www.rapida.ru

Часть 3. Общие Технические требования.

Раздел 8. Расширение протокола ПП Платежи (Расширенные параметры)

Версия:	012
Статус:	Действующий
Дата:	01.03.2018

Оглавление

История изменений документа

Внимание!

Перед работой с настоящим документом необходимо убедится в его актуальности. Актуальные версии документов размещаются на сайте http://soft.rapida.ru

Версия	Дата	Внесенные изменения
001	05.10.2012	
002	15.10.2012	Изменена структура XML
003	18.10.2012	Исключены избыточные элементы, Внесено описание новых атрибутов, дополнено описание, исправлены опечатки
004	13.12.2012	Добавлены новые типы групповых элементов
005	19.04.2013	Добавлен элемент Action
006	19.05.2013	Внесены незначительные изменения
007	24.07.2013	Внесены незначительные изменения
008	31.10.2013	Добавлен новый пример ответа и зарезервированные имена элементов
009	11.02.2014	Добавлено описание элемента с именем Debt
010	12.03.2014	Исключен элемент с именем Debt. Добавлены информационные атрибуты
011	07.11.2016	Добавлено описание формата вызова getextinfo. Исключено описание не используемой в данном документе XSD Схемы.
012	01.03.2016	Добавлен раздел обработки ошибок

1 Общие сведения

Ряд получателей платежей (ТСП) подразумевает передачу в свой биллинг данных, не укладывающихся в стандартную для Сервиса Рапида схему с 10 параметрами.

Кроме того, многие получатели имеют динамическое распределение параметров, перечень и назначение которых невозможно статически хранить Рапиды или Участника.

В связи с вышеперечисленными фактами предлагается данное расширение протокола ПП Платежи.

2 Изменение формата запросов check и payment

Для вызова функций Участник Системы производит вызов процедур по стандартному протоколу Рапида Протокол ПП Платежи. Расширенный набор данных передается в стандартном наборе значений параметра Params виде одного нового параметра платежа с кодом 0.

На значение параметра накладываются те же самые требования что и на остальные параметры, кроме самого размера.

Отличие этого параметра платежа от остальных то, что ограничение в 255 символов увеличено до 4096 символов.

Пример:

```
Params=11+1581315;53+154333;16+148;17+77;0+Расширенный|длинный|параметр
```

3 Статическое расширение

Расширенный параметр может использоваться для расширения статического набора данных.

В данном случае информация о формате указанного параметра передается в справочнике ТСП [Формат справочника получателей]

Схема данного элемента:

```
<xs:element name="xfield">
  <xs:complexType>
    <xs:choice minOccurs="0">
      <xs:sequence>
        <xs:element ref="reg"/>
        <xs:choice minOccurs="0">
           <xs:sequence>
             <xs:element ref="field" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:choice>
      </xs:sequence>
    </xs:choice>
    <xs:attribute name="name" type="xs:string" use="optional"/>
    <xs:attribute name="code" type="xs:short" use="optional"/>
    <xs:attribute name="delimiter" type="xs:string" use="required"/>
  </xs:complexType>
</xs:element>
```

Согласно указанной схеме, передается статически набор расширенных параметров, согласно которым будет заполняться соответствующий подпараметр параметра Params в запросах check и payment.

Пример:

```
<field name="Номер колоды" code="777">
  <reg>^\d{1,2}$</reg>
</field>
<xfield name="Расширенное поле" code="0" delimiter="|">
  <field name="Вес" code="1">
    <reg>^\d{1,2}$</reg>
  </field>
  <field name="Масть" code="2">
    <reg>^(0|1|2|3)$</reg>
    <def value="0" note="Пики"/>
    <def value="1" note="Трефы"/>
    <def value="2" note="Бубны"/>
    <def value="3" note="Червы"/>
  </field>
</xfield>
```

Заполнения параметра Params для указанного примера:

```
Params=777+2;0+5|3
```

4 Динамическое расширение

Расширенный параметр может использоваться для расширения динамического набора данных.

В этом случае элемент xfield не содержит описания внутренних элементов.

