**Приложение. Список методов**

http://veep.su/Rest.svc

1. **Аккаунт**
   1. **Поиск аккаунта при запуске Приложения**

/// <summary>

/// Поиск аккаунта по номеру телефона при старте Приложения

/// </summary>

/// <param name="phoneNumber">Номер телефона</param>

/// <param name="phoneCode">Код страны</param>

/// <param name="language">Язык интерфейса</param>

Account FindAccount(string phoneNumber, string phoneCode= “7”, int language=0)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| phoneNumber | строковый | Номер телефона |
| phoneCode | строковый | Код страны (по умолчанию “7”) |
| language | числовой | Язык интерфейса: по умолчанию 0 – русский,  1 - английский |

**Возвращаемые типы данных**

Аккаунт не найден – Account (Message)

Аккаунт найден – Account (часть полей)

class Account {

int ID;

string PhoneModel;

string PhoneCode;  
string PhoneNumber;  
string PhoneUID;  
PhoneOS PhoneOS;

string Pswd;   
 AccountBlock AccountBlock;

AccountStatus AccountStatus; ChangeNameCard

string Message;  
 }

class PhoneOS {

int ID;

string Name;

}

class AccountBlock {

int ID;

string Name;

}

class AccountStatus {

int ID;

string Name;

}

|  |  |  |
| --- | --- | --- |
| **Элемент** | **Тип значения** | **Значение** |
| ID | числовой | ID пользователя |
| PhoneModel | строковый | Модель телефона |
| PhoneCode | строковый | Код страны (по умолчанию “7”) |
| PhoneNumber | строковый | Номер телефона |
| PhoneUID | строковый | Уникальный идентификатор |
| PhoneOS | PhoneOS | ОС телефона |
| Pswd | строковый | Хэш пароля пользователя |
| AccountBlock | AccountBlock | Причина блокировки аккаунта:  1 - Смена устройства/перепрошивка  2 - Мошенничество  3 - Множественная валидация номера телефона  4 - Неверный логин/пароль |
| AccountStatus | AccountStatus | Статус аккаунта:  1 - Не активен  2 - Активен  3 - Активен (заблокирован вызов официантов)  4 - Заблокирован |
| Message | строковый | Сообщения / ошибки сервиса |

**Request XML:**

<phoneCode>7</phoneCoder>

<phoneNumber>9151234567</phoneNumber>

<language>0</language>

**Response XML:**

<Account>

<ID>123</ID>

<PhoneModel>iPhone 1</PhoneModel>

<PhoneCode>7</PhoneCode>

<PhoneNumber>9151234567</PhoneNumber>  
 <PhoneUID>HGR345765</PhoneUID>  
 <PhoneOS>

<ID>1</ID>

<Name>iOS</Name>

</PhoneOS>

<Pswd>sf56gdgdlhj42jkjhfr45ghy875fjy6l</Pswd>  
 <AccountBlock>

<ID>0<ID>

<Name></Name>

</AccountBlock>

<AccountStatus>

<ID>1</ID>

<Name>Не активен</Name>

</ AccountStatus>

<Message></Message>

</Account>

* 1. **Регистрация аккаунта**

/// <summary>

/// Регистрация аккаунта

/// </summary>

/// <param name="FirstName"> Имя пользователя</param>

/// <param name="LastName"> Фамилия пользователя</param>

/// <param name="PhoneModel"> Модель телефона</param>

/// <param name="PhoneCode">Код страны</param>

/// <param name="PhoneNumber">Номер телефона</param>

/// <param name="PhoneUID"> Уникальный идентификатор телефона</param>

/// <param name="PhoneOS.ID">ID ОС телефона</param>

/// <param name=”Pswd">Хэш пароля</param>

/// <param name=”Email">Email пользователя</param>

/// <param name="PhoneCode">Код страны</param>

/// <param name="language">Язык интерфейса</param>

Account RegistrAccount (string FirstName, string LastName, string PhoneModel, string PhoneNumber, string PhoneUID, int? PhoneOS\_ID, string Pswd, string Email, string PhoneCode = “7”, int language=0)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| FirstName | строковый | Имя пользователя |
| LastName | строковый | Фамилия пользователя |
| PhoneModel | строковый | Модель телефона |
| PhoneNumber | строковый | Номер телефона |
| PhoneUID | строковый | Уникальный идентификатор телефона |
| PhoneOS\_ID | числовой | ID ОС телефона |
| Pswd | строковый | Хэш пароля |
| Email | строковый | Email пользователя |
| PhoneCode | строковый | Код страны (по умолчанию “7”) |
| language | числовой | Язык интерфейса: по умолчанию 0 – русский,  1 - английский |

**Возвращаемые типы данных**

Аккаунт зарегистрирован – Account (часть полей)

Аккаунт не зарегистрирован – Account (Message)

class Account {

int ID;

string FirstName;

string LastName;

string PhoneModel;

string PhoneCode;

string PhoneNumber;

string PhoneUID;

PhoneOS PhoneOS;   
 string SMScode;  
 int IsValid;

AccountStatus AccountStatus;

string Message;

}

class PhoneOS {

int ID;

string Name;

}

class AccountStatus {

int ID;

string Name;

}

|  |  |  |
| --- | --- | --- |
| **Элемент** | **Тип значения** | **Значение** |
| ID | числовой | ID пользователя |
| FirstName | строковый | Имя пользователя |
| LastName | строковый | Фамилия пользователя |
| PhoneModel | строковый | Модель телефона |
| PhoneCode | строковый | Код страны (по умолчанию “7”) |
| PhoneNumber | строковый | Номер телефона |
| PhoneUID | строковый | Уникальный идентификатор телефона |
| PhoneOS.ID | числовой | ID ОС телефона |
| SMScode | строковый | Код, отправленный по SMS |
| IsValid | числовой | 1 – номер телефона валиден 0 – не валиден |
| AccountStatus.ID | числовой | ID статуса аккаунта:  1 - Не активен  2 - Активен  3 - Активен (заблокирован вызов официантов)  4 - Заблокирован |
| Message | строковый | Сообщения / ошибки сервиса |

