintBookEntriesByClient">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getListOfComplaintBookEntriesByClientResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:listOfComplaintBookEntriesResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="listOfComplaintBookEntriesResult">  <xs:sequence>  <xs:element minOccurs="0" name="listOfComplaintBookEntries" type="tns:ListOfComplaintBookEntries"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ListOfComplaintBookEntries">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="E" type="tns:ListOfComplaintBookEntriesExt"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ListOfComplaintBookEntriesExt">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="Iterations" type="tns:ListOfComplaintIterations"/>  </xs:sequence>  <xs:attribute name="ContractId" type="xs:long"/>  <xs:attribute name="GuidOfGood" type="xs:string"/>  <xs:attribute name="NameOfGood" type="xs:string"/>  <xs:attribute name="Guid" type="xs:string"/>  <xs:attribute name="DeletedState" type="xs:boolean"/>  <xs:attribute name="CreatedDate" type="xs:date"/>  <xs:attribute name="OrgOwner" type="xs:long"/>  </xs:complexType>  <xs:complexType name="ListOfComplaintIterations">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="I" type="tns:ListOfComplaintIterationsExt"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ListOfComplaintIterationsExt">  <xs:sequence>  <xs:element minOccurs="0" name="ProblemDescription" type="xs:string"/>  <xs:element maxOccurs="unbounded" minOccurs="0" name="Orders" type="tns:ListOfComplaintOrders"/>  <xs:element maxOccurs="unbounded" minOccurs="0" name="Causes" type="tns:ListOfComplaintCauses"/>  </xs:sequence>  <xs:attribute name="IterationNumber" type="xs:int"/>  <xs:attribute name="GoodComplaintIterationStatus" type="xs:int"/>  <xs:attribute name="Conclusion" type="xs:string"/>  <xs:attribute name="Guid" type="xs:string"/>  <xs:attribute name="DeletedState" type="xs:boolean"/>  <xs:attribute name="CreatedDate" type="xs:date"/>  <xs:attribute name="OrgOwner" type="xs:long"/>  </xs:complexType>  <xs:complexType name="ListOfComplaintOrders">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="O" type="tns:ListOfComplaintOrdersExt"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ListOfComplaintOrdersExt">  <xs:sequence/>  <xs:attribute name="IdOfOrderDetail" type="xs:long"/>  <xs:attribute name="MenuDetailName" type="xs:string"/>  <xs:attribute name="DateOfOrder" type="xs:anySimpleType"/>  <xs:attribute name="Guid" type="xs:string"/>  <xs:attribute name="DeletedState" type="xs:boolean"/>  <xs:attribute name="CreatedDate" type="xs:date"/>  <xs:attribute name="OrgOwner" type="xs:long"/>  </xs:complexType>  <xs:complexType name="ListOfComplaintCauses">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="C" type="tns:ListOfComplaintCausesExt"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ListOfComplaintCausesExt">  <xs:sequence/>  <xs:attribute name="Cause" type="xs:int"/>  <xs:attribute name="CauseDescription" type="xs:string"/>  <xs:attribute name="Guid" type="xs:string"/>  <xs:attribute name="DeletedState" type="xs:boolean"/>  <xs:attribute name="CreatedDate" type="xs:date"/>  <xs:attribute name="OrgOwner" type="xs:long"/>  </xs:complexType>  <xs:complexType name="getListOfComplaintBookEntriesByOrg">  <xs:sequence>  <xs:element minOccurs="0" name="orgId" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getListOfComplaintBookEntriesByOrgResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:listOfComplaintBookEntriesResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="transferBalance">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="fromSub" type="xs:int"/>  <xs:element minOccurs="0" name="toSub" type="xs:int"/>  <xs:element minOccurs="0" name="amount" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name=" transferBalanceResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:result"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getChronopayConfig">  <xs:sequence/>  </xs:complexType>  <xs:complexType name="getChronopayConfigResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:chronopayConfigResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="chronopayConfigResult">  <xs:sequence>  <xs:element minOccurs="0" name="chronopayConfig" type="tns:ChronopayConfigExt"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ChronopayConfigExt">  <xs:sequence/>  <xs:attribute name="SharedSec" type="xs:string"/>  <xs:attribute name="Rate" type="xs:double"/>  <xs:attribute name="IP" type="xs:string"/>  <xs:attribute name="ContragentName" type="xs:string"/>  <xs:attribute name="PurchaseUri" type="xs:string"/>  <xs:attribute name="CallbackUrl" type="xs:string"/>  <xs:attribute name="Show" type="xs:boolean"/>  </xs:complexType>  <xs:complexType name="getClientStats">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="startDate" type="xs:dateTime"/>  <xs:element minOccurs="0" name="endDate" type="xs:dateTime"/>  <xs:element name="type" type="xs:int"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getClientStatsResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:clientStatsResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="clientStatsResult">  <xs:sequence>  <xs:element name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  <xs:element minOccurs="0" name="stats" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="changeMobilePhone">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="mobilePhone" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="changeMobilePhoneResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:result"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getStudentsByCanNotConfirmPayment">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getStudentsByCanNotConfirmPaymentResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:studentsConfirmPaymentData"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="studentsConfirmPaymentData">  <xs:sequence>  <xs:element minOccurs="0" name="studentsConfirmPaymentList" type="tns:studentsConfirmPaymentList"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="studentsConfirmPaymentList">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="StudentMustPayItems" type="tns:StudentMustPayItem"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="StudentMustPayItem">  <xs:sequence/>  <xs:attribute name="FirstName" type="xs:string"/>  <xs:attribute name="Surname" type="xs:string"/>  <xs:attribute name="SecondName" type="xs:string"/>  <xs:attribute name="Balance" type="xs:long"/>  <xs:attribute name="CreateTime" type="xs:dateTime"/>  <xs:attribute name="PaySum" type="xs:long"/>  </xs:complexType>  <xs:complexType name="generateLinkingToken">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="generateLinkingTokenResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:generateLinkingTokenResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="generateLinkingTokenResult">  <xs:complexContent>  <xs:extension base="tns:result">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="linkingToken" type="xs:string"/>  </xs:sequence>  </xs:extension>  </xs:complexContent>  </xs:complexType>  <xs:complexType name="checkPasswordRestoreRequest">  <xs:sequence>  <xs:element minOccurs="0" name="request" type="tns:requestWebParam"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="requestWebParam">  <xs:sequence>  <xs:element minOccurs="0" name="url" type="xs:string"/>  <xs:element minOccurs="0" name="pParam" type="xs:string"/>  <xs:element minOccurs="0" name="dateParam" type="xs:string"/>  <xs:element minOccurs="0" name="contractIdParam" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="checkPasswordRestoreRequestResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:checkPasswordResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="checkPasswordResult">  <xs:sequence>  <xs:element name="succeeded" type="xs:boolean"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getListOfProducts">  <xs:sequence>  <xs:element minOccurs="0" name="orgId" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getListOfProductsResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:listOfProductsResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="listOfProductsResult">  <xs:sequence>  <xs:element minOccurs="0" name="listOfProductGroups" type="tns:ListOfProductGroups"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ListOfProductGroups">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="PG" type="tns:ListOfProductGroupsExt"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ListOfProductGroupsExt">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="Products" type="tns:ListOfProducts"/>  </xs:sequence>  <xs:attribute name="NameOfGroup" type="xs:string"/>  <xs:attribute name="ClassificationCode" type="xs:string"/>  <xs:attribute name="DeletedState" type="xs:boolean"/>  <xs:attribute name="Guid" type="xs:string"/>  <xs:attribute name="OrgOwner" type="xs:long"/>  <xs:attribute name="CreatedDate" type="xs:date"/>  </xs:complexType>  <xs:complexType name="ListOfProducts">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="P" type="tns:ListOfProductsExt"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ListOfProductsExt">  <xs:sequence/>  <xs:attribute name="Code" type="xs:string"/>  <xs:attribute name="OKPCode" type="xs:string"/>  <xs:attribute name="ClassificationCode" type="xs:string"/>  <xs:attribute name="DeletedState" type="xs:boolean"/>  <xs:attribute name="Guid" type="xs:string"/>  <xs:attribute name="ProductName" type="xs:string"/>  <xs:attribute name="FullName" type="xs:string"/>  <xs:attribute name="Density" type="xs:float"/>  <xs:attribute name="OrgOwner" type="xs:long"/>  <xs:attribute name="CreatedDate" type="xs:date"/>  </xs:complexType>  <xs:complexType name="sendLinkingTokenByMobile">  <xs:sequence>  <xs:element minOccurs="0" name="mobilePhone" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="sendLinkingTokenByMobileResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:result"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getSummary">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getSummaryResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:clientSummaryResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="clientSummaryResult">  <xs:sequence>  <xs:element minOccurs="0" name="clientSummary" type="tns:ClientSummaryExt"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ClientSummaryExt">  <xs:sequence/>  <xs:attribute name="ContractId" type="xs:long"/>  <xs:attribute name="DateOfContract" type="xs:date"/>  <xs:attribute name="StateOfContract" type="xs:string"/>  <xs:attribute name="Balance" type="xs:long"/>  <xs:attribute name="OverdraftLimit" type="xs:long"/>  <xs:attribute name="ExpenditureLimit" type="xs:long"/>  <xs:attribute name="FirstName" type="xs:string"/>  <xs:attribute name="LastName" type="xs:string"/>  <xs:attribute name="MiddleName" type="xs:string"/>  <xs:attribute name="Grade" type="xs:string"/>  <xs:attribute name="OfficialName" type="xs:string"/>  <xs:attribute name="LastEnterEventCode" type="xs:int"/>  <xs:attribute name="LastEnterEventTime" type="xs:dateTime"/>  <xs:attribute name="NotifyViaSMS" type="xs:boolean"/>  <xs:attribute name="NotifyViaEmail" type="xs:boolean"/>  <xs:attribute name="MobilePhone" type="xs:string"/>  <xs:attribute name="Email" type="xs:string"/>  <xs:attribute name="DefaultMerchantId" type="xs:long"/>  <xs:attribute name="DefaultMerchantInfo" type="xs:string"/>  <xs:attribute name="Address" type="xs:string"/>  <xs:attribute name="Phone" type="xs:string"/>  <xs:attribute name="FreePayMaxCount" type="xs:int"/>  <xs:attribute name="FreePayCount" type="xs:int"/>  <xs:attribute name="LastFreePayTime" type="xs:date"/>  <xs:attribute name="DiscountMode" type="xs:int"/>  <xs:attribute name="Limit" type="xs:long"/>  </xs:complexType>  <xs:complexType name="changePaymentOrderStatus">  <xs:sequence>  <xs:element minOccurs="0" name="idOfClient" type="xs:long"/>  <xs:element minOccurs="0" name="idOfClientPaymentOrder" type="xs:long"/>  <xs:element name="orderStatus" type="xs:int"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="changePaymentOrderStatusResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:result"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="giveConclusionOnComplaint">  <xs:sequence>  <xs:element minOccurs="0" name="complaintId" type="xs:long"/>  <xs:element minOccurs="0" name="conclusion" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="giveConclusionOnComplaintResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:result"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getHiddenPages">  <xs:sequence/>  </xs:complexType>  <xs:complexType name="getHiddenPagesResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:hiddenPagesResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="hiddenPagesResult">  <xs:sequence>  <xs:element minOccurs="0" name="hiddenPages" type="xs:string"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getNotificationTypes">  <xs:sequence/>  </xs:complexType>  <xs:complexType name="getNotificationTypesResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:clientNotificationSettingsResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="clientNotificationSettingsResult">  <xs:sequence>  <xs:element minOccurs="0" name="description" type="xs:string"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element maxOccurs="unbounded" minOccurs="0" name="settings" nillable="true" type="tns:clientNotificationSettingsItem"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="clientNotificationSettingsItem">  <xs:sequence>  <xs:element minOccurs="0" name="nameOfNotification" type="xs:string"/>  <xs:element minOccurs="0" name="typeOfNotification" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCardListBySan">  <xs:sequence>  <xs:element minOccurs="0" name="san" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCardListBySanResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:cardListResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="cardListResult">  <xs:sequence>  <xs:element minOccurs="0" name="cardList" type="tns:CardList"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="CardList">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="C" type="tns:CardItem"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="CardItem">  <xs:sequence/>  <xs:attribute name="IdOfCard" type="xs:long"/>  <xs:attribute name="CrystalId" type="xs:long"/>  <xs:attribute name="Type" type="xs:int"/>  <xs:attribute name="State" type="xs:int"/>  <xs:attribute name="LifeState" type="xs:int"/>  <xs:attribute name="ExpiryDate" type="xs:date"/>  <xs:attribute name="ChangeDate" type="xs:dateTime"/>  </xs:complexType>  <xs:complexType name="getPaymentList">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="startDate" type="xs:dateTime"/>  <xs:element minOccurs="0" name="endDate" type="xs:dateTime"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getPaymentListResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:paymentListResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="paymentListResult">  <xs:sequence>  <xs:element minOccurs="0" name="paymentList" type="tns:PaymentList"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="PaymentList">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="P" type="tns:Payment"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="Payment">  <xs:sequence/>  <xs:attribute name="Time" type="xs:dateTime"/>  <xs:attribute name="Sum" type="xs:long"/>  <xs:attribute name="Origin" type="xs:string"/>  </xs:complexType>  <xs:complexType name="getDishProhibitionsList">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getDishProhibitionsListResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:prohibitionsListResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="prohibitionsListResult">  <xs:sequence>  <xs:element minOccurs="0" name="prohibitionsList" type="tns:ProhibitionsList"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ProhibitionsList">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="P" type="tns:ProhibitionsListExt"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ProhibitionsListExt">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="Exclusions" type="tns:ProhibitionExclusionsList"/>  </xs:sequence>  <xs:attribute name="Guid" type="xs:string"/>  <xs:attribute name="DeletedState" type="xs:boolean"/>  <xs:attribute name="CreatedDate" type="xs:date"/>  <xs:attribute name="ContractId" type="xs:long"/>  <xs:attribute name="GuidOfProducts" type="xs:string"/>  <xs:attribute name="GuidOfProductGroups" type="xs:string"/>  <xs:attribute name="GuidOfGood" type="xs:string"/>  <xs:attribute name="GuidOfGoodsGroup" type="xs:string"/>  </xs:complexType>  <xs:complexType name="ProhibitionExclusionsList">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="E" type="tns:ProhibitionExclusionsListExt"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ProhibitionExclusionsListExt">  <xs:sequence/>  <xs:attribute name="Guid" type="xs:string"/>  <xs:attribute name="DeletedState" type="xs:boolean"/>  <xs:attribute name="CreatedDate" type="xs:date"/>  <xs:attribute name="GuidOfGood" type="xs:string"/>  <xs:attribute name="GuidOfGoodsGroup" type="xs:string"/>  </xs:complexType>  <xs:complexType name="getCardList">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCardListResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:cardListResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="detachGuardSan">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="guardSan" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="detachGuardSanResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:detachGuardSanResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="detachGuardSanResult">  <xs:complexContent>  <xs:extension base="tns:result">  <xs:sequence/>  </xs:extension>  </xs:complexContent>  </xs:complexType>  <xs:complexType name="enableNotificationBySMS">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element name="state" type="xs:boolean"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="enableNotificationBySMSResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:result"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getClientSmsList">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="startDate" type="xs:dateTime"/>  <xs:element minOccurs="0" name="endDate" type="xs:dateTime"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getClientSmsListResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:clientSmsListResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="clientSmsListResult">  <xs:sequence>  <xs:element minOccurs="0" name="clientSmsList" type="tns:ClientSmsList"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ClientSmsList">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="S" type="tns:Sms"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="Sms">  <xs:sequence/>  <xs:attribute name="ServiceSendTime" type="xs:dateTime"/>  <xs:attribute name="Price" type="xs:long"/>  <xs:attribute name="TransactionSum" type="xs:long"/>  <xs:attribute name="CardNo" type="xs:long"/>  <xs:attribute name="DeliveryStatus" type="xs:int"/>  <xs:attribute name="ContentsType" type="xs:int"/>  </xs:complexType>  <xs:complexType name="setProhibitionOnGoodGroup">  <xs:sequence>  <xs:element minOccurs="0" name="orgId" type="xs:long"/>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="idOfGoodGroup" type="xs:long"/>  <xs:element minOccurs="0" name="isDeleted" type="xs:boolean"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="setProhibitionOnGoodGroupResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:idResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getListOfGoods">  <xs:sequence>  <xs:element minOccurs="0" name="orgId" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getListOfGoodsResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:listOfGoodsResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="listOfGoodsResult">  <xs:sequence>  <xs:element minOccurs="0" name="listOfGoodGroups" type="tns:ListOfGoodGroups"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ListOfGoodGroups">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="GG" type="tns:ListOfGoodGroupsExt"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ListOfGoodGroupsExt">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="Goods" type="tns:ListOfGoods"/>  </xs:sequence>  <xs:attribute name="NameOfGoodsGroup" type="xs:string"/>  <xs:attribute name="Guid" type="xs:string"/>  <xs:attribute name="DeletedState" type="xs:boolean"/>  <xs:attribute name="OrgOwner" type="xs:long"/>  <xs:attribute name="CreatedDate" type="xs:date"/>  </xs:complexType>  <xs:complexType name="ListOfGoods">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="G" type="tns:ListOfGoodsExt"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ListOfGoodsExt">  <xs:sequence/>  <xs:attribute name="GoodsCode" type="xs:string"/>  <xs:attribute name="Guid" type="xs:string"/>  <xs:attribute name="DeletedState" type="xs:boolean"/>  <xs:attribute name="OrgOwner" type="xs:long"/>  <xs:attribute name="NameOfGood" type="xs:string"/>  <xs:attribute name="FullName" type="xs:string"/>  <xs:attribute name="UnitsScale" type="xs:int"/>  <xs:attribute name="NetWeight" type="xs:long"/>  <xs:attribute name="Lifetime" type="xs:long"/>  <xs:attribute name="Margin" type="xs:long"/>  <xs:attribute name="CreatedDate" type="xs:date"/>  </xs:complexType>  <xs:complexType name="getActiveMenuQuestions">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="currentDate" type="xs:dateTime"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getActiveMenuQuestionsResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:questionaryResultList"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="questionaryResultList">  <xs:sequence>  <xs:element minOccurs="0" name="questionaryList" type="tns:QuestionaryList"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="QuestionaryList">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="Q" type="tns:QuestionaryItem"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="QuestionaryItem">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="Answer" type="tns:AnswerItem"/>  </xs:sequence>  <xs:attribute name="IdOfQuestionary" type="xs:long"/>  <xs:attribute name="Question" type="xs:string" use="required"/>  <xs:attribute name="Name" type="xs:string" use="required"/>  <xs:attribute name="Description" type="xs:string"/>  <xs:attribute name="Status" type="xs:int"/>  <xs:attribute name="Type" type="xs:int"/>  <xs:attribute name="ViewDate" type="xs:dateTime"/>  <xs:attribute name="CheckedAnswer" type="xs:long"/>  </xs:complexType>  <xs:complexType name="AnswerItem">  <xs:sequence/>  <xs:attribute name="id" type="xs:long"/>  <xs:attribute name="answer" type="xs:string" use="required"/>  <xs:attribute name="weight" type="xs:int"/>  <xs:attribute name="description" type="xs:string"/>  </xs:complexType>  <xs:complexType name="changeEmail">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="email" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="changeEmailResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:result"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="excludeGoodGroupFromProhibition">  <xs:sequence>  <xs:element minOccurs="0" name="orgId" type="xs:long"/>  <xs:element minOccurs="0" name="idOfProhibition" type="xs:long"/>  <xs:element minOccurs="0" name="idOfGoodGroup" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="excludeGoodGroupFromProhibitionResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:idResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="enableNotificationByEmail">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element name="state" type="xs:boolean"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="enableNotificationByEmailResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:result"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="attachGuardSan">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="guardSan" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="attachGuardSanResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:attachGuardSanResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="attachGuardSanResult">  <xs:complexContent>  <xs:extension base="tns:result">  <xs:sequence/>  </xs:extension>  </xs:complexContent>  </xs:complexType>  <xs:complexType name="setAnswerFromQuestion">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="IdOfAnswer" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="setAnswerFromQuestionResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:result"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="changePersonalInfo">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="limit" type="xs:long"/>  <xs:element minOccurs="0" name="address" type="xs:string"/>  <xs:element minOccurs="0" name="phone" type="xs:string"/>  <xs:element minOccurs="0" name="mobilePhone" type="xs:string"/>  <xs:element minOccurs="0" name="email" type="xs:string"/>  <xs:element name="smsNotificationState" type="xs:boolean"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="changePersonalInfoResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:result"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getStudentListByIdOfClientGroup">  <xs:sequence>  <xs:element minOccurs="0" name="idOfClientGroup" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getStudentListByIdOfClientGroupResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:classStudentListResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="classStudentListResult">  <xs:sequence>  <xs:element minOccurs="0" name="classStudentList" type="tns:ClassStudentList"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ClassStudentList">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="C" type="tns:ClientSummaryExt"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getPaymentListBySan">  <xs:sequence>  <xs:element minOccurs="0" name="san" type="xs:string"/>  <xs:element minOccurs="0" name="startDate" type="xs:dateTime"/>  <xs:element minOccurs="0" name="endDate" type="xs:dateTime"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getPaymentListBySanResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:paymentListResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getClientsByGuardSan">  <xs:sequence>  <xs:element minOccurs="0" name="san" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getClientsByGuardSanResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:clientsData"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="clientsData">  <xs:sequence>  <xs:element minOccurs="0" name="clientList" type="tns:clientList"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="clientList">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="Client" type="tns:ClientItem"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ClientItem">  <xs:sequence/>  <xs:attribute name="ContractId" type="xs:long"/>  <xs:attribute name="San" type="xs:string"/>  </xs:complexType>  <xs:complexType name="setProhibitionOnProduct">  <xs:sequence>  <xs:element minOccurs="0" name="orgId" type="xs:long"/>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="idOfProduct" type="xs:long"/>  <xs:element minOccurs="0" name="isDeleted" type="xs:boolean"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="setProhibitionOnProductResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:idResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getGroupListByOrg">  <xs:sequence>  <xs:element minOccurs="0" name="idOfOrg" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getGroupListByOrgResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:clientGroupListResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="clientGroupListResult">  <xs:sequence>  <xs:element minOccurs="0" name="clientGroupList" type="tns:ClientGroupList"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ClientGroupList">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="G" type="tns:ClientGroupItem"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ClientGroupItem">  <xs:sequence/>  <xs:attribute name="idOfClientGroup" type="xs:long"/>  <xs:attribute name="GroupName" type="xs:string"/>  </xs:complexType>  <xs:complexType name="getSummaryBySan">  <xs:sequence>  <xs:element minOccurs="0" name="san" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getSummaryBySanResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:clientSummaryResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getEnterEventList">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="startDate" type="xs:dateTime"/>  <xs:element minOccurs="0" name="endDate" type="xs:dateTime"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getEnterEventListResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:enterEventListResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="enterEventListResult">  <xs:sequence>  <xs:element minOccurs="0" name="enterEventList" type="tns:EnterEventList"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="EnterEventList">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="E" type="tns:EnterEventItem"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="EnterEventItem">  <xs:sequence/>  <xs:attribute name="DateTime" type="xs:dateTime"/>  <xs:attribute name="Day" type="xs:int"/>  <xs:attribute name="EnterName" type="xs:string"/>  <xs:attribute name="Direction" type="xs:int"/>  <xs:attribute name="TemporaryCard" type="xs:int"/>  </xs:complexType>  <xs:complexType name="getBanks">  <xs:sequence/>  </xs:complexType>  <xs:complexType name="getBanksResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:banksData"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="banksData">  <xs:sequence>  <xs:element minOccurs="0" name="banksList" type="tns:banksList"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="banksList">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="Banks" type="tns:BankItem"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="BankItem">  <xs:sequence>  <xs:element minOccurs="0" name="idOfBank" type="xs:long"/>  </xs:sequence>  <xs:attribute name="Name" type="xs:string"/>  <xs:attribute name="LogoUrl" type="xs:string"/>  <xs:attribute name="TerminalsUrl" type="xs:string"/>  <xs:attribute name="Rate" type="xs:double"/>  <xs:attribute name="MinRate" type="xs:double"/>  <xs:attribute name="EnrollmentType" type="xs:string"/>  </xs:complexType>  <xs:complexType name="setProhibitionOnGood">  <xs:sequence>  <xs:element minOccurs="0" name="orgId" type="xs:long"/>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="idOfGood" type="xs:long"/>  <xs:element minOccurs="0" name="isDeleted" type="xs:boolean"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="setProhibitionOnGoodResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:idResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="setProhibitionOnProductGroup">  <xs:sequence>  <xs:element minOccurs="0" name="orgId" type="xs:long"/>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="idOfProductGroup" type="xs:long"/>  <xs:element minOccurs="0" name="isDeleted" type="xs:boolean"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="setProhibitionOnProductGroupResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:idResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getContractIdByCardNo">  <xs:sequence>  <xs:element minOccurs="0" name="cardId" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getContractIdByCardNoResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getMenuListBySan">  <xs:sequence>  <xs:element minOccurs="0" name="san" type="xs:string"/>  <xs:element minOccurs="0" name="startDate" type="xs:dateTime"/>  <xs:element minOccurs="0" name="endDate" type="xs:dateTime"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getMenuListBySanResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:menuListResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="menuListResult">  <xs:sequence>  <xs:element minOccurs="0" name="menuList" type="tns:MenuListExt"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="MenuListExt">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="M" type="tns:MenuDateItemExt"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="MenuDateItemExt">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="E" type="tns:MenuItemExt"/>  </xs:sequence>  <xs:attribute name="Date" type="xs:date"/>  </xs:complexType>  <xs:complexType name="MenuItemExt">  <xs:sequence/>  <xs:attribute name="Group" type="xs:string"/>  <xs:attribute name="Name" type="xs:string"/>  <xs:attribute name="Price" type="xs:long"/>  <xs:attribute name="Calories" type="xs:double"/>  <xs:attribute name="VitB1" type="xs:double"/>  <xs:attribute name="VitC" type="xs:double"/>  <xs:attribute name="VitA" type="xs:double"/>  <xs:attribute name="VitE" type="xs:double"/>  <xs:attribute name="MinCa" type="xs:double"/>  <xs:attribute name="MinP" type="xs:double"/>  <xs:attribute name="MinMg" type="xs:double"/>  <xs:attribute name="MinFe" type="xs:double"/>  </xs:complexType>  <xs:complexType name="getSummaryByTypedId">  <xs:sequence>  <xs:element minOccurs="0" name="id" type="xs:string"/>  <xs:element name="idType" type="xs:int"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getSummaryByTypedIdResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:clientSummaryResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getSummaryByGuardSan">  <xs:sequence>  <xs:element minOccurs="0" name="guardSan" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getSummaryByGuardSanResponse">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="return" type="tns:ClientSummaryExt"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getPurchaseListBySan">  <xs:sequence>  <xs:element minOccurs="0" name="san" type="xs:string"/>  <xs:element minOccurs="0" name="startDate" type="xs:dateTime"/>  <xs:element minOccurs="0" name="endDate" type="xs:dateTime"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getPurchaseListBySanResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:purchaseListResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="purchaseListResult">  <xs:sequence>  <xs:element minOccurs="0" name="purchaseList" type="tns:PurchaseListExt"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="PurchaseListExt">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="P" type="tns:PurchaseExt"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="PurchaseExt">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="E" type="tns:PurchaseElementExt"/>  </xs:sequence>  <xs:attribute name="Time" type="xs:dateTime"/>  <xs:attribute name="Sum" type="xs:long"/>  <xs:attribute name="SocDiscount" type="xs:long"/>  <xs:attribute name="TrdDiscount" type="xs:long"/>  <xs:attribute name="Donation" type="xs:long"/>  <xs:attribute name="ByCash" type="xs:long"/>  <xs:attribute name="ByCard" type="xs:long"/>  <xs:attribute name="IdOfCard" type="xs:long"/>  </xs:complexType>  <xs:complexType name="PurchaseElementExt">  <xs:sequence/>  <xs:attribute name="IdOfOrderDetail" type="xs:long"/>  <xs:attribute name="Name" type="xs:string"/>  <xs:attribute name="Sum" type="xs:long"/>  <xs:attribute name="Amount" type="xs:long"/>  <xs:attribute name="Type" type="xs:int"/>  </xs:complexType>  <xs:complexType name="getMenuList">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="startDate" type="xs:dateTime"/>  <xs:element minOccurs="0" name="endDate" type="xs:dateTime"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getMenuListResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:menuListResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getIdOfContragent">  <xs:sequence>  <xs:element minOccurs="0" name="contragentName" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getIdOfContragentResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:idResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="openComplaint">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="orderOrgId" type="xs:long"/>  <xs:element maxOccurs="unbounded" minOccurs="0" name="idOfOrderDetail" type="xs:long"/>  <xs:element maxOccurs="unbounded" minOccurs="0" name="causeNumber" type="xs:int"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="openComplaintResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:idResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getRBKMoneyConfig">  <xs:sequence/>  </xs:complexType>  <xs:complexType name="getRBKMoneyConfigResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:rbkMoneyConfigResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="rbkMoneyConfigResult">  <xs:sequence>  <xs:element minOccurs="0" name="rbkConfig" type="tns:RBKMoneyConfigExt"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="RBKMoneyConfigExt">  <xs:sequence/>  <xs:attribute name="EshopId" type="xs:string"/>  <xs:attribute name="ServiceName" type="xs:string"/>  <xs:attribute name="ContragentName" type="xs:string"/>  <xs:attribute name="PurchaseUri" type="xs:string"/>  <xs:attribute name="SecretKey" type="xs:string"/>  <xs:attribute name="Rate" type="xs:double"/>  <xs:attribute name="Show" type="xs:boolean"/>  </xs:complexType>  <xs:complexType name="authorizeClient">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="token" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="authorizeClientResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:result"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="sendPasswordRecoverURLFromEmail">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="request" type="tns:requestWebParam"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="sendPasswordRecoverURLFromEmailResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:sendResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="sendResult">  <xs:sequence>  <xs:element name="recoverStatus" type="xs:int"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getNotificationSettings">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getNotificationSettingsResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:clientNotificationSettingsResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCirculationList">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element name="state" type="xs:int"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getCirculationListResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:circulationListResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="circulationListResult">  <xs:sequence>  <xs:element minOccurs="0" name="circulationList" type="tns:CirculationItemList"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="CirculationItemList">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="C" type="tns:CirculationItem"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="CirculationItem">  <xs:sequence>  <xs:element name="Publication" type="tns:PublicationItem"/>  </xs:sequence>  <xs:attribute name="IssuanceDate" type="xs:date"/>  <xs:attribute name="RefundDate" type="xs:date"/>  <xs:attribute name="RealRefundDate" type="xs:date"/>  <xs:attribute name="Status" type="xs:int"/>  </xs:complexType>  <xs:complexType name="PublicationItem">  <xs:sequence/>  <xs:attribute name="Author" type="xs:string"/>  <xs:attribute name="Title" type="xs:string"/>  <xs:attribute name="Title2" type="xs:string"/>  <xs:attribute name="PublicationDate" type="xs:string"/>  <xs:attribute name="Publisher" type="xs:string"/>  </xs:complexType>  <xs:complexType name="changeComplaintStatusToConsideration">  <xs:sequence>  <xs:element minOccurs="0" name="complaintId" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="changeComplaintStatusToConsiderationResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:result"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="changePassword">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="base64passwordHash" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="changePasswordResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:result"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getPurchaseList">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="startDate" type="xs:dateTime"/>  <xs:element minOccurs="0" name="endDate" type="xs:dateTime"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getPurchaseListResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:purchaseListResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="changeExpenditureLimit">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element name="limit" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="changeExpenditureLimitResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:result"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getIdOfClient">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getIdOfClientResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:idResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getMenuListByOrg">  <xs:sequence>  <xs:element minOccurs="0" name="orgId" type="xs:long"/>  <xs:element minOccurs="0" name="startDate" type="xs:dateTime"/>  <xs:element minOccurs="0" name="endDate" type="xs:dateTime"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getMenuListByOrgResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:menuListResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getComplexList">  <xs:sequence>  <xs:element minOccurs="0" name="contractId" type="xs:long"/>  <xs:element minOccurs="0" name="arg1" type="xs:dateTime"/>  <xs:element minOccurs="0" name="arg2" type="xs:dateTime"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getComplexListResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:complexListResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="complexListResult">  <xs:sequence>  <xs:element minOccurs="0" name="complexDateList" type="tns:ComplexDateList"/>  <xs:element minOccurs="0" name="resultCode" type="xs:long"/>  <xs:element minOccurs="0" name="description" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ComplexDateList">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="E" type="tns:ComplexDate"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ComplexDate">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="E" type="tns:Complex"/>  </xs:sequence>  <xs:attribute name="Date" type="xs:date"/>  </xs:complexType>  <xs:complexType name="Complex">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="E" type="tns:ComplexDetail"/>  </xs:sequence>  <xs:attribute name="Name" type="xs:string"/>  </xs:complexType>  <xs:complexType name="ComplexDetail">  <xs:sequence/>  <xs:attribute name="Name" type="xs:string"/>  </xs:complexType>  <xs:complexType name="attachGuardSanBySan">  <xs:sequence>  <xs:element minOccurs="0" name="san" type="xs:string"/>  <xs:element minOccurs="0" name="guardSan" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="attachGuardSanBySanResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:attachGuardSanResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="detachGuardSanBySan">  <xs:sequence>  <xs:element minOccurs="0" name="san" type="xs:string"/>  <xs:element minOccurs="0" name="guardSan" type="xs:string"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="detachGuardSanBySanResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:detachGuardSanResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getEnterEventListBySan">  <xs:sequence>  <xs:element minOccurs="0" name="san" type="xs:string"/>  <xs:element minOccurs="0" name="startDate" type="xs:dateTime"/>  <xs:element minOccurs="0" name="endDate" type="xs:dateTime"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="getEnterEventListBySanResponse">  <xs:sequence>  <xs:element minOccurs="0" name="return" type="tns:enterEventListResult"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="ClientSummary">  <xs:sequence/>  <xs:attribute name="DateOfContract" type="xs:date"/>  <xs:attribute name="StateOfContract" type="xs:string"/>  <xs:attribute name="Balance" type="xs:long"/>  <xs:attribute name="OverdraftLimit" type="xs:long"/>  </xs:complexType>  <xs:complexType name="PurchaseList">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="P" type="tns:Purchase"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="Purchase">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="E" type="tns:PurchaseElement"/>  </xs:sequence>  <xs:attribute name="Time" type="xs:dateTime"/>  <xs:attribute name="Sum" type="xs:long"/>  <xs:attribute name="Discount" type="xs:long"/>  <xs:attribute name="Donation" type="xs:long"/>  <xs:attribute name="ByCash" type="xs:long"/>  <xs:attribute name="ByCard" type="xs:long"/>  <xs:attribute name="IdOfCard" type="xs:long"/>  </xs:complexType>  <xs:complexType name="PurchaseElement">  <xs:sequence/>  <xs:attribute name="Name" type="xs:string"/>  <xs:attribute name="Sum" type="xs:long"/>  <xs:attribute name="Amount" type="xs:long"/>  </xs:complexType>  <xs:complexType name="MenuList">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="M" type="tns:MenuDateItem"/>  </xs:sequence>  </xs:complexType>  <xs:complexType name="MenuDateItem">  <xs:sequence>  <xs:element maxOccurs="unbounded" minOccurs="0" name="E" type="tns:MenuItem"/>  </xs:sequence>  <xs:attribute name="Date" type="xs:date"/>  </xs:complexType>  <xs:complexType name="MenuItem">  <xs:sequence/>  <xs:attribute name="Group" type="xs:string"/>  <xs:attribute name="Name" type="xs:string"/>  <xs:attribute name="Price" type="xs:long"/>  </xs:complexType>  </xs:schema>  </wsdl:types>  <wsdl:message name="authorizeClientResponse">  <wsdl:part element="tns:authorizeClientResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="checkPasswordRestoreRequest">  <wsdl:part element="tns:checkPasswordRestoreRequest" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="setProhibitionOnGoodResponse">  <wsdl:part element="tns:setProhibitionOnGoodResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getSummaryByTypedIdResponse">  <wsdl:part element="tns:getSummaryByTypedIdResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getEnterEventListBySan">  <wsdl:part element="tns:getEnterEventListBySan" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getMenuListByOrgResponse">  <wsdl:part element="tns:getMenuListByOrgResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getBanks">  <wsdl:part element="tns:getBanks" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="changeMobilePhone">  <wsdl:part element="tns:changeMobilePhone" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getMenuListBySan">  <wsdl:part element="tns:getMenuListBySan" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getCardList">  <wsdl:part element="tns:getCardList" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getSummaryBySan">  <wsdl:part element="tns:getSummaryBySan" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getEnterEventListResponse">  <wsdl:part element="tns:getEnterEventListResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="setNotificationSettingsResponse">  <wsdl:part element="tns:setNotificationSettingsResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="createPaymentOrderResponse">  <wsdl:part element="tns:createPaymentOrderResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getClientStatsResponse">  <wsdl:part element="tns:getClientStatsResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getCardListResponse">  <wsdl:part element="tns:getCardListResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="sendLinkingTokenByContractIdResponse">  <wsdl:part element="tns:sendLinkingTokenByContractIdResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="setProhibitionOnProductGroupResponse">  <wsdl:part element="tns:setProhibitionOnProductGroupResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getGroupListByOrg">  <wsdl:part element="tns:getGroupListByOrg" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getStudentsByCanNotConfirmPayment">  <wsdl:part element="tns:getStudentsByCanNotConfirmPayment" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getStudentListByIdOfClientGroupResponse">  <wsdl:part element="tns:getStudentListByIdOfClientGroupResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="generateLinkingTokenResponse">  <wsdl:part element="tns:generateLinkingTokenResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getIdOfClient">  <wsdl:part element="tns:getIdOfClient" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getDishProhibitionsListResponse">  <wsdl:part element="tns:getDishProhibitionsListResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getMenuListBySanResponse">  <wsdl:part element="tns:getMenuListBySanResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="attachGuardSanBySanResponse">  <wsdl:part element="tns:attachGuardSanBySanResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getMenuList">  <wsdl:part element="tns:getMenuList" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="giveConclusionOnComplaintResponse">  <wsdl:part element="tns:giveConclusionOnComplaintResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getContractIdByCardNoResponse">  <wsdl:part element="tns:getContractIdByCardNoResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="sendLinkingTokenByMobileResponse">  <wsdl:part element="tns:sendLinkingTokenByMobileResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getPaymentListBySan">  <wsdl:part element="tns:getPaymentListBySan" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getSummaryBySanResponse">  <wsdl:part element="tns:getSummaryBySanResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="attachGuardSanResponse">  <wsdl:part element="tns:attachGuardSanResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="checkPasswordRestoreRequestResponse">  <wsdl:part element="tns:checkPasswordRestoreRequestResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getHiddenPagesResponse">  <wsdl:part element="tns:getHiddenPagesResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="changePaymentOrderStatusResponse">  <wsdl:part element="tns:changePaymentOrderStatusResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getComplexList">  <wsdl:part element="tns:getComplexList" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getBanksResponse">  <wsdl:part element="tns:getBanksResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getListOfGoodsResponse">  <wsdl:part element="tns:getListOfGoodsResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getSummaryResponse">  <wsdl:part element="tns:getSummaryResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="attachGuardSanBySan">  <wsdl:part element="tns:attachGuardSanBySan" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="detachGuardSanResponse">  <wsdl:part element="tns:detachGuardSanResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="changePersonalInfo">  <wsdl:part element="tns:changePersonalInfo" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="changeComplaintStatusToConsiderationResponse">  <wsdl:part element="tns:changeComplaintStatusToConsiderationResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="setProhibitionOnProduct">  <wsdl:part element="tns:setProhibitionOnProduct" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="changeEmailResponse">  <wsdl:part element="tns:changeEmailResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getClientsByGuardSan">  <wsdl:part element="tns:getClientsByGuardSan" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getListOfComplaintBookEntriesByOrg">  <wsdl:part element="tns:getListOfComplaintBookEntriesByOrg" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getPurchaseListBySanResponse">  <wsdl:part element="tns:getPurchaseListBySanResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="changePersonalInfoResponse">  <wsdl:part element="tns:changePersonalInfoResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="changeComplaintStatusToConsideration">  <wsdl:part element="tns:changeComplaintStatusToConsideration" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getCardListBySan">  <wsdl:part element="tns:getCardListBySan" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getStudentsByCanNotConfirmPaymentResponse">  <wsdl:part element="tns:getStudentsByCanNotConfirmPaymentResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getListOfComplaintBookEntriesByClient">  <wsdl:part element="tns:getListOfComplaintBookEntriesByClient" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="changeComplaintStatusToInvestigationResponse">  <wsdl:part element="tns:changeComplaintStatusToInvestigationResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getNotificationTypesResponse">  <wsdl:part element="tns:getNotificationTypesResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getPurchaseList">  <wsdl:part element="tns:getPurchaseList" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getDishProhibitionsList">  <wsdl:part element="tns:getDishProhibitionsList" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="detachGuardSanBySan">  <wsdl:part element="tns:detachGuardSanBySan" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="enableNotificationByEmail">  <wsdl:part element="tns:enableNotificationByEmail" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="setProhibitionOnGoodGroup">  <wsdl:part element="tns:setProhibitionOnGoodGroup" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getContractIdByCardNo">  <wsdl:part element="tns:getContractIdByCardNo" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="changeExpenditureLimitResponse">  <wsdl:part element="tns:changeExpenditureLimitResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getListOfGoods">  <wsdl:part element="tns:getListOfGoods" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="enableNotificationByEmailResponse">  <wsdl:part element="tns:enableNotificationByEmailResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="detachGuardSan">  <wsdl:part element="tns:detachGuardSan" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="setProhibitionOnGood">  <wsdl:part element="tns:setProhibitionOnGood" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getListOfProductsResponse">  <wsdl:part element="tns:getListOfProductsResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getActiveMenuQuestions">  <wsdl:part element="tns:getActiveMenuQuestions" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getSummaryByGuardSan">  <wsdl:part element="tns:getSummaryByGuardSan" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getCirculationList">  <wsdl:part element="tns:getCirculationList" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getRBKMoneyConfigResponse">  <wsdl:part element="tns:getRBKMoneyConfigResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getClientsByGuardSanResponse">  <wsdl:part element="tns:getClientsByGuardSanResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getListOfComplaintBookEntriesByOrgResponse">  <wsdl:part element="tns:getListOfComplaintBookEntriesByOrgResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getIdOfClientResponse">  <wsdl:part element="tns:getIdOfClientResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getSummaryByTypedId">  <wsdl:part element="tns:getSummaryByTypedId" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="giveConclusionOnComplaint">  <wsdl:part element="tns:giveConclusionOnComplaint" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="generateLinkingToken">  <wsdl:part element="tns:generateLinkingToken" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="excludeGoodFromProhibition">  <wsdl:part element="tns:excludeGoodFromProhibition" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getNotificationTypes">  <wsdl:part element="tns:getNotificationTypes" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getChronopayConfig">  <wsdl:part element="tns:getChronopayConfig" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getMenuListResponse">  <wsdl:part element="tns:getMenuListResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="excludeGoodFromProhibitionResponse">  <wsdl:part element="tns:excludeGoodFromProhibitionResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="changePaymentOrderStatus">  <wsdl:part element="tns:changePaymentOrderStatus" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getPurchaseListBySan">  <wsdl:part element="tns:getPurchaseListBySan" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="excludeGoodGroupFromProhibitionResponse">  <wsdl:part element="tns:excludeGoodGroupFromProhibitionResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="sendLinkingTokenByMobile">  <wsdl:part element="tns:sendLinkingTokenByMobile" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getRBKMoneyConfig">  <wsdl:part element="tns:getRBKMoneyConfig" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="setProhibitionOnProductGroup">  <wsdl:part element="tns:setProhibitionOnProductGroup" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="excludeGoodGroupFromProhibition">  <wsdl:part element="tns:excludeGoodGroupFromProhibition" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="changeComplaintStatusToInvestigation">  <wsdl:part element="tns:changeComplaintStatusToInvestigation" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="sendPasswordRecoverURLFromEmailResponse">  <wsdl:part element="tns:sendPasswordRecoverURLFromEmailResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="transferBalanceResponse">  <wsdl:part element="tns: transferBalanceResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getSummaryByGuardSanResponse">  <wsdl:part element="tns:getSummaryByGuardSanResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getChronopayConfigResponse">  <wsdl:part element="tns:getChronopayConfigResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getSummary">  <wsdl:part element="tns:getSummary" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getActiveMenuQuestionsResponse">  <wsdl:part element="tns:getActiveMenuQuestionsResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getIdOfContragent">  <wsdl:part element="tns:getIdOfContragent" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getClientSmsListResponse">  <wsdl:part element="tns:getClientSmsListResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getStudentListByIdOfClientGroup">  <wsdl:part element="tns:getStudentListByIdOfClientGroup" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="attachGuardSan">  <wsdl:part element="tns:attachGuardSan" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="changeEmail">  <wsdl:part element="tns:changeEmail" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getPaymentListResponse">  <wsdl:part element="tns:getPaymentListResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getListOfProducts">  <wsdl:part element="tns:getListOfProducts" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="authorizeClient">  <wsdl:part element="tns:authorizeClient" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="openComplaint">  <wsdl:part element="tns:openComplaint" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="detachGuardSanBySanResponse">  <wsdl:part element="tns:detachGuardSanBySanResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getNotificationSettings">  <wsdl:part element="tns:getNotificationSettings" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getIdOfContragentResponse">  <wsdl:part element="tns:getIdOfContragentResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="setAnswerFromQuestion">  <wsdl:part element="tns:setAnswerFromQuestion" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getCardListBySanResponse">  <wsdl:part element="tns:getCardListBySanResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="sendPasswordRecoverURLFromEmail">  <wsdl:part element="tns:sendPasswordRecoverURLFromEmail" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="transferBalance">  <wsdl:part element="tns: transferBalance" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getPaymentList">  <wsdl:part element="tns:getPaymentList" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="changeMobilePhoneResponse">  <wsdl:part element="tns:changeMobilePhoneResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="openComplaintResponse">  <wsdl:part element="tns:openComplaintResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="setNotificationSettings">  <wsdl:part element="tns:setNotificationSettings" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getPaymentListBySanResponse">  <wsdl:part element="tns:getPaymentListBySanResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="setProhibitionOnProductResponse">  <wsdl:part element="tns:setProhibitionOnProductResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="sendLinkingTokenByContractId">  <wsdl:part element="tns:sendLinkingTokenByContractId" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="setAnswerFromQuestionResponse">  <wsdl:part element="tns:setAnswerFromQuestionResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="changeExpenditureLimit">  <wsdl:part element="tns:changeExpenditureLimit" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getComplexListResponse">  <wsdl:part element="tns:getComplexListResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="activateLinkingTokenResponse">  <wsdl:part element="tns:activateLinkingTokenResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="changePassword">  <wsdl:part element="tns:changePassword" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getNotificationSettingsResponse">  <wsdl:part element="tns:getNotificationSettingsResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getListOfComplaintBookEntriesByClientResponse">  <wsdl:part element="tns:getListOfComplaintBookEntriesByClientResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="enableNotificationBySMSResponse">  <wsdl:part element="tns:enableNotificationBySMSResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getEnterEventListBySanResponse">  <wsdl:part element="tns:getEnterEventListBySanResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="changePasswordResponse">  <wsdl:part element="tns:changePasswordResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getHiddenPages">  <wsdl:part element="tns:getHiddenPages" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getClientStats">  <wsdl:part element="tns:getClientStats" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getClientSmsList">  <wsdl:part element="tns:getClientSmsList" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getCirculationListResponse">  <wsdl:part element="tns:getCirculationListResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="setProhibitionOnGoodGroupResponse">  <wsdl:part element="tns:setProhibitionOnGoodGroupResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="activateLinkingToken">  <wsdl:part element="tns:activateLinkingToken" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getMenuListByOrg">  <wsdl:part element="tns:getMenuListByOrg" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="createPaymentOrder">  <wsdl:part element="tns:createPaymentOrder" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getPurchaseListResponse">  <wsdl:part element="tns:getPurchaseListResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getEnterEventList">  <wsdl:part element="tns:getEnterEventList" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="enableNotificationBySMS">  <wsdl:part element="tns:enableNotificationBySMS" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:message name="getGroupListByOrgResponse">  <wsdl:part element="tns:getGroupListByOrgResponse" name="parameters">  </wsdl:part>  </wsdl:message>  <wsdl:portType name="ClientRoomController">  <wsdl:operation name="changeComplaintStatusToInvestigation">  <wsdl:input message="tns:changeComplaintStatusToInvestigation" name="changeComplaintStatusToInvestigation">  </wsdl:input>  <wsdl:output message="tns:changeComplaintStatusToInvestigationResponse" name="changeComplaintStatusToInvestigationResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="activateLinkingToken">  <wsdl:input message="tns:activateLinkingToken" name="activateLinkingToken">  </wsdl:input>  <wsdl:output message="tns:activateLinkingTokenResponse" name="activateLinkingTokenResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="setNotificationSettings">  <wsdl:input message="tns:setNotificationSettings" name="setNotificationSettings">  </wsdl:input>  <wsdl:output message="tns:setNotificationSettingsResponse" name="setNotificationSettingsResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="excludeGoodFromProhibition">  <wsdl:input message="tns:excludeGoodFromProhibition" name="excludeGoodFromProhibition">  </wsdl:input>  <wsdl:output message="tns:excludeGoodFromProhibitionResponse" name="excludeGoodFromProhibitionResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="sendLinkingTokenByContractId">  <wsdl:input message="tns:sendLinkingTokenByContractId" name="sendLinkingTokenByContractId">  </wsdl:input>  <wsdl:output message="tns:sendLinkingTokenByContractIdResponse" name="sendLinkingTokenByContractIdResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="createPaymentOrder">  <wsdl:input message="tns:createPaymentOrder" name="createPaymentOrder">  </wsdl:input>  <wsdl:output message="tns:createPaymentOrderResponse" name="createPaymentOrderResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getListOfComplaintBookEntriesByClient">  <wsdl:input message="tns:getListOfComplaintBookEntriesByClient" name="getListOfComplaintBookEntriesByClient">  </wsdl:input>  <wsdl:output message="tns:getListOfComplaintBookEntriesByClientResponse" name="getListOfComplaintBookEntriesByClientResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getListOfComplaintBookEntriesByOrg">  <wsdl:input message="tns:getListOfComplaintBookEntriesByOrg" name="getListOfComplaintBookEntriesByOrg">  </wsdl:input>  <wsdl:output message="tns:getListOfComplaintBookEntriesByOrgResponse" name="getListOfComplaintBookEntriesByOrgResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="transferBalance">  <wsdl:input message="tns: transferBalance" name="transferBalance">  </wsdl:input>  <wsdl:output message="tns: transferBalanceResponse" name="transferBalanceResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getChronopayConfig">  <wsdl:input message="tns:getChronopayConfig" name="getChronopayConfig">  </wsdl:input>  <wsdl:output message="tns:getChronopayConfigResponse" name="getChronopayConfigResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getClientStats">  <wsdl:input message="tns:getClientStats" name="getClientStats">  </wsdl:input>  <wsdl:output message="tns:getClientStatsResponse" name="getClientStatsResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="changeMobilePhone">  <wsdl:input message="tns:changeMobilePhone" name="changeMobilePhone">  </wsdl:input>  <wsdl:output message="tns:changeMobilePhoneResponse" name="changeMobilePhoneResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getStudentsByCanNotConfirmPayment">  <wsdl:input message="tns:getStudentsByCanNotConfirmPayment" name="getStudentsByCanNotConfirmPayment">  </wsdl:input>  <wsdl:output message="tns:getStudentsByCanNotConfirmPaymentResponse" name="getStudentsByCanNotConfirmPaymentResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="generateLinkingToken">  <wsdl:input message="tns:generateLinkingToken" name="generateLinkingToken">  </wsdl:input>  <wsdl:output message="tns:generateLinkingTokenResponse" name="generateLinkingTokenResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="checkPasswordRestoreRequest">  <wsdl:input message="tns:checkPasswordRestoreRequest" name="checkPasswordRestoreRequest">  </wsdl:input>  <wsdl:output message="tns:checkPasswordRestoreRequestResponse" name="checkPasswordRestoreRequestResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getListOfProducts">  <wsdl:input message="tns:getListOfProducts" name="getListOfProducts">  </wsdl:input>  <wsdl:output message="tns:getListOfProductsResponse" name="getListOfProductsResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="sendLinkingTokenByMobile">  <wsdl:input message="tns:sendLinkingTokenByMobile" name="sendLinkingTokenByMobile">  </wsdl:input>  <wsdl:output message="tns:sendLinkingTokenByMobileResponse" name="sendLinkingTokenByMobileResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getSummary">  <wsdl:input message="tns:getSummary" name="getSummary">  </wsdl:input>  <wsdl:output message="tns:getSummaryResponse" name="getSummaryResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="changePaymentOrderStatus">  <wsdl:input message="tns:changePaymentOrderStatus" name="changePaymentOrderStatus">  </wsdl:input>  <wsdl:output message="tns:changePaymentOrderStatusResponse" name="changePaymentOrderStatusResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="giveConclusionOnComplaint">  <wsdl:input message="tns:giveConclusionOnComplaint" name="giveConclusionOnComplaint">  </wsdl:input>  <wsdl:output message="tns:giveConclusionOnComplaintResponse" name="giveConclusionOnComplaintResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getHiddenPages">  <wsdl:input message="tns:getHiddenPages" name="getHiddenPages">  </wsdl:input>  <wsdl:output message="tns:getHiddenPagesResponse" name="getHiddenPagesResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getNotificationTypes">  <wsdl:input message="tns:getNotificationTypes" name="getNotificationTypes">  </wsdl:input>  <wsdl:output message="tns:getNotificationTypesResponse" name="getNotificationTypesResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getCardListBySan">  <wsdl:input message="tns:getCardListBySan" name="getCardListBySan">  </wsdl:input>  <wsdl:output message="tns:getCardListBySanResponse" name="getCardListBySanResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getPaymentList">  <wsdl:input message="tns:getPaymentList" name="getPaymentList">  </wsdl:input>  <wsdl:output message="tns:getPaymentListResponse" name="getPaymentListResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getDishProhibitionsList">  <wsdl:input message="tns:getDishProhibitionsList" name="getDishProhibitionsList">  </wsdl:input>  <wsdl:output