Но сам элемент tsp содержит информацию о применимому к нему сценарию и набору минимальных данных для запроса getextinfo в атрибуте getbal.

Пример:

```
<tsp name="ЖКХ Мегаквартиры" code="1234" getbal="5,777">
<field name="Номер лицевого счета" code="777">
  <reg>^\d{1,2}$</reg>
</field>
<xfield name="Расширенное поле" code="0"/>
```

Данный пример указывает, что клиентскому приложению Участника системы необходимо вызвать запрос GetExtInfo передав ему на обработку данные параметра 777 (номер лицевого счета) после чего обработать результат согласно сценарию 5, и из обработанных данных сформировать расширенное поле.

Для вызова функции Участник Системы производит вызов процедуры с 3-я параметрами:

№	Описание параметра	Параметр	Формат
1	Идентификатор запроса	Function	Константа = «getextinfo»
2	Код ТСП в ПС «Рапида»	PaymSubjTp	(3-5)-значный цифровой код получателя в ПС «Рапида»
3	Коды и значения параметров платежа указанные в атрибуте getbal	Params	Строка с параметрами платежа в специальном формате (п.1.3).

Пример вызова:

```
/?function=getextinfo&PaymSubjTp=<код услуги>&Params=[параметры: <код параметра>
<значение параметра>]
```

5 Правило обработки/визуализации информации, возвращаемой на запросе getextinfo

Предлагаемая XML схеме ответа сценария 5 для запроса GetExtInfo приведена в Приложении.

После получения ответа на запрос GetExtInfo, Участник должен подготовить соответствующие формы для заполнения их дополнительными данными согласно описанным в данном разделе правилам.

5.1 Атрибуты элементов

5.1.1 Атрибуты общего назначения

Любой из элементов может содержать следующие атрибуты:

- **input** = (integer) Является ли элемент информационными или требуется формирование поля для его ввода. Неприменимо для type = "group|check|choice".
- **show** = (integer) Определяет необходимость визуализировать данный элемент в интерфейсе.
  Допустимые значения:
  - show="0" – скрытое поле
  - show="1" – Визуализация поля описываемого элементом рекомендована но не обязательна
  - show="2" – Визуализация поля описываемого элементом обязательна

  Значение по умолчанию ="2"
- **title** = (string) Текстовая подпись поля
- **hint** = (string) Текстовая подсказка для поля
- **return** = (integer) признак наличия элемента в check/payment
  - return="0" – не предназначен для check/payment
  - return="1" – предназначен для check/payment в случае непустого значения
  - return="2" – предназначен для check/payment даже в случае пустого
  - return="3" – предназначен для check/payment, пустое значение не допустимо
- **type** = (string) тип поля.
  Допустимые значения:
  - type = "money" сумма в копейках
  - type = "string" строковое поле
  - type = "digital" цифровое поле
  - type = "date" дата в формате YYYY-MM-DD
  - type = "time" время в формате HH:mm
  - type = "boolean" логическое значение "true" или "false"
  - type = "list" поле список (выбор значения).
  - type = "group|check|choice" элемент - группа.

  Значение по умолчанию = "string"
- **payment** = атрибут изоляции платежа для группы choice (см описание в п. **Ошибка! Источник ссылки не найден.**.)
  Допустимые значения:
  - payment ="0" – группа не имеет выраженной изоляции для единого check/payment.
  - payment ="1" – группа является изоляцией для единого check/payment.
- **reg** = (string) регулярное выражение для проверки корректности веденных данных . Применимо только совместно с input="1".

Значение по умолчанию = ".*"

- **min** = минимальное значение (число, длина строки, минимально допустимая дата) в зависимости от type. Применимо только совместно с input="1".
- **max** = максимальное значение (число, длина строки, максимально допустимая дата) в зависимости от type. Применимо только совместно с input="1"..
- **value** = (string) Значение поля по умолчанию. Применимо только совместно с input="1".

Указанные атрибуты являются необязательными.