**Request XML:**

<FirstName>Иван</FirstName>

<LastName>Петров</LastName>

< PhoneModel>iPhone 1</PhoneModel>

<PhoneCode>7</PhoneCode>

<PhoneNumber>9151234567</PhoneNumber>  
 <PhoneUID>HER345765</PhoneUID>  
 <PhoneOS\_ID>1</PhoneOS\_ID>

<Pswd>sf56gdgdlhj42jkjhfr45ghy875fjy6l</Pswd>  
 <Email>com@com.com</Email>

<language>0</language>

**Response XML:**

<Account>

<ID>123</ID>

<FirstName>Иван</FirstName>

<LastName>Петров</LastName>

<PhoneModel>iPhone 1</PhoneModel>

<PhoneCode>7</PhoneCode>

<PhoneNumber>9151234567</PhoneNumber>  
 <PhoneUID>HER345765</PhoneUID>  
 <PhoneOS>

<ID>1</ID>

<Name>iOS</Name>

</PhoneOS>

<SMScode>1234</SMScode>

<isValid>0</isValid>

<AccountStatus>

<ID>1</ID>

<Name>Не активен</Name>

</ AccountStatus>

<Message></Message>

</Account>

* 1. **Изменение статуса аккаунта**

/// <summary>

/// Изменение статуса аккаунта

/// </summary>

/// <param name=" accountID">ID аккаунта</param>

/// <param name=" statusID">ID статуса</param>

int UpdateAccountStatus (int accountID, int statusID)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| accountID | числовой | ID аккаунта |
| statusID | числовой | ID статуса аккаунта:  1 - Не активен  2 - Активен  3 - Активен (заблокирован вызов официантов)  4 - Заблокирован |

**Возвращаемые типы данных**

Статус аккаунта изменен – int accountID

Статус аккаунта не изменен – 0

**Request XML:**

<accountID>123</accountID>

<statusID>2</statusID>

**Response XML:**

<accountID>123</accountID>

* 1. **Авторизация аккаунта**

/// <summary>

/// Авторизация аккаунта

/// </summary>

/// <param name="phoneNumber">Номер телефона</param>

/// <param name="psw">Пароль</param>

/// <param name="phoneCode">Код страны</param>

string Login (string phoneNumber, string psw, string phoneCode = “7”)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| phoneNumber | строковый | Номер телефона |
| Psw | строковый | Хэш пароля |
| phoneCode | строковый | Код страны (по умолчанию “7”) |

**Возвращаемые типы данных**

Аккаунт зарегистрирован – string user\_key

Аккаунт не авторизован – user\_key = “”

**Request XML:**

<phoneNumber>9151234567</phoneNumber>

<Psw>sf56gdgdlhj42jkjhfr45ghy875fjy6l</Psw>

<phoneCode>7</phoneCode>

**Response XML:**

<user\_key>dlhj42jkjhfr45ghygdgdl</user\_key>

* 1. **Блокировка аккаунта**

/// <summary>

/// Блокировка аккаунта

/// </summary>

/// <param name="phoneNumber">Номер телефона</param>

/// <param name="block\_id"> Причина блокировки аккаунта</param>

/// <param name="phoneCode">Код страны</param>

int BlockedAccount (string phoneNumber, string block\_id, string phoneCode = “7”)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| phoneNumber | строковый | Номер телефона |
| block\_id | числовой | Причина блокировки аккаунта:  1 - Смена устройства/перепрошивка  2 - Мошенничество  3 - Множественная валидация номера телефона  4 - Неверный логин/пароль  5 – Новая регистрация аккаунта |
| phoneCode | строковый | Код страны (по умолчанию “7”) |

**Возвращаемые типы данных**

int result – результат блокировки: 1 – заблокирован, 0 – ошибка

**Request XML:**

<phoneNumber>9151234567</phoneNumber>

<block\_id>3</block\_id>

<phoneCode>7</phoneCode>

**Response XML:**

<result>1</result>

* 1. **Проверка открытой сессии**

/// <summary>

/// Проверка открытой сессии

/// </summary>

/// <param name="phoneNumber">Номер телефона</param>

/// <param name="phoneCode">Код страны</param>

string CheckSession (string phoneNumber, string phoneCode = “7”)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| phoneNumber | строковый | Номер телефона |
| phoneCode | строковый | Код страны (по умолчанию “7”) |

**Возвращаемые типы данных**

string user\_key – id открытой сессии

null – нет открытой сессии

**Request XML:**

<phoneNumber>9151234567</phoneNumber>

<phoneCode>7</phoneCode>

**Response XML:**

<user\_key>dlhj42jkjhfr45ghygdgdl</user\_key>

* 1. **Обновление данных аккаунта**

/// <summary>

/// Обновление данных аккаунта

/// </summary>

/// <param name="FirstName"> Имя пользователя</param>

/// <param name="LastName"> Фамилия пользователя</param>

/// <param name="PhoneModel"> Модель телефона</param>

/// <param name="PhoneCode">Код страны</param>

/// <param name="PhoneNumber">Номер телефона</param>

/// <param name="PhoneUID"> Уникальный идентификатор телефона</param>

/// <param name="PhoneOS.ID">ID ОС телефона</param>

/// <param name=”Email">Email пользователя</param>

/// <param name=”user\_key">ID сессии, если не авторизован (сессия закрыта) – то null</param>

/// <param name=” language ">Язык интерфейса</param>

Account UpdateAccount (string FirstName, string LastName, string PhoneModel, string PhoneNumber, string PhoneUID, int? PhoneOS\_ID, string Email, string user\_key, string PhoneCode = “7”, int language=0)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| FirstName | строковый | Имя пользователя |
| LastName | строковый | Фамилия пользователя |
| PhoneModel | строковый | Модель телефона |
| PhoneNumber | строковый | Номер телефона |
| PhoneUID | строковый | Уникальный идентификатор телефона |
| PhoneOS\_ID | числовой | ID ОС телефона |
| Email | строковый | Email пользователя |
| user\_key | строковый | ID сессии |
| PhoneCode | строковый | Код страны (по умолчанию “7”) |
| language | числовой | Язык интерфейса: по умолчанию 0 – русский,  1 - английский |