message="tns:getDishProhibitionsListResponse" name="getDishProhibitionsListResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getCardList">  <wsdl:input message="tns:getCardList" name="getCardList">  </wsdl:input>  <wsdl:output message="tns:getCardListResponse" name="getCardListResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="detachGuardSan">  <wsdl:input message="tns:detachGuardSan" name="detachGuardSan">  </wsdl:input>  <wsdl:output message="tns:detachGuardSanResponse" name="detachGuardSanResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="enableNotificationBySMS">  <wsdl:input message="tns:enableNotificationBySMS" name="enableNotificationBySMS">  </wsdl:input>  <wsdl:output message="tns:enableNotificationBySMSResponse" name="enableNotificationBySMSResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getClientSmsList">  <wsdl:input message="tns:getClientSmsList" name="getClientSmsList">  </wsdl:input>  <wsdl:output message="tns:getClientSmsListResponse" name="getClientSmsListResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="setProhibitionOnGoodGroup">  <wsdl:input message="tns:setProhibitionOnGoodGroup" name="setProhibitionOnGoodGroup">  </wsdl:input>  <wsdl:output message="tns:setProhibitionOnGoodGroupResponse" name="setProhibitionOnGoodGroupResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getListOfGoods">  <wsdl:input message="tns:getListOfGoods" name="getListOfGoods">  </wsdl:input>  <wsdl:output message="tns:getListOfGoodsResponse" name="getListOfGoodsResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getActiveMenuQuestions">  <wsdl:input message="tns:getActiveMenuQuestions" name="getActiveMenuQuestions">  </wsdl:input>  <wsdl:output message="tns:getActiveMenuQuestionsResponse" name="getActiveMenuQuestionsResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="changeEmail">  <wsdl:input message="tns:changeEmail" name="changeEmail">  </wsdl:input>  <wsdl:output message="tns:changeEmailResponse" name="changeEmailResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="excludeGoodGroupFromProhibition">  <wsdl:input message="tns:excludeGoodGroupFromProhibition" name="excludeGoodGroupFromProhibition">  </wsdl:input>  <wsdl:output message="tns:excludeGoodGroupFromProhibitionResponse" name="excludeGoodGroupFromProhibitionResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="enableNotificationByEmail">  <wsdl:input message="tns:enableNotificationByEmail" name="enableNotificationByEmail">  </wsdl:input>  <wsdl:output message="tns:enableNotificationByEmailResponse" name="enableNotificationByEmailResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="attachGuardSan">  <wsdl:input message="tns:attachGuardSan" name="attachGuardSan">  </wsdl:input>  <wsdl:output message="tns:attachGuardSanResponse" name="attachGuardSanResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="setAnswerFromQuestion">  <wsdl:input message="tns:setAnswerFromQuestion" name="setAnswerFromQuestion">  </wsdl:input>  <wsdl:output message="tns:setAnswerFromQuestionResponse" name="setAnswerFromQuestionResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="changePersonalInfo">  <wsdl:input message="tns:changePersonalInfo" name="changePersonalInfo">  </wsdl:input>  <wsdl:output message="tns:changePersonalInfoResponse" name="changePersonalInfoResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getStudentListByIdOfClientGroup">  <wsdl:input message="tns:getStudentListByIdOfClientGroup" name="getStudentListByIdOfClientGroup">  </wsdl:input>  <wsdl:output message="tns:getStudentListByIdOfClientGroupResponse" name="getStudentListByIdOfClientGroupResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getPaymentListBySan">  <wsdl:input message="tns:getPaymentListBySan" name="getPaymentListBySan">  </wsdl:input>  <wsdl:output message="tns:getPaymentListBySanResponse" name="getPaymentListBySanResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getClientsByGuardSan">  <wsdl:input message="tns:getClientsByGuardSan" name="getClientsByGuardSan">  </wsdl:input>  <wsdl:output message="tns:getClientsByGuardSanResponse" name="getClientsByGuardSanResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="setProhibitionOnProduct">  <wsdl:input message="tns:setProhibitionOnProduct" name="setProhibitionOnProduct">  </wsdl:input>  <wsdl:output message="tns:setProhibitionOnProductResponse" name="setProhibitionOnProductResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getGroupListByOrg">  <wsdl:input message="tns:getGroupListByOrg" name="getGroupListByOrg">  </wsdl:input>  <wsdl:output message="tns:getGroupListByOrgResponse" name="getGroupListByOrgResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getSummaryBySan">  <wsdl:input message="tns:getSummaryBySan" name="getSummaryBySan">  </wsdl:input>  <wsdl:output message="tns:getSummaryBySanResponse" name="getSummaryBySanResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getEnterEventList">  <wsdl:input message="tns:getEnterEventList" name="getEnterEventList">  </wsdl:input>  <wsdl:output message="tns:getEnterEventListResponse" name="getEnterEventListResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getBanks">  <wsdl:input message="tns:getBanks" name="getBanks">  </wsdl:input>  <wsdl:output message="tns:getBanksResponse" name="getBanksResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="setProhibitionOnGood">  <wsdl:input message="tns:setProhibitionOnGood" name="setProhibitionOnGood">  </wsdl:input>  <wsdl:output message="tns:setProhibitionOnGoodResponse" name="setProhibitionOnGoodResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="setProhibitionOnProductGroup">  <wsdl:input message="tns:setProhibitionOnProductGroup" name="setProhibitionOnProductGroup">  </wsdl:input>  <wsdl:output message="tns:setProhibitionOnProductGroupResponse" name="setProhibitionOnProductGroupResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getContractIdByCardNo">  <wsdl:input message="tns:getContractIdByCardNo" name="getContractIdByCardNo">  </wsdl:input>  <wsdl:output message="tns:getContractIdByCardNoResponse" name="getContractIdByCardNoResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getMenuListBySan">  <wsdl:input message="tns:getMenuListBySan" name="getMenuListBySan">  </wsdl:input>  <wsdl:output message="tns:getMenuListBySanResponse" name="getMenuListBySanResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getSummaryByTypedId">  <wsdl:input message="tns:getSummaryByTypedId" name="getSummaryByTypedId">  </wsdl:input>  <wsdl:output message="tns:getSummaryByTypedIdResponse" name="getSummaryByTypedIdResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getSummaryByGuardSan">  <wsdl:input message="tns:getSummaryByGuardSan" name="getSummaryByGuardSan">  </wsdl:input>  <wsdl:output message="tns:getSummaryByGuardSanResponse" name="getSummaryByGuardSanResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getPurchaseListBySan">  <wsdl:input message="tns:getPurchaseListBySan" name="getPurchaseListBySan">  </wsdl:input>  <wsdl:output message="tns:getPurchaseListBySanResponse" name="getPurchaseListBySanResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getMenuList">  <wsdl:input message="tns:getMenuList" name="getMenuList">  </wsdl:input>  <wsdl:output message="tns:getMenuListResponse" name="getMenuListResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getIdOfContragent">  <wsdl:input message="tns:getIdOfContragent" name="getIdOfContragent">  </wsdl:input>  <wsdl:output message="tns:getIdOfContragentResponse" name="getIdOfContragentResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="openComplaint">  <wsdl:input message="tns:openComplaint" name="openComplaint">  </wsdl:input>  <wsdl:output message="tns:openComplaintResponse" name="openComplaintResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getRBKMoneyConfig">  <wsdl:input message="tns:getRBKMoneyConfig" name="getRBKMoneyConfig">  </wsdl:input>  <wsdl:output message="tns:getRBKMoneyConfigResponse" name="getRBKMoneyConfigResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="authorizeClient">  <wsdl:input message="tns:authorizeClient" name="authorizeClient">  </wsdl:input>  <wsdl:output message="tns:authorizeClientResponse" name="authorizeClientResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="sendPasswordRecoverURLFromEmail">  <wsdl:input message="tns:sendPasswordRecoverURLFromEmail" name="sendPasswordRecoverURLFromEmail">  </wsdl:input>  <wsdl:output message="tns:sendPasswordRecoverURLFromEmailResponse" name="sendPasswordRecoverURLFromEmailResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getNotificationSettings">  <wsdl:input message="tns:getNotificationSettings" name="getNotificationSettings">  </wsdl:input>  <wsdl:output message="tns:getNotificationSettingsResponse" name="getNotificationSettingsResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getCirculationList">  <wsdl:input message="tns:getCirculationList" name="getCirculationList">  </wsdl:input>  <wsdl:output message="tns:getCirculationListResponse" name="getCirculationListResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="changeComplaintStatusToConsideration">  <wsdl:input message="tns:changeComplaintStatusToConsideration" name="changeComplaintStatusToConsideration">  </wsdl:input>  <wsdl:output message="tns:changeComplaintStatusToConsiderationResponse" name="changeComplaintStatusToConsiderationResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="changePassword">  <wsdl:input message="tns:changePassword" name="changePassword">  </wsdl:input>  <wsdl:output message="tns:changePasswordResponse" name="changePasswordResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getPurchaseList">  <wsdl:input message="tns:getPurchaseList" name="getPurchaseList">  </wsdl:input>  <wsdl:output message="tns:getPurchaseListResponse" name="getPurchaseListResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="changeExpenditureLimit">  <wsdl:input message="tns:changeExpenditureLimit" name="changeExpenditureLimit">  </wsdl:input>  <wsdl:output message="tns:changeExpenditureLimitResponse" name="changeExpenditureLimitResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getIdOfClient">  <wsdl:input message="tns:getIdOfClient" name="getIdOfClient">  </wsdl:input>  <wsdl:output message="tns:getIdOfClientResponse" name="getIdOfClientResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getMenuListByOrg">  <wsdl:input message="tns:getMenuListByOrg" name="getMenuListByOrg">  </wsdl:input>  <wsdl:output message="tns:getMenuListByOrgResponse" name="getMenuListByOrgResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getComplexList">  <wsdl:input message="tns:getComplexList" name="getComplexList">  </wsdl:input>  <wsdl:output message="tns:getComplexListResponse" name="getComplexListResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="attachGuardSanBySan">  <wsdl:input message="tns:attachGuardSanBySan" name="attachGuardSanBySan">  </wsdl:input>  <wsdl:output message="tns:attachGuardSanBySanResponse" name="attachGuardSanBySanResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="detachGuardSanBySan">  <wsdl:input message="tns:detachGuardSanBySan" name="detachGuardSanBySan">  </wsdl:input>  <wsdl:output message="tns:detachGuardSanBySanResponse" name="detachGuardSanBySanResponse">  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getEnterEventListBySan">  <wsdl:input message="tns:getEnterEventListBySan" name="getEnterEventListBySan">  </wsdl:input>  <wsdl:output message="tns:getEnterEventListBySanResponse" name="getEnterEventListBySanResponse">  </wsdl:output>  </wsdl:operation>  </wsdl:portType>  <wsdl:binding name="ClientRoomControllerWSServiceSoapBinding" type="tns:ClientRoomController">  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>  <wsdl:operation name="changeComplaintStatusToInvestigation">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="changeComplaintStatusToInvestigation">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="changeComplaintStatusToInvestigationResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="activateLinkingToken">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="activateLinkingToken">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="activateLinkingTokenResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="excludeGoodFromProhibition">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="excludeGoodFromProhibition">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="excludeGoodFromProhibitionResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="setNotificationSettings">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="setNotificationSettings">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="setNotificationSettingsResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="createPaymentOrder">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="createPaymentOrder">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="createPaymentOrderResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="sendLinkingTokenByContractId">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="sendLinkingTokenByContractId">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="sendLinkingTokenByContractIdResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getListOfComplaintBookEntriesByClient">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getListOfComplaintBookEntriesByClient">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getListOfComplaintBookEntriesByClientResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getListOfComplaintBookEntriesByOrg">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getListOfComplaintBookEntriesByOrg">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getListOfComplaintBookEntriesByOrgResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getChronopayConfig">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getChronopayConfig">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getChronopayConfigResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getClientStats">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