Значения атрибутов по умолчанию, если отдельно не указано в описании атрибута:

- для числовых атрибутов 0,
- для строковых атрибутов – пустая строка

5.1.2 Атрибуты специального назначения

Атрибуты специального назначения являются обязательными для обработки Рапида на шаге check/payment (далее платежный шаг).

5.1.2.1 Атрибут id (Уникальный идентификатор)

Атрибут id определяет уникальный идентификатор элемента и предназначен для обработки Рапида на платежном шаге.

Для обработки клиентским приложением не несет никакой смысловой нагрузки.

5.1.2.2 Атрибут name (функциональный идентификатор)

Атрибут name определяет функциональный идентификатор элемента.

Данный атрибут является регистр независимым, то есть name="name" и name="Name" не различаются.

Существуют следующие специализированные идентификаторы: amount, item, tariff, lastindication, newindication, их обработка описана ниже.

В дальнейшем по тексту элементы Field указанными в абзаце выше атрибутами так и будут называться "Элемент с именем …."

Значение атрибута name не специфицированное данным документом не несут для клиентсого приложения смысловой нагрузки.

5.1.2.3 Атрибут code (Идентификатор для параметра params)

Атрибут code определяет уникальный идентификатор элемента и предназначен для обработки Рапида на платежном шаге как один из элементов параметра Params базового протокола.

5.1.3 Атрибуты информационного назначения

Атрибуты информационного назначения предназначены для необязательной обработки на стороне Участника и не предназначены для последующей обработки Рапида на платежном шаге.

5.1.3.1 Атрибут debt (задолженность по получателю)

Атрибут debt применим исключительно для элемента Subscriber и определяет текущую задолженность Плательщика по указанному Получателю.

Отсутствие атрибута указывает на отсутствии информации о задолженности, но не об ее отсутствии.

Указанное значение является информационным и не участвует в последующей обработке платежа.

Данные могут быть использованы в упрощенной процедуре обработке информационного ответа, чтобы донести до клиента результирующую сумму по задолженности.

Примеры:

```
<Subscriber show="2" return="2" name="468318" debt="0" charge="1464000">
<Subscriber show="2" return="2" name="468318" debt="1034" charge="1464000">
<Subscriber show="2" return="2" name="468318" charge="1464000">
<Subscriber show="2" return="2" name="468318" debt="1034">
```

Тип данных атрибута поля всегда "money" (сумма в копейках)

5.1.3.2 Атрибут charge (начисление по получателю)

Атрибут charge применим исключительно для элемента Subscriber и определяет выставленное начисление Плательщику по указанному Получателю. Если это применимо.

Отсутствие атрибута указывает на отсутствии информации о начислении, но не об его отсутствии.

Указанное значение является информационным и не участвует в последующей обработке платежа.

Данные могут быть использованы в упрощенной процедуре обработке информационного ответа, чтобы донести до клиента результирующую сумму по начислениям.

Тип данных атрибута поля всегда "money" (сумма в копейках)

Примеры приведены в предыдущем пункте.

5.2Элемент Subscriber

В случае, если запрос содержит несколько элементов Subscriber, клиенту предлагается для платежа выбрать только один из них. Например, по одному адресу может проживать несколько жильцов. Участник должен предложить первичный выбор соответствующего Subscriber, если их количество более 1.

Элемент Subscriber по своему определению подразумевает атрибут общего назначения type="group", и поэтому им может не сопровождаться.

Каждый из элементов Subscriber может содержать несколько элементов Field с именами Item, Counter и Service (Элементов учета), общих элементов Field и специализированных элементов Field. (см. далее)

5.3Элемент Action

Если групповой элемент запрос содержит элемент **Action**, это определяет, какой запрос должен последовать после обработки текущего.

- Значение равное "loop" означает, что по результатам обработки текущего запроса исполнение платежного шага недоступно. Участнику необходимо выполнить дополнительный - уточняющий вызов getextinfo с учетом обработки результатов текущего запроса.

- Значение равное "next" означает, что по результатам обработки текущего запроса требуется вызов запросов платежного шага
- Значение равное "any" означает, что по результатам обработки текущего запроса уже возможно исполнение запросов платежного шага, но на усмотрение Участника (Плательщика) возможен вызов уточняющего детали запроса getextinfo.