**Возвращаемые типы данных**

Аккаунт обновлен – Account (часть полей)

Аккаунт не обновлен - null

class Account {

int ID;

string FirstName;

string LastName;

string PhoneModel;

string PhoneCode;

string PhoneNumber;

string PhoneUID;

PhoneOS PhoneOS;

string Email;

AccountStatus AccountStatus;

string Message;

}

class PhoneOS {

int ID;

string Name;

}

class AccountStatus {

int ID;

string Name;

}

|  |  |  |
| --- | --- | --- |
| **Элемент** | **Тип значения** | **Значение** |
| ID | числовой | ID пользователя |
| FirstName | строковый | Имя пользователя |
| LastName | строковый | Фамилия пользователя |
| PhoneModel | строковый | Модель телефона |
| PhoneCode | строковый | Код страны (по умолчанию “7”) |
| PhoneNumber | PhoneOS | Номер телефона |
| PhoneUID | строковый | Уникальный идентификатор телефона |
| PhoneOS.ID | числовой | ID ОС телефона |
| AccountStatus.ID | числовой | ID статуса аккаунта:  1 - Не активен  2 - Активен  3 - Активен (заблокирован вызов официантов)  4 - Заблокирован |
| Message | строковый | Сообщения / ошибки сервиса |

**Request XML:**

<FirstName>Иван</FirstName>

<LastName>Петров</LastName>

< PhoneModel>iPhone</PhoneModel>

<PhoneCode>7</PhoneCode>

<PhoneNumber>9151234567</PhoneNumber>  
 <PhoneUID>HGR345765</PhoneUID>  
 <PhoneOS\_ID>1</PhoneOS\_ID>

<Email>com@com.com</Email>

**Response XML:**

<Account>

<ID>123</ID>

<FirstName>Иван</FirstName>

<LastName>Петров</LastName>

<PhoneModel>iPhone 1</PhoneModel>

<PhoneCode>7</PhoneCode>

<PhoneNumber>9151234567</PhoneNumber>  
 <PhoneUID>HGR345765</PhoneUID>  
 <PhoneOS>

<ID>1</ID>

<Name>iOS</Name>

</PhoneOS>

<AccountStatus>

<ID>1</ID>

<Name>Не активен</Name>

</ AccountStatus>

<Message></Message>

</Account>

* 1. **Смена пароля**

/// <summary>

/// Смена пароля

/// </summary>

/// <param name="phoneNumber">Номер телефона</param>

/// <param name=”user\_key">ID сессии </param>

/// <param name="phoneCode">Код страны</param>

int ChangePsw (string phoneNumber, string oldPsw, string newPsw, string user\_key, string phoneCode = “7”)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| phoneNumber | строковый | Номер телефона |
| oldPsw | строковый | Прежний пароль |
| newPsw | строковый | Новый пароль |
| phoneCode | строковый | Код страны (по умолчанию “7”) |
| user\_key | строковый | ID сессии |

**Возвращаемые типы данных**

int result – результат обновления пароля: 1 – пароль обновлен, 0 – ошибка

**Request XML:**

<phoneNumber>9151234567</phoneNumber>

<phoneCode>7</phoneCode>

<oldPsw>12345678</oldPsw>

<newPsw>87654321</newPsw>

<user\_key> dlhj42jkjhfr45ghygdgdl </user\_key>

**Response XML:**

<result>1</result>

* 1. **Выход и завершение сессии**

/// <summary>

/// Выход и завершение сессии

/// </summary>

/// <param name="phoneNumber">Номер телефона</param>

/// <param name=”user\_key">ID сессии </param>

/// <param name="phoneCode">Код страны</param>

int Exit (string phoneNumber, string user\_key, string phoneCode = “7”)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| phoneNumber | строковый | Номер телефона |
| phoneCode | строковый | Код страны (по умолчанию “7”) |
| user\_key | строковый | ID сессии |

**Возвращаемые типы данных**

int result – результат завершения сессии: 1 – сессия завершена, 0 – ошибка

**Request XML:**

<phoneNumber>9151234567</phoneNumber>

<phoneCode>7</phoneCode>

<user\_key> dlhj42jkjhfr45ghygdgdl </user\_key>

**Response XML:**

<result>1</result>

* 1. **Проверка кода SMS**

/// <summary>

/// Проверка кода SMS

/// </summary>

/// <param name="phoneNumber">Номер телефона</param>

/// <param name="SMScode">Введенный код SMS</param>

/// <param name="phoneCode">Код страны</param>

bool CheckSMS (string phoneNumber, string SMScode, string phoneCode = “7”)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| phoneNumber | строковый | Номер телефона |
| phoneCode | строковый | Код страны (по умолчанию “7”) |
| SMScode | строковый | Введенный код SMS |

**Возвращаемые типы данных**

bool result – результат: true – правильный код, false – ошибочный код

Отслеживание количества попыток ввода кода производится в Приложении.

**Request XML:**

<phoneNumber>89032456789</phoneNumber>

<SMScode>1234</SMScode>

<phoneCode>7</phoneCode>

**Response XML:**

<result>true</result>

1. **Карты лояльности**
   1. **Загрузка привязанных карт в Приложение**

/// <summary>

/// Загрузка привязанных карт в Приложение

/// </summary>

/// <param name="phoneNumber">Номер телефона</param>

/// <param name=”user\_key">ID сессии </param>

/// <param name="phoneCode">Код страны</param>

List<DiscountCard> FindDiscountCard(string phoneNumber, string user\_key, string phoneCode = “7”)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| phoneNumber | строковый | Номер телефона |
| phoneCode | строковый | Код страны (по умолчанию “7”) |
| user\_key | строковый | ID сессии |