getClientStats">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getClientStatsResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="changeMobilePhone">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="changeMobilePhone">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="changeMobilePhoneResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="checkPasswordRestoreRequest">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="checkPasswordRestoreRequest">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="checkPasswordRestoreRequestResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="generateLinkingToken">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="generateLinkingToken">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="generateLinkingTokenResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getStudentsByCanNotConfirmPayment">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getStudentsByCanNotConfirmPayment">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getStudentsByCanNotConfirmPaymentResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getListOfProducts">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getListOfProducts">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getListOfProductsResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="sendLinkingTokenByMobile">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="sendLinkingTokenByMobile">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="sendLinkingTokenByMobileResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getSummary">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getSummary">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getSummaryResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="changePaymentOrderStatus">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="changePaymentOrderStatus">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="changePaymentOrderStatusResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="giveConclusionOnComplaint">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="giveConclusionOnComplaint">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="giveConclusionOnComplaintResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getHiddenPages">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getHiddenPages">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getHiddenPagesResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getCardListBySan">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getCardListBySan">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getCardListBySanResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getNotificationTypes">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getNotificationTypes">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getNotificationTypesResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getPaymentList">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getPaymentList">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getPaymentListResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="detachGuardSan">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="detachGuardSan">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="detachGuardSanResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getCardList">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getCardList">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getCardListResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getDishProhibitionsList">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getDishProhibitionsList">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getDishProhibitionsListResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="enableNotificationBySMS">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="enableNotificationBySMS">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="enableNotificationBySMSResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getClientSmsList">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getClientSmsList">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getClientSmsListResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="changeEmail">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="changeEmail">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="changeEmailResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getActiveMenuQuestions">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getActiveMenuQuestions">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getActiveMenuQuestionsResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="setProhibitionOnGoodGroup">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="setProhibitionOnGoodGroup">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="setProhibitionOnGoodGroupResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getListOfGoods">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getListOfGoods">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getListOfGoodsResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="excludeGoodGroupFromProhibition">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="excludeGoodGroupFromProhibition">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="excludeGoodGroupFromProhibitionResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="enableNotificationByEmail">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="enableNotificationByEmail">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="enableNotificationByEmailResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="attachGuardSan">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="attachGuardSan">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="attachGuardSanResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="setAnswerFromQuestion">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="setAnswerFromQuestion">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="setAnswerFromQuestionResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="changePersonalInfo">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="changePersonalInfo">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="changePersonalInfoResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getPaymentListBySan">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getPaymentListBySan">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getPaymentListBySanResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getStudentListByIdOfClientGroup">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getStudentListByIdOfClientGroup">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getStudentListByIdOfClientGroupResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getClientsByGuardSan">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getClientsByGuardSan">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getClientsByGuardSanResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="setProhibitionOnProduct">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="setProhibitionOnProduct">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="setProhibitionOnProductResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getGroupListByOrg">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getGroupListByOrg">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getGroupListByOrgResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getBanks">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getBanks">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getBanksResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getEnterEventList">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getEnterEventList">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getEnterEventListResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getSummaryBySan">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getSummaryBySan">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getSummaryBySanResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="setProhibitionOnGood">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="setProhibitionOnGood">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="setProhibitionOnGoodResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="setProhibitionOnProductGroup">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="setProhibitionOnProductGroup">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="setProhibitionOnProductGroupResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getContractIdByCardNo">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getContractIdByCardNo">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getContractIdByCardNoResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getMenuListBySan">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getMenuListBySan">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getMenuListBySanResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getSummaryByTypedId">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getSummaryByTypedId">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getSummaryByTypedIdResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getSummaryByGuardSan">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getSummaryByGuardSan">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getSummaryByGuardSanResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getIdOfContragent">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getIdOfContragent">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getIdOfContragentResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getMenuList">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getMenuList">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getMenuListResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getPurchaseListBySan">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getPurchaseListBySan">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getPurchaseListBySanResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getRBKMoneyConfig">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getRBKMoneyConfig">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getRBKMoneyConfigResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="openComplaint">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="openComplaint">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="openComplaintResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="authorizeClient">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="authorizeClient">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="authorizeClientResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="sendPasswordRecoverURLFromEmail">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="sendPasswordRecoverURLFromEmail">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="sendPasswordRecoverURLFromEmailResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getCirculationList">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getCirculationList">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getCirculationListResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getNotificationSettings">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getNotificationSettings">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getNotificationSettingsResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="changeComplaintStatusToConsideration">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="changeComplaintStatusToConsideration">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="changeComplaintStatusToConsiderationResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="changePassword">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="changePassword">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="changePasswordResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getPurchaseList">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getPurchaseList">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getPurchaseListResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="changeExpenditureLimit">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="changeExpenditureLimit">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="changeExpenditureLimitResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getIdOfClient">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getIdOfClient">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getIdOfClientResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getMenuListByOrg">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getMenuListByOrg">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getMenuListByOrgResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getComplexList">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getComplexList">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getComplexListResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="attachGuardSanBySan">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="attachGuardSanBySan">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="attachGuardSanBySanResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="detachGuardSanBySan">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="detachGuardSanBySan">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="detachGuardSanBySanResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  <wsdl:operation name="getEnterEventListBySan">  <soap:operation soapAction="" style="document"/>  <wsdl:input name="getEnterEventListBySan">  <soap:body use="literal"/>  </wsdl:input>  <wsdl:output name="getEnterEventListBySanResponse">  <soap:body use="literal"/>  </wsdl:output>  </wsdl:operation>  </wsdl:binding>  <wsdl:service name="ClientRoomControllerWSService">  <wsdl:port binding="tns:ClientRoomControllerWSServiceSoapBinding" name="ClientRoomControllerWSPort">  <soap:address location="http://localhost:8080/processor/soap/client"/>  </wsdl:port>  </wsdl:service>  </wsdl:definitions> |