Отсутствие элемента **Action** равносильно его значению "next"

5.4Элементы учета

Элементом учета называется элемент с именем Item, Counter или Service.

Каждый из элементов учета может содержать несколько вложенных элементов учета , общих элементов Field, и одну группу специализированных элементов.

По каждому из элементов Item необходимо формирование отдельной группы полей для ввода.

Элемент учета по своему определению подразумевает атрибут общего назначения type="group", и поэтому им может не сопровождаться.

5.5Элемент Field общего назначения

Элементы общего назначения Field – это элементы, не попадающие под описание пп. 5.4 и 5.8.

Данные элементы описывают дополнительные поля, которые необходимо запросить у клиента по данному подписчику и/или элементу учета или визуализировать их содержание в информационных целях.

Любой элемент Field может содержать вложенные элементы Field. В этом случае он обязательно имеет атрибут type="group", это обозначает, что по возможности необходимо визуализировать вложенные в него элементы-поля в отдельно выделенной группе.

Пример:

```
<Field name="Fio" title="Абонент" type="group">
<Field name="message" title="сообщение" input="0" value="Уважаемый абонент компании
XXX. Напоминаем, что в этом месяце состоится общее собрание." show="1" />
<Field name="firstName" title="Имя" input="1" show="2" />
<Field name="lastName" title="Фамилия" input="1" show="2" />
<Field name="citizen" title="Гражданство" input="1" value="РФ" show="2" />
</Field>
```

Сообщает, что необходимо визуализировать именованный блок из двух полей для ввода.

5.6Элемент Field со списочным значением (type="list")

Элементы общего назначения Field имеющие атрибут type="list" определяют поле имеющее возможность выбора значения из списка. Для таких Элементов доступные значения передаются во вложенных элементах Value

Вложенный элемент Value может иметь атрибут общего назначения title

Примеры:

```
<Field name="User" title="Пол" type="list">
  <Value>Мужской</Value>
  <Value>Женский</Value>
  <Value>Не имеет значения</Value>
</Field>
Список из текстовых значений (значение в отображаемом списке равно его содержимому)

<Field name="User" title="Пол" type="list" value="0">
```

```
<Value tittle="Мужской">1</Value>
<Value tittle="Женский">2</Value>
<Value tittle="Не имеет значения">0</Value>
</Field>
```

Список из именованных значений (значение в отображаемом списке не равно его содержимому)

5.7 Группы элементов Field (type = "group|check|choice")

Элемент с одним из этих типов сам по себе не является полем, а имеет вложенные элементы – поля.

Возможны три варианта подобных видов групп. Каждый из них отдельно описан ниже.

5.7.1 Набор взаимоисключающих групп (type = "choice")

Элемент с типом "choice" является определителем того, что сам по себе не является полем ввода но содержит в себе другие элементы Field. Но вне зависимости от значения атрибутов return или show элементов следующего уровня, обработан (выбран) может быть только один из них.

Пример:

```
<Field name="Service" title="Оплачиваемая услуга" return="0" type="choice" value="1">
   <Field name="S1" id="1" title="Газ" return="3" type="group">
…
   </Field>
   <Field name="S2" id="3" title="Электричество" return="3" type="group">
…
   </Field>
   <Field name="S2" id="67" title="Телефон" return="3" type="group">
…
   </Field>
</Field>
```

В этом случае клиенту предлагается оплатить только одну из предложенных услуг.

Интерфейсно клиенту сначала предлагается список услуг, а потому уже ввод значений в выбранном элементе, если они имеют место.

Невыбранные элементы не отображаются и в дальнейшей обработке участия не принимают вне зависимости от значений атрибутов return и show.

5.7.1.1 Платежные группы (payment = "1")

Элемент с типом "choice" может определять группу полей предназначенных для изолированных check/payment (платежных шагов). Например, содержать перечень услуг, которые могут быть оплачены только по отдельности независимыми запросами платежных шагов, что и предопределяет их взаимоисключение.