**Возвращаемые типы данных**

List<DiscountCard> - список привязанных карт

Null – привязанных карт не найдено

class DiscountCard {

int ID // ID карты в системе

int RestaurantID // ID ресторана/сети ресторанов

Int 64 CardNumber // Номер карты

Account Account // Аккаунт пользователя

DateTime InsertDate // Дата привязки карты (полный формат DateTime (2015-07-22 21:18:37.630))

DateTime LastDate // Дата последней проверки карты

int CardStatus // Статус карты: 0 – карта не применялась, 1 – карта применялась

}

class Account {

int ID; // ID аккаунта

}

**Request XML:**

<phoneNumber>9151234567</phoneNumber>

<user\_key>dlhj42jkjhfr45ghygdgdl</user\_key>

<phoneCode>7</phoneCode>

**Response XML:**

<DiscountCard>

<ID>123</ID>

<RestaurantID>15</RestaurantID>

<CardNumber>65679876</CardNumber>

<Account>

<ID>123</ID>

</Account>

<InsertDate>2015-08-25 23:49:41.003</InsertDate>

<LastDate>2015-08-27 23:49:41.003</LastDate>

<CardStatus>1</CardStatus >

</DiscountCard >

<DiscountCard>

<ID>146</ID>

<RestaurantID>19</RestaurantID>

<CardNumber>656776554</CardNumber>

<Account>

<ID>123</ID>

</Account>

<InsertDate>2015-07-25 23:49:41.003</InsertDate>

<LastDate>2015-07-27 23:49:41.003</LastDate>

<CardStatus>1</CardStatus>

</DiscountCard >

* 1. **Привязка дисконтной карты в Системе**

/// <summary>

/// Привязка дисконтной карты в Системе

/// </summary>

/// <param name="phoneNumber">Номер телефона</param>

/// <param name="card">Данные дисконтной карты</param>

/// <param name=”user\_key">ID сессии </param>

/// <param name="phoneCode">Код страны</param>

int InsertDiscountCard(string phoneNumber, DiscountCard card, string user\_key, string phoneCode = “7”)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| phoneNumber | строковый | Номер телефона |
| card | DiscountCard | Дисконтная карта |
| phoneCode | строковый | Код страны (по умолчанию “7”) |
| user\_key | строковый | ID сессии |

**Возвращаемые типы данных**

int DiscountCard.ID – ID привязанной карты в Системе

0 – карта не привязана

class DiscountCard {

int RestaurantID // ID ресторана

Int64 CardNumber // Номер карты

int CardStatus // Статус карты: 0 – карта не применялась, 1 – карта применялась, 2 – требует проверки

}

**Request XML:**

<phoneNumber>9151234567</phoneNumber>

<phoneCode>7</phoneCode>

<DiscountCard>

<RestaurantID>19</RestaurantID>

<CardNumber>656776554</CardNumber>

<CardStatus>1</CardStatus>

</DiscountCard >

<user\_key>dlhj42jkjhfr45ghygdgdl</user\_key>

**Response XML:**

<DiscountCard>

<ID>123</ID>

</DiscountCard >

* 1. **Изменение статуса (удаление) дисконтной карты в Системе**

/// <summary>

/// Изменение статуса (удаление) дисконтной карты в Системе

/// </summary>

/// <param name="phoneNumber">Номер телефона</param>

/// <param name=" cardNumber">Номер карты</param>

/// <param name="status">Статус карты</param>

/// <param name=”user\_key">ID сессии </param>

/// <param name="phoneCode">Код страны</param>

int UpdateDiscountCard(string phoneNumber, int cardNumber, int status, string user\_key, string phoneCode = “7”)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| phoneNumber | строковый | Номер телефона |
| cardNumber | числовой | Номер карты |
| status | числовой | Статус карты |
| phoneCode | строковый | Код страны (по умолчанию “7”) |
| user\_key | строковый | ID сессии |

**Возвращаемые типы данных**

Int64 DiscountCard.ID – номер карты при успешном изменении статуса карты

0 – ошибка при изменении статуса

**Request XML:**

<phoneNumber>9151234567</phoneNumber>

<phoneCode>7</phoneCode>

<CardNumber>656776554</CardNumber>

<CardStatus>1</CardStatus>

<user\_key>dlhj42jkjhfr45ghygdgdl</user\_key>

**Response XML:**

<DiscountCard>

<ID>123</ID>

</DiscountCard >

**Статусы дисконтной карты:**

0 – карта не привязана и не применялась

1 – карта применялась ранее

2 – карта требует проверки

-1 – карта удалена

1. **Заказ**
   1. **Получение информации о заказах стола, привязанных к техническому блюду**

/// <summary>

/// Получение информации о заказах стола, привязанных к техническому блюду

/// </summary>

/// <param name="phoneNumber">Номер телефона</param>

/// <param name="restaurantID">ID ресторана</param>

/// <param name="techItem">техническое блюдо (количество)</param>

/// <param name=”user\_key">ID сессии </param>

/// <param name="phoneCode">Код страны</param>

List<Order> FindOrders(string phoneNumber, int restaurantID, int techItem, string qr, string user\_key, string phoneCode = “7” , int language=0)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| phoneNumber | строковый | Номер телефона |
| restaurantID | числовое | ID ресторана |
| techItem | числовое | Техническое блюдо (количество) |
| qr | строковый | QR код |
| phoneCode | строковый | Код страны (по умолчанию “7”) |
| user\_key | строковый | ID сессии |