Описание общих структур данных

### Параметр комплексного типа: ClientSummaryExt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | ContractId | Номер договора (лицевого счета) | + | xs:long | Rgu:ResultInformation |
| 2 | dateOfContract | Дата регистрации клиента | + | xs:date |  |
| 3 | stateOfContract | Состояние договора клиента (Активен, Не заключен, Расторгнут по времени, Расторгнут по желанию клиента) | + | xs:string |  |
| 4 | balance | Текущий баланс счета | + | xs:long | Суммарный баланс лицевого счета, включает в себя сумму основного счета и счета на Абонементное питание |
| 5 | subBalance0 | Текущий баланс основного счета | + | xs:long |  |
| 6 | subBalance1 | Текущий баланс на Абонементное питание | + | xs:long |  |
| 7 | overdraftLimit | Лимит овердрафта | + | xs:long |  |
| 8 | expenditureLimit | Дневной лимит затрат | + | xs:long |  |
| 9 | firstName | Имя клиента | + | xs:string |  |
| 10 | lastName | Фамилия клиента | + | xs:string |  |
| 11 | MiddleName | Отчество клиента | + | xs:string |  |
| 12 | grade | Класс (группа) | + | xs:string |  |
| 13 | officialName | Название организации | + | xs:string |  |
| 14 | LastEnterEventCode | Код последнего события контроля доступа:  Entry = 0; // вход  Exit = 1; // выход  PassageIsForbidden = 2; // проход запрещен  EventWithoutPassage = 4; // событие без прохода  PassageRufusal = 5; // отказ от прохода  ReEntry = 6; // повторный вход  ReExit = 7; // повторный выход | + | xs:int |  |
| 15 | LastEnterEventTime | Время последнего события контроля доступа | + | xs:dateTime |  |
| 16 | NotifyViaSMS | Включено ли уведомление по SMS | + | xs:boolean |  |
| 17 | NotifyViaEmail | Включено ли уведомление по E-Mail | + | xs:boolean |  |
| 18 | MobilePhone | Номер телефона для SMS уведомлений | + | xs:string |  |
| 19 | Email | E-mail для уведомлений | + | xs:string |  |
| 20 | DefaultMerchantId | Идентификатор поставщика по-умолчанию |  | xs:long |  |
|  | DefaultMerchantInfo | Дополнительные данные по поставщику |  | xs:string |  |

### Параметр комплексного типа: ClientItem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | ContractId | Номер договора (лицевого счета) | + | xs:long |  |
| 2 | San | СНИЛС | - | xs:string |  |

### Параметр комплексного типа: PurchaseListExt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | Time | Дата и время регистрации покупки | + | xs:dateTime |  |
| 2 | Sum | Сумма | + | xs:long |  |
| 3 | SocDiscount | Социальная скидка (для льготного питания) | + | xs:long |  |
| 4 | TrdDiscount | Торговая скидка | + | xs:long |  |
| 5 | Donation | Дотация | + | xs:long |  |
| 6 | ByCash | Сумма, оплаченная наличными | + | xs:long |  |
| 7 | ByCard | Сумма, оплаченная по карте | + | xs:long |  |
| 8 | E | Состав покупки |  | PurchaseElementExt |  |

1. Параметр комплексного типа: PurchaseElementExt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | Name | Наименование товара | + | xs:string |  |
| 2 | Sum | Сумма | + | xs:long |  |
| 3 | Amount | Количество | + | xs:long |  |
| 4 | Type | Тип | + | xs:int | 0 – блюдо, 1 – комплекс, 2 – блюдо из комплекса |

### Параметр комплексного типа: PurchaseExt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | Name | Наименование товара | + | xs:string |  |
| 2 | Sum | Сумма | + | xs:long |  |
| 3 | Amount | Количество | + | xs:long |  |
| 4 | Type | Тип | + | xs:int | 0 – блюдо, 1 – комплекс, 2 – блюдо из комплекса |

### Параметр комплексного типа: PaymentList

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | P | Данные пополнения |  | Payment |  |

### Параметр комплексного типа: Payment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | Time | Данные пополнения |  | xs:dateTime |  |
| 2 | Sum | Сумма |  | xs:long | В копейках |
| 3 | Origin | Наименование системы и метода оплаты |  | xs:string |  |

### Параметр комплексного типа: MenuListExt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | M | Данные меню на день |  | MenuDateItemExt |  |

### Параметр комплексного типа: MenuDateItemExt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | E | Элементы меню |  | MenuItemExt |  |
| 2 | Date | Дата меню | + | xs:date |  |

### Параметр комплексного типа: MenuItemExt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | Group | Группа (Первое, Второе, Напитки, и т.д.) | + | MenuItemExt |  |
| 2 | Name | Наименование | + | xs:date |  |
| 3 | Price | Цена | + | xs:long |  |
| 4 | Calories | Калории |  | xs:double |  |
| 5 | VitB1 | Витамин B1 |  | xs:double |  |
| 6 | VitC | Витамин C |  | xs:double |  |
| 7 | VitA | Витамин A |  | xs:double |  |
| 8 | VitE | Витамин E |  | xs:double |  |
| 9 | MinCa | Кальций |  | xs:double |  |
| 10 | MinP | Фосфор |  | xs:double |  |
| 11 | MinMg | Магний |  | xs:double |  |
| 12 | MinFe | Железо |  | xs:double |  |

### Параметр комплексного типа: EnterEventList

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | E | Данные прохода |  | EnterEventItem |  |

### Параметр комплексного типа: EnterEventItem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | dateTime | Дата и время события | + | xs:dateTime |  |
| 2 | day | День недели события | + | xs:int |  |
| 3 | enterName | Наименование входной группы | + | xs:string |  |
| 4 | direction | Направление прохода | + | xs:int | 0 – вход, 1 – выход, 2- проход запрещен, 3 – взлом турникета, 4 – событие без прохода, 5 – отказ от прохода, 6 – повторный вход, 7 – повторный выход |
| 5 | temporaryCard | Событие совершено по временной карте | + | xs:int | 0 — нет, 1 — да |
| 6 | GuardianSan | Снилс родителя который провел ребенка в детском садике | - | xs:string |  |

### Параметр комплексного типа: ClientNotificationSettingsItem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | typeOfNotification | Код типа уведомления | + | xs:long |  |
| 2 | nameOfNotification | Наименование | + | xs:long |  |

### Параметр комплексного типа: StudentsConfirmPaymentList

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | StudentMustPayItems | Данные клиентов |  | StudentMustPayItem |  |

### Параметр комплексного типа: StudentMustPayItem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | FirstName | Имя клиента | + | xs:string |  |
| 2 | Surname | Фамилия клиента | + | xs:string |  |
| 3 | SecondName | Отчество клиента | + | xs:string |  |
| 4 | Balance | Входной баланс в моент соверщения покупки | + | xs:long |  |
| 5 | CreateTime | Время покупки | + | xs:dateTime |  |
| 6 | PaySum | Стоимость покупки | + | xs:long |  |

### Параметр комплексного типа: CycleDiagramIn

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | Monday | Список идентификаторов комплексов питания на понедельник |  | xs:string | Разделитель идентификаторов – «;» |
| 2 | Tuesday | Список идентификаторов комплексов питания на вторник |  | xs:string | Разделитель идентификаторов – «;» |
| 3 | Wednesday | Список идентификаторов комплексов питания на среду |  | xs:string | Разделитель идентификаторов – «;» |
| 4 | Thursday | Список идентификаторов комплексов питания на четверг |  | xs:string | Разделитель идентификаторов – «;» |
| 5 | Friday | Список идентификаторов комплексов питания на пятницу |  | xs:string | Разделитель идентификаторов – «;» |
| 6 | Saturday | Список идентификаторов комплексов питания на субботу |  | xs:string | Разделитель идентификаторов – «;» |

### Параметр комплексного типа: ComplexInfoList

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | ComplexInfo | Описание одного комплекса |  | ComplexInfoExt |  |

### Параметр комплексного типа: ComplexInfoExt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Код параметра | Описание параметра | Обязательность | Способ заполнения/Тип | Комментарий |
| 1 | idOfComplexInfo | Идентификатор информации о комплексе | + | xs:long |  |
| 2 | idOfComplex | Идентификатор комплекса | + | xs:int |  |
| 3 | complexName | Наименование комплекса | + | xs:string |  |
| 4 | menuDate | Дата, на которую данная информация о комплексе актуальна | + | xs:dateTime |  |
| 5 | currentPrice | Цена комплекса | + | xs:long |  |
| 6 | usedSubscriptionFeeding | Отметка об участии комплекса в АП | + | xs:int | 1 – комплекс может участвовать в АП,  0 – не может. |

Все параметры данного типа – атрибуты.