То есть по результатам обработки этих элементов должны быть произведены независимые вызовы запросов платежных шагов по каждому из элементов группы. Атрибут payment = "1" определяет что данная группа относится именно к этому типу.

Пример:

```
<Field name="Service" title="Оплачиваемая услуга" return="0" type="choice"
payment="1" value="1" >
   <Field name="S1" id="1" title="Газ" return="3" type="group">
…
   </Field>
```

```
   <Field name="S2" id="3" title="Электричество" return="3" type="group">
…
   </Field>
   <Field name="S2" id="67" title="Телефон" return="3" type="group">
…
   </Field>
</Field>
```

В этом случае есть два варианта реализации интерфейса.

1. Аналогично описанному в п. 5.7.1., то есть для каждого элемента группы формируется расширенный параметр, и вызывается соответствующий платежный шаг, после чего требуется повторный вызов getextinfo, для формирования расширенного параметра по очередной услуге и вызов очередного платежного шага.
2. Предложить клиенту оплатить несколько из предложенных услуг, по каждой из них сформировать отдельный расширенный элемент. Каждая из услуг все равно потребует отдельной пары платежных шагов. Но перед вызовом каждого отдельного платежного шага можно воспользоваться результатами обработки единого сценария getextinfo.

В случае если в интерфейсе Участника это не востребовано, то указанный тип должен восприниматься как обычный type = "choice".

5.7.2 Набор взаимо-неисключающих групп (type = "check")

Элемент с типом "check" является определителем того что сам по себе не является полем ввода но содержит в себе другие элементы Field. Но, вне зависимости от значения атрибутов return или show элементов следующего уровня, обработано (выбрано) может быть любое ненулевое количество элементов следующего уровня.

Пример:

```
<Field name="Service" title="Оплачиваемая услуга" return="0" type="check">
   <Field name="S1" title="Газ" return="2" type="group">
…
   </Field>
   <Field name="S2" title="Электричество" return="2" type="group">
…
   </Field>
   <Field name="S2" title="Телефон" return="2" type="group">
…
   </Field>
</Field>
```

В этом случае клиенту предлагается оплатить любое количество из предложенных услуг.

Интерфейсно клиенту сначала предлагается список услуг, а потому уже ввод значений в выбранных элементах. Выбрано может быть любое ненулевое количество вложенных элементов.

Невыбранные элементы, не отображаются и в дальнейшей обработке участия не принимают вне зависимости от значений атрибутов return и show.

5.7.3 Простой набор вложенных элементов (type = "group")

Элемент с типом "group" является определителем того что сам по себе не является полем ввода но содержит в себе другие элементы Field. Каждый из вложенных элементов подлежит обработке в зависимости от его атрибутов.

Пример:

```
<Field name="Service" title="Оплачиваемая услуга" return="0" type="group">
   <Field name="S1" title="Газ" return="1" type="group">
…
   </Field>
   <Field name="S2" title="Электричество" return="1" type="group">
…
   </Field>
   <Field name="S2" title="Телефон" return="1" type="group">
…
   </Field>
</Field>
```

В этом случае клиенту предлагается заполнить информацию каждому из трех видов услуг на свое усмотрение. Поля вложенных и элементов визуализируются и участвуют в обработке в зависимости от значений своих атрибутов

5.8 Специализированные элементы с именами Amount, Tariff и *Indications

Элемент учета может содержать группу специализированных элементов с именами Amount, Tariff, *Indications.

Элемент с именем Amount определяет сумму платежа по данному элементу учета и присутствует в единственном числе.

В случае если имеются элементы именами *Indications сопровождаемые элементом с именем Tariff, то при изменении значений в полях определяемых LastIndication и NewIndication, значение элемента с именем Amount должно автоматически пересчитываться по формуле:

Amount=(NewIndication-LastIndication)*Tariff

Значения элемента с именем Amount вышестоящих элементов с тем же именем так же должно автоматически высчитываться как сумма нижестоящих элементов Amount.

В случае отсутствия в группе элементов учета элемента с именем Amount, остальные элементы специального назначения этой группы расцениваются как элементы общего назначения.