**Возвращаемые типы данных**

List<Order> - список заказов

Null – открытые заказы не найдены

class Order {

string OrderNumber // Номер заказа (уникальный для данного ресторана)

int RestaurantID // ID ресторана

string TableID // ID стола

Waiter Waiter // Официант

List<OrderItem> Items // Позиции заказа

DiscountCard DiscountCard // Данные карты лояльности

OrderStatus OrderStatus // Статус заказа

OrderPayment OrderPayment // Стоимость заказа с учетом скидок

string Message // Любое сообщение

int ErrorCode // Код ошибки

string Error // Сообщение об ошибке

string FormUrl // Url, на который переходит приложение при оплате

}

class Waiter

{

int ID // ID официанта

string Name // ФИО официанта

}

class OrderItem

{

string Name // Наименование блюда

decimal Price // Прайсовая стоимость блюда

decimal Qty // Количество

}

class OrderStatus

{

int StatusID // ID статуса: 0 - Открыт, 1 – Пречек, 2 - Закрыт

DateTime StatusDate // Дата/время назначения статуса

}

class DiscountCard

{

Int64 CardNumber // Номер карты

DateTime LastDate // Дата/время текущей проверки карты (полный формат DateTime (2015-07-22 21:18:37.630))

int CardStatus // Статус карты: 0 – карта не применена, 1 – карта применена

}

class OrderPayment

{

decimal OrderTotal // Стоимость заказа с учетом скидок, но без учета чаевых, полученная от ИМ

decimal DiscountSum // Суммарная скидка (для информации клиента), полученная от ИМ

decimal OrderSum // Стоимость заказа к оплате (OrderTotal минус DiscountSum)

Int64 OrderBank // Стоимость заказа к оплате в банк (OrderSum плюс чаевые)

}

**Request XML:**

<phoneNumber>9151234567</phoneNumber>

<phoneCode>7</phoneCode>

<RestaurantID>19</RestaurantID>

<techItem>6</techItem>

<qr>2010010550554</qr>

<user\_key>dlhj42jkjhfr45ghygdgdl</user\_key>

В большинстве запросов возвращается список List<Order>, который может содержать один или несколько объектов Order. В зависимости от сценария в объекте Order заполняются/изменяются те или иные поля. Ниже приведен пример возвращаемого XML.

**Response XML:**

<OrderList>

<Order>

<OrderNumber>12345678</OrderNumber>

<RestaurantID>45</RestaurantID>

<TableID>ABC1</TableID>

<Waiter>

<ID>5</ID>

<Name>Джабраилов Васиссуалий</Name>

</Waiter>

<OrderItems>

<OrderItem>

<Name>Пиво</Name>

<Price>200.00</Price>

<Qty>4</Qty>

</OrderItem>

<OrderItem>

<Name>Гренки</Name>

<Price>100.00</Price>

<Qty>4</Qty>

</OrderItem>

</OrderItems>

<OrderStatus>

<StatusID>0</StatusID>

<StatusDate>2015-08-25 23:49:41.003</StatusDate>

</OrderStatus>

<DiscountCard>

<CardNumber>12345678</CardNumber>

<LastDate>2015-08-25 23:49:41.003</LastDate>

<CardStatus>0</CardStatus>

</DiscountCard>

<OrderPayment>

<OrderTotal>1200.00</OrderTotal>

<DiscountSum>0.00</DiscountSum>

<OrderSum>1200.00</OrderSum>

<OrderBank></OrderBank>

</OrderPayment>

<Message>Welcome</Message>

<ErrorCode></ErrorCode>

<Error></Error>

<FormUrl></FormUrl>

<Order>

</OrderList>

* 1. **Получение информации о заказе по номеру**

/// <summary>

/// Получение информации о заказе по номеру

/// </summary>

/// <param name="restaurantID">ID ресторана</param>

/// <param name="orderNumber">Номер заказа (уникальный для данного ресторана)</param>

/// <param name=”user\_key">ID сессии </param>

List<Order> GetOrder(int restaurantID, string orderNumber, string user\_key, int language=0)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| restaurantID | числовое | ID ресторана |
| orderNumber | строковое | Номер заказа |
| user\_key | строковый | ID сессии |

**Возвращаемые типы данных**

List<Order> – информация о заказе

Null – заказ с данным номером не найден

**Request XML:**

<restaurantID>19</restaurantID>

<orderNumber>150</orderNumber>

<user\_key>dlhj42jkjhfr45ghygdgdl</user\_key>

**Response XML:**

Структура возвращаемого XML аналогична приведенной выше.

* 1. **Вызов официанта**

/// <summary>

/// Вызов официанта

/// </summary>

int CallWaiter (int restaurantID, string tableID, string orderNumber, int code, string user\_key)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| restaurantID | числовое | ID ресторана |
| tableID | строковый | ID стола |
| orderNumber | строковое | Номер заказа |
| code | числовое | Код вызова |
| user\_key | строковый | ID сессии |

**Коды вызова:**

1 – вызов официанта

2 – запрос счета

3 – отмена счета

8 – клиент успешно рассчитался через приложение

**Возвращаемые типы данных**

int CallResult – результат вызова официанта, полученный от интеграционного модуля:

0 – ошибка при вызове/вызов невозможен;

1 - официант проинформирован;

2 - официант не проинформирован.

**Request XML:**

<restaurantID>19</restaurantID>

<tableID>3А</tableID>

<orderNumber>150</orderNumber>

<code>1</code>

<user\_key>dlhj42jkjhfr45ghygdgdl</user\_key>

**Response XML:**

<CallResult>1</CallResult>

* 1. **Начало оплаты заказа**

/// <summary>

/// Начало оплаты заказа

/// </summary>

/// <param name="restaurantID">ID ресторана</param>

/// <param name="orderNumber">Номер заказа (уникальный для данного ресторана)</param>

/// <param name="paymentSum">Суммарная стоимость заказа без учета чаевых</param>

/// <param name=”discountCardNumber">Номер дисконтной карты </param>

/// <param name=”user\_key">ID сессии </param>

List<Order> OrderPrecheck(int restaurantID, string orderNumber, decimal paymentSum, Int64? discountCardNumber, string user\_key, int language=0)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| restaurantID | числовое | ID ресторана |
| orderNumber | строковое | Номер заказа |
| paymentSum | десятичное | Суммарная стоимость заказа к оплате без учета чаевых |
| discountCardNumber | числовое | Номер дисконтной карты |
| user\_key | строковый | ID сессии |

**Возвращаемые типы данных**

List<Order> : При совпадении суммы к оплате в списке возвращается информация об одном заказе с изменением статуса заказа (=1) и датой/временем установки этого статуса. Если заказ оплачен и закрыт, то передается этот статус с датой/временем закрытия заказа

При несовпадении суммы в свойстве Order.Error возвращается ошибка «Сумма заказа не совпадает», в свойстве ErrorCode – код ошибки 20. При этом статус заказа остается прежний (=0). Сам заказ обновляется с новой суммой.

Null – заказ с данным номером не найден

**Request XML:**

<restaurantID>45</restaurantID>

<orderNumber>10</orderNumber>

<discountCardNumber>12345678910</discountCardNumber>

<user\_key>dlhj42jkjhfr45ghygdgdl</user\_key>

**Response XML:**

Структура возвращаемого XML аналогична приведенной выше.

* 1. **Список возможных связок (привязанные банковские карты)**

/// <summary>

/// Список возможных связок

/// </summary>

/// <param name="restaurantID">ID ресторана</param>

/// <param name=”user\_key">ID сессии </param>

List<Binding> GetBindings(string phoneNumber, string user\_key)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| phoneNumber | строковый | Номер телефона |
| user\_key | строковый | ID сессии |

**Возвращаемые типы данных**

List<Binding> – список возможных связок банковских карт.

Null – привязанные карты отсутствуют.

class Binding {

string clientId // = Account.ID

string bindingId // ID связки

string cardName // Название карты, присвоенное пользователем. По умолчанию для вновь привязанной карты: «Название карты»

string maskedPan // Максированный номер карты (напр., 411111\*\*1111) для отображения в Приложении

string expiryDate // Срок истечения действия карты в формате YYYYMM

int active // 1 – карта активна, 0 – карта неактивна

}

* 1. **Редактирование пользовательского названия карты**

/// <summary>

/// Редактирование названия карты

/// </summary>

/// <param name=”phoneNumber"> Номер телефона </param>

/// <param name=”bindingId"> ID связки </param>

/// <param name=”cardName"> Новое название карты, присвоенное пользователем </param>

/// <param name=”user\_key">ID сессии </param>

/// <param name="phoneCode">Код страны</param>

List<Binding> ChangeCardName (string phoneNumber, string bindingId, string cardName, string user\_key, string phoneCode = “7”)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| phoneNumber | строковый | Номер телефона |
| bindingId | строковый | ID связки |
| cardName | строковый | Новое название |
| phoneCode | строковый | Код страны (по умолчанию “7”) |
| user\_key | строковый | ID сессии |

**Возвращаемые типы данных**

List<Binding> – возвращается весь список банковских карт, включая карту с новым названием (аналогично запросу GetBindings).

Null – привязанные карты отсутствуют.

class Binding {

string clientId // = Account.ID

string bindingId // ID связки

string cardName // Название карты, присвоенное пользователем. По умолчанию для вновь привязанной карты: «Название карты»

string maskedPan // Максированный номер карты (напр., 411111\*\*1111) для отображения в Приложении

string expiryDate // Срок истечения действия карты в формате YYYYMM

int active // 1 – карта активна, 0 – карта неактивна

}

* 1. **Отмена начала оплаты заказа**

/// <summary>

/// Отмена начала оплаты заказа

/// </summary>

/// <param name="restaurantID">ID ресторана</param>

/// <param name="orderNumber">Номер заказа (уникальный для данного ресторана)</param>

/// <param name=”discountCardNumber"> Номер дисконтной карты </param>

/// <param name=”user\_key">ID сессии </param>

List<Order> OrderCancelPrecheck(int restaurantID, string orderNumber, Int64? discountCardNumber, string user\_key, int language=0)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| restaurantID | числовое | ID ресторана |
| orderNumber | строковое | Номер заказа |
| discountCardNumber | числовое | Номер дисконтной карты |
| user\_key | строковый | ID сессии |

**Возвращаемые типы данных**

List<Order> – в списке возвращается информация об одном заказе с изменением статуса заказа на «Открыт» и датой/временем установки этого статуса. Если заказ уже оплачен и закрыт, то статус заказа меняется на «Закрыт» с указанием даты/времени его закрытия.

Null – заказ с данным номером не найден

**Request XML:**

<restaurantID>45</restaurantID>

<orderNumber>10</orderNumber>

<discountCardNumber>12345678910</discountCardNumber>

<user\_key>dlhj42jkjhfr45ghygdgdl</user\_key>

**Response XML:**

Структура возвращаемого XML аналогична приведенной выше.

* 1. **Оплата заказа**

Описание взаимодействия:

1. После начала оплаты и получения заказом статуса Precheck пользователь подтверждает оплату заказа.
2. Приложение делает запрос в Основную систему
3. В Основной системе производится проверка совпадения суммы к оплате paymentSum и суммы к оплате в ИМ ресторана.
4. При несовпадении суммы в Приложение возвращается объект List<Order> с новой суммой и записью в свойстве Error=«Сумма заказа не совпадает» и кодом ошибки ErrorCode=20. Приложение обновляет сумму на экране оплаты и пользователь производит оплату снова.
5. При совпадении суммы платежа на запрос Основной системы о регистрации платежа платежный шлюз возвращает URL, на который Приложение должно перенаправить пользователя для получения платежной формы.
6. Основная система в ответе на запрос GetPayment передает Приложению объект List<Order> с заполненным свойством FormUrl = URL, полученный на шаге 5.
7. Приложение открывает URL.
8. В качестве страницы по указанному URL Приложение получает платежную форму.
9. Пользователь заполняет полученную форму и отправляет данные на сервер платежного шлюза.
10. Система проверяет вовлеченности карты в 3DSecure (SecureCode).
11. Шлюз отправляет Приложению redirect на страницу ACS банка эмитента (этот шаг является необходимым для реализации 3DS).
12. Приложение запрашивает у ACS форму авторизации пользователя (у каждого эмитента это делается по-своему).
13. ACS отправляет в Приложение эту форму, клиент ее видит.
14. Клиент заполняет форму и отправляет ее в ACS.
15. ACS обрабатывает заполненную форму и (в не зависимости от результата) передает Приложению redirect на URL страницы платежного шлюза. Вместе с этим URL передаются зашифрованные параметры результата авторизации.
16. Приложение запрашивает страницу платежного шлюза, передавая зашифрованные параметры результата авторизации.
17. Платежный шлюз производит оплату (списание).
18. После проведения оплаты, платежный шлюз передает Приложению URL возврата (указанный ранее при регистрации заказа магазином), напр.:

[http://185.26.113.204/Payment/Result? orderId=7ee6-4f1a-b3d5-6ca41a982283](http://185.26.113.204/Payment/Result?%20orderId=7ee6-4f1a-b3d5-6ca41a982283)

1. Приложение запрашивает у Основной системы по URL возврата результат оплаты.
2. Основная система передает в Приложение html сообщение с результатами оплаты или текстом ошибки.