Примеры:

Элемент учета, сопровождаемый группой специализированных элементов (Элементом и именем Amount сопровождаемый счётчиками и тарифом)

```
<Field name="Item" id="123" title="электричество" >
  <Field name="Amount" return="1" min="1000" input="0" show="1"/>
  <Field name="tariff" return="0" input="0" value="2300" />
  <Field name="lastindication" return="0" input="0" value="3534534" />
  <Field name="newindication" return="0" min="3534534" input="1"/>
</Field>
```

Элемент учета, сопровождаемый независимым элементом с именем Amount

```
<Field name="Item" id="123" title="электричество" >
  <Field name="Amount" return="1" min="1000" input="1" show="1"/>
</Field>
```

Элемент учета, сопровождаемый независимым элементом счетчика с именем NewIndication, в качестве элемента общего нгазначения.

```
<Field name="Item" id="123" title="электричество" >
      <Field name="newindication" return="0" min="3534534" input="1"/>
</Field>
```

Элемент учета со вложенными элементами учета Amount

```
<Field name="Item" id="123" title="электричество" >
  <Field name="Amount" return="2" min="1000" input="0" show="0"/>
    <Field name="Counter" id="123" title="день">
      <Field name="Amount" return="1" min="1000" input="1" show="1"/>
      <Field name="tariff" return="0" input="0" value="2300" />
      <Field name="lastindication" return="0" input="0" value="3534534" />
      <Field name="newindication" return="0" min="3534534" input="1"/>
    </Field>
    <Field name="Counter" id="123" title="ночь" >
      <Field name="Amount" return="1" min="1000" input="1" show="1"/>
      <Field name="tariff" return="0" input="0" value="2300" />
      <Field name="lastindication" return="0" input="0" value="3534534" />
      <Field name="newindication" return="0" min="3534534" input="1"/>
    </Field>
</Field>
```

Общая сумма к оплате по каждой точке учёта должна быть равна сумме сумм по всем элементам. Все суммы передаются в копейках.

5.9 Примеры возможных комбинаций атрибутов общего назначения

```
<Field id="135" input="1" show="1" input="1" title="Номер счетчика"
reg="\d{10}"><Field>
```

Сообщает, что по возможности надо визуализировать изменяемое поле с именем "Номер счетчика". И проверить введенные данные регулярным выражением "\d{10}".

```
<Field input="1" show="2" title="Номер счетчика" reg="\d{10}">1234567890<Field>
```

Сообщает, что строго необходимо визуализировать изменяемое поле с именем "Номер счетчика". И проверить введенные данные регулярным выражением "\d{10}". Но заранее заполнив его значением 1234567890

```
<Field show="2" title="Номер счетчика">1234567890<Field>
```

Сообщает, что строго необходимо визуализировать информационное поле с именем "Номер счетчика" и значением 1234567890.

5.10 Стандартизированные значения имен элементов

Кроме описанных выше специализированных имен, система так же стандартизовала имена для облегчения построения визуального интерфейса

```
<Field name="Fio">
```

Фамилия Имя Отчество (разделенные пробелом)

```
<Field name="FirstName">
```

Имя

```
<Field name="MiddleName">
```

Отчество

```
<Field name="LastName">
```

Фамилия

```
<Field name="Address">
```

Адрес

```
<Field name="Area">
```

Регион

<Field name="Street">
Улица
<Field name="House">
Номер дома
<Field name="Flat">
Номер квартиры

6 Правило формирования расширенного параметра для запросов check и payment, или уточняющего вызова getextinfo

По результатам введения данных в поля сформированные в предыдущем пункте, необходимо сформировать XML по следующему правилу.

1. Сохранить оригинальную структуру XML для выбранного элемента **Subscriber**.
2. Исключить из нее все элементы у которых атрибут **return="0".**
3. Исключить из нее все невыбранные элементы элементов с **type = "check|choice".**
4. Заполнить все элементы обновленными значениями.
5. Исключить из элементов атрибуты общего назначения.
6. В случае если элемент содержал атрибуты **id** и **name**. Данные атрибуты должны быть сохранены без изменения.
7. Исключить из XML все символы переноса строк и незначащие пробелы и символы табуляции.