Если по истечении отведенных на оплату 20 минут клиент не вернулся с платежного шлюза на страницу результатов оплаты (на URL возврата клиента), то оплата считается неудачной.

При платеже по связкам параметр bindingId берется из выпадающего списка привязанных банксовких карт. Данные для списка привязанных банковских карт – из запроса GetBindings (п.3.5).

При проведении платежа по связкам указывается параметр bindingId, соответствующий выбранной в привязанной карте. После проведения оплаты платежный шлюз передает Приложению URL возврата.

/// <summary>

/// Оплата заказа

/// </summary>

/// <param name="restaurantID">ID ресторана</param>

/// <param name="orderNumber">Номер заказа</param>

/// <param name="paymentSum">Суммарная стоимость заказа без учета чаевых</param>

/// <param name="paymentBank">Стоимость заказа к оплате в банк с учетом чаевых, в ***копейках***</param>

/// <param name=”user\_key">ID сессии </param>

/// <param name="bindingId">ID связки</param>

string GetPayment(int restaurantID, string orderNumber, decimal paymentSum, int paymentBank, string user\_key) – т.к. этот метод пока не используется, изменения в него не вносились

List<Order> GetPaymentBinding(int restaurantID, string orderNumber, decimal paymentSum, int paymentBank, string user\_key, string bindingId, int language=0)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| restaurantID | числовое | ID ресторана |
| orderNumber | строковое | Номер заказа |
| paymentSum | десятичное | Суммарная стоимость заказа к оплате без учета чаевых |
| paymentBank | числовое | Стоимость заказа к оплате в банк с учетом чаевых, **в копейках** |
| user\_key | строковый | ID сессии |
| bindingId | строковый | ID связки |

**Возвращаемые типы данных**

List<Order> : При совпадении суммы к оплате в свойстве Order.FormUrl указывается URL, на который надо перенаправить пользователя для получения платежной формы. При несовпадении суммы в свойстве Order.Error возвращается ошибка «Сумма заказа не совпадает», в свойстве ErrorCode – код ошибки 20.

При ошибке платежного шлюза в свойстве Order.Error будет возвращено описание ошибки, полученное от платежной системы.

При отсутствии информации о заказе в БД – Order.Error =“Заказ в БД не найден”.

**Request XML:**

<restaurantID>720920001</restaurantID>

<orderNumber>12345678</orderNumber>

<paymentSum>150.00</paymentSum>

<paymentBank>17000</paymentBank>

<paymentDate>2015-08-25 23:49:41.003</paymentDate>

<user\_key>dlhj42jkjhfr45ghygdgdl</user\_key>

**Response XML:**

<OrderList>

<Order>

……….

<FormUrl>[https://test.paymentgate.ru/testpayment/merchants/rbs/mobile\_payment\_ru.html?mdOrder=05fcbc62-7ee6-4f1a-b3d5-6ca41-a982283</FormUrl](https://test.paymentgate.ru/testpayment/merchants/rbs/mobile_payment_ru.html?mdOrder=05fcbc62-7ee6-4f1a-b3d5-6ca41-a982283%3c/FormUrl)>

<ErrorCode></ErrorCode>

<Error></Error>

</Order>

</OrderList>

* 1. **Предварительная привязка карты**

/// <summary>

/// Оплата заказа

/// </summary>

/// <param name="restaurantID">ID ресторана</param>

/// <param name=”user\_key">ID сессии </param>

string RegisterBinding(int restaurantID, string orderNumber, string user\_key)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| restaurantID | числовое | ID ресторана |
| user\_key | строковый | ID сессии |

**Возвращаемые типы данных**

string – URL, на который надо перенаправить пользователя для получения формы ввода данных карты.

При ошибке будет возвращено описание ошибки, полученное от платежной системы.

После ввода данных карты пользователь перенаправляется на страницу с результатом привязки карты.

**Request XML:**

<restaurantID>720920001</restaurantID>

<orderNumber>12345678</orderNumber>

<user\_key>dlhj42jkjhfr45ghygdgdl</user\_key>

**Response XML:**

<formUrl>https://test.paymentgate.ru/testpayment/merchants/rbs/mobile\_payment\_ru.html?mdOrder=05fcbc62-7ee6-4f1a-b3d5-6ca41-a982283</formUrl>

* 1. **Удаление привязки карты (деактивация связки)**

/// <summary>

/// Удаление привязки карты (деактивация связки)

/// </summary>

/// <param name="bindingId">ID связки</param>

/// <param name=”user\_key">ID сессии </param>

string UnbindCard(string bindingId, string user\_key, int language=0)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| bindingId | строковый | ID связки |
| user\_key | строковый | ID сессии |

**Возвращаемые типы данных**

string – результат деактивации связки.

**Request XML:**

<bindingId> fd3afc57-c6d0-4e08-aaef-1b7cfeb093dc </bindingId>

<user\_key>dlhj42jkjhfr45ghygdgdl</user\_key>

**Response XML:**

<ErrorMessage>Успешно</EroorMessage>

* 1. **Восстановление привязки карты (активация удаленной ранее связки)**

/// <summary>

/// Восстановление привязки карты (активация удаленной ранее связки)

/// </summary>

/// <param name="bindingId">ID связки</param>

/// <param name=”user\_key">ID сессии </param>

string BindCard(string bindingId, string user\_key, int language=0)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| bindingId | строковый | ID связки |
| user\_key | строковый | ID сессии |

**Возвращаемые типы данных**

string – результат активации связки.

**Request XML:**

<bindingId> fd3afc57-c6d0-4e08-aaef-1b7cfeb093dc </bindingId>

<user\_key>dlhj42jkjhfr45ghygdgdl</user\_key>

**Response XML:**

<ErrorMessage>Успешно</EroorMessage>

* 1. **Оценка качества обслуживания в ресторане**

/// <summary>

/// Оценка качества обслуживания в ресторане

/// </summary>

/// <param name="restaurantID">ID ресторана</param>

/// <param name="orderNumber">Номер заказа</param>

/// <param name="rating">Код оценки качества (от 1 до 5)</param>

/// <param name=”user\_key">ID сессии </param>

int SaveRating(int restaurantID, string orderNumber, int rating, string user\_key)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| restaurantID | числовое | ID ресторана |
| orderNumber | строковое | Номер заказа |
| rating | числовое | Код оценки качества ресторана (от 1 до 5) |
| user\_key | строковый | ID сессии |

**Возвращаемые типы данных**

Int result : 1– результат оценки сохранен, 0 – результат оценки не сохранен

**Request XML:**

<restaurantID>45</restaurantID>

<orderNumber>10</orderNumber>

<rating>4</rating>

<user\_key>dlhj42jkjhfr45ghygdgdl</user\_key>

**Response XML:**

<result>1</result>

1. **Информация о ресторане**
   1. **Список ресторанов, подключенных к сети**

/// <summary>

/// Список ресторанов, подключенных к сети

/// </summary>

/// <param name=”user\_key">ID сессии </param>

List<RestNetwork> FindRestaurant(string user\_key, int language=0)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| user\_key | строковый | ID сессии |

**Возвращаемые типы данных**

List<RestNetwork> - список сетей ресторанов

class RestNetwork {

int ID // ID сети ресторанов

string Name // Наименование сети

string Logo // Файл логотипа

string Description // Описание сети

string WWW // Сайт сети

string Image // Изображение для описания

List<Restaurant> Restaurants // Список ресторанов

}

class Restaurant

{

int ID // ID ресторана

string Name // Наименование ресторана

string Logo // Файл логотипа

string Notes // Описание ресторана

string Image // Изображение для описания

string WorkTime // Время работы ресторана

string Address // Адрес ресторана

string Phone // Телефон ресторана

string WWW // Сайт ресторана

string Geocode // Координаты (E134.854,S25.828)

bool Call // Возможность вызова официанта: true – да, false - нет

int Tipping // Возможность начисления чаевых: 1 – да, 0 - нет

int Rating // Рейтинг ресторана

}

**Request XML:**

<user\_key>dlhj42jkjhfr45ghygdgdl</user\_key>

**Response XML:**

<RestNetwork>

<ID>12</ID>

<Name>Грабли</Name>

<Logo>grabli.gif</Logo>

<Notes>Описание сети ресторанов</Notes>

<WWW></WWW>

<Image>grabli.jpg</Image>

<Restaurant>

<ID>1</ID>

<Name>Грабли Семеновская</Name>

<Logo>grabli.gif</Logo>

<Notes>Описание ресторана</Notes>

<Image>grabli.jpg</Image>

<WorkTime>от рассвета до заката</WorkTime>

<Address>Семеновская пл., 1а</Address>

<Phone>(495) 123-45-67</Phone>

<WWW></WWW>

<Geocode>N55.783195,E37.719423</Geocode>

<Call>true</Call>

<Tipping>1</Tipping>

<Rating>3</Rating>

</Restaurant>

</RestNetwork>

<RestNetwork>

<Restaurant>

<ID>34</ID>

<Name>Столовая номер 21</Name>

<Logo></Logo>

<Notes>Описание ресторана</Notes>

<Image>по250идомой.jpg</Image>

<WorkTime>от заката до рассвета</WorkTime>

<Address>там же</Address>

<Phone>(495) 765-43-21</Phone>

<WWW></WWW>

<Geocode>N55.783195,E37.719423</Geocode>

<Call>false</Call>

<Tipping>1</Tipping>

<Rating>3</Rating>

</Restaurant>

</RestNetwork>

* 1. **Запрос информации о ресторане**

/// <summary>

/// Запрос информации о ресторане

/// </summary>

/// <param name="restaurantID">ID ресторана</param>

/// <param name=”user\_key">ID сессии </param>

Restaurant RestaurantInfo(int restaurantID, string user\_key, int language=0)

|  |  |  |
| --- | --- | --- |
| **Параметр** | **Тип значения** | **Значение** |
| restaurantID | числовое | ID ресторана |
| user\_key | строковый | ID сессии |

**Возвращаемые типы данных**

Restaurant – информация о ресторане с данным ID

class Restaurant

{

int ID // ID ресторана

string Name // Наименование ресторана

string Logo // Файл логотипа

string Notes // Описание ресторана

string Image // Изображение для описания

string WorkTime // Время работы ресторана

string Address // Адрес ресторана

string Phone // Телефон ресторана

string WWW // Сайт ресторана

string Geocode // Координаты (E134.854,S25.828)

bool Call // Возможность вызова официанта: true – да, false - нет

int Tipping // Возможность начисления чаевых: 1 – да, 0 - нет

int Rating // Рейтинг ресторана

}

**Request XML:**

<restaurantID>45</restaurantID>

<user\_key>dlhj42jkjhfr45ghygdgdl</user\_key>

**Response XML:**

<Restaurant>

<ID>1</ID>

<Name>Грабли Семеновская</Name>

<Logo>grabli.gif</Logo>

<Notes>Описание ресторана</Notes>

Image>grabli.jpg</Image>

<WorkTime>от рассвета до заката</WorkTime>

<Address>Семеновская пл., 1а</Address>

<Phone>(495) 123-45-67</Phone>

<WWW></WWW>

<Geocode>N55.783195,E37.719423</Geocode>

<Call>true</Call>

<Tipping>1</Tipping>

<Rating>3</Rating>

</Restaurant>