Произвести URL кодирование в кодировке win-1251.

7 Примеры ответа getextinfo

Пример 1:

```
<Response>
<Result>OK</Result>
<Description>Абонент найден: Иванов И.И.</Description>
<ErrCode>0</ErrCode>
<Info>
  <Name>Gkh</Name>
  <PID>127919473216</PID>
  <Date>2010-10-10 10:10:10</Date>
</Info>
<Data>
  <Subscriber id="12" return="1">
    <Field name="Fio" title="Абонент" return="0" type="group">
      <Field name="firstName" title="Имя" input="0" show="1" return="0"
value="Иванов" />
      <Field name="lastName" title="Фамилия" input="0" show="1" return="0"
value="Иван" />
    </Field>
    <Field name="Debt" title="Общая сумма счета" show="2" value="12400">
    <Field code="10" title="Книга учета" input="1" show="2" return="3" type="digital"
reg="\d{2}" />
    <Field name="Balance" input="0" show="2" return="0" title="Задолженность"
value="100.00" />
    <Field name="Service" id="11" title="Электричество" show="2" return="1">
      <Field name="Amount" max="5000" min="10" show="2" input="0" return="1"/>
      <Field name="Counter" id="200" title="Счетчик День" return="1" show="2">
        <Field return="1" name="Amount" />
        <Field name="Tariff" show="1" tittle="Тариф 2.50 р/квч" value="250" />
        <Field name="LastIndication" input="0" title="Показание на 2012-08-13"
return="1" value="11111" />
        <Field name="NewIndication" input="1" title="Новое показание" return="1"
type="digital" min="11111" />
      </Field>
      <Field name="Counter" id="300" title="Счетчик Ночь" show="2">
        <Field return="1" name="Amount" />
        <Tariff show="1" tittle="Тариф 1.00 р/квч" value="100" />
        <Field name="LastIndication" input="0" title="Показание на 2012-08-13"
return="1" value="12334" />
        <Field name="NewIndication" input="1" title="Новое показание" return="1"
type="digital" min="12334" />
      </Field>
    </Field>
  </Subscriber>
</Data>
</Response>
```

Пример 2:

```
<?xml version="1.0" encoding="windows-1251"?>
<Response>
  <Result>OK</Result>
  <Description>Запрос успешно исполнен.</Description>
  <ErrCode>0</ErrCode>
  <Info>
```

```
    <Name>rostelecom </Name>
    <PID>138320209611</PID>
  </Info>
  <Data>
    <Subscriber id="138320209669" return="0">
      <Field name="payeeName" type="string" show="2" return="0" title="Инициалы
абонента" value="I.IVAN IVANOVICH"/>
      <Field name="payeeRemain" type="money" show="2" return="0" title="Остаток на
лицевом счете получателя" value="100"/>
      <Field name="payeeRecPay" type="money" show="2" return="0" title="Рекомендуемый
платеж" value="050"/>
      <Field type="group" name="services" id="1" return="0" show="2">
        <Field name="service" type="group" show="1" return="0">
          <Field name="srv_id" show="2" return="0" value="1" title="Номер субсчета"
type="string"/>
          <Field name="srv_name" show="2" return="0" value="Ростелеком (ЮТК) местная
и внутризоновая телефонная связь" title="Название субсчета" type="string"/>
          <Field name="Amount" show="2" return="0" value="100" title="Остаток"
type="money"/>
        </Field>
        <Field name="service" type="group" show="1" return="0">
          <Field name="srv_id" show="2" return="0" value="2" title="Номер субсчета"
type="string"/>
          <Field name="srv_name" show="2" return="0" value="Ростелеком МГ/МН
предвыбор" title="Название субсчета" type="string"/>
          <Field name="Amount" show="2" return="0" value="300" title="Остаток"
type="money"/>
        </Field>
      </Field>
    </Subscriber>
  </Data>
</Response>
```

8 Обработка ошибок

В случае если запрос обработан с ошибкой, то он сопровождается элементом Result равным значению Error. Детальная причина описана в элементе Description.

При получении подобного ответа сценарий необходимо прервать с соответствующим сообщением.

Так же возможно прерывание сценария и при ответах с Result равным значению ОК, в случае если элемент Data не содержит значимых элементов. Например если запрос обработался успешно, но у клиента нет никаких начислений для оплаты.

Для детального анализа причин прерывания сценария Элемент (Result ="Error"), ответ в дополнении к элементу Result, может быть сопровожден элементом ErrCode, содержащим цифровой код ответа.

Наличие элемента ErrCode является опциональным и зависит от конкретного поставщика услуги, предоставляющего информацию о начислениях.

Обработка данного элемента рекомендована Участнику. Она позволяет более гибко информировать клиента о причинах негативного завершения сценария. Но в связи с опциональностью самого элемента в ответе, указанный функционал не является обязательным.

Перечень возможных значений элемента ErrCode

Result	ErrCode	Описание
ОК	0	Список начислений по предоставленным реквизитам или документам формирован успешно
ОК	-1	Конкретное начисление уже было оплачено
ОК	-2	Конкретное начисление не оплачено, но сквитировано по другим причинам и не подлежит оплате
ОК	-3	Список начислений по предоставленным документам не найден
Error	1	Проблемы с регистрацией или установленными разрешениями для Участника, либо срабатывание ограничений безопасности, установленных Участником
Error	4	Система не смогла обработать запрос, возможен неверный формат запроса.
Error	9	Временная проблема с обработкой запроса на стороне ИС
Error	13	От биллинга получателя получен некорректный ответ
Error	14	От биллинга получателя получена ошибка, Подробности в элементе Description.
Error	15	Биллинг получателя временно недоступен
Error	60	Передан некорректный расширенный параметр
Error	61	Перечень необходимых для получения начислений реквизитов недостаточен

9 Заполненый расширеный параметр

Без учета пп.6. и 7 правила.

```
<Subscriber id="12">
<Field code="10">50</Field>
<Field name="Service" id="11">
  <Field name="Amount>35000</Field>
    <Field name="Counter" id="100">
      <Field name="Amount">25000</Field>
      <Field name="LastIndication">0</Field>
      <Field name="NewIndication">100</Field>
    </Field>
    <Field name="Counter" id="300">
      <Field name="Amount">10000</Field>
      <Field name="LastIndication">200</Field>
      <Field name="NewIndication">300</Field>
    </Field>
  </Field>
</Subscriber>
```

Приложение 2 (XML Схемы)

GetExtInfo

```
<?xml version="1.0" encoding="windows-1251"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="CommonType" mixed="true">
    <xs:attribute name="id" type="xs:string" use="optional"/>
    <xs:attribute name="return" use="optional" default="0">
      <xs:simpleType>
        <xs:restriction base="xs:integer">
          <xs:pattern value="0|1|2|3"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:anyAttribute namespace="##any"/>
  </xs:complexType>
  <xs:complexType name="ActionType" mixed="true">
    <xs:complexContent mixed="true">
      <xs:extension base="CommonType">
        <xs:attribute name="value" use="required">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:pattern value="next|loop|any"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="SubscriberType" mixed="true">
    <xs:complexContent mixed="true">
      <xs:extension base="CommonType">
        <xs:sequence>
          <xs:element name="Field" type="FieldType" minOccurs="0" maxOccurs="unbounded"/>
          <xs:element name="Action" type="ActionType" minOccurs="0"/>
        </xs:sequence>
        <xs:attribute name="code" type="xs:integer" use="optional"/>
```

```
      <xs:attribute name="title" type="xs:string" use="optional"/>
      <xs:attribute name="hint" type="xs:string" use="optional"/>
      <xs:attribute name="input" use="optional" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:pattern value="0|1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="show" use="optional" default="0">
        <xs:simpleType>
          <xs:restriction base="xs:integer">
            <xs:pattern value="0|1|2"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:anyAttribute namespace="##any"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="FieldType" mixed="true">
  <xs:complexContent mixed="true">
    <xs:extension base="SubscriberType">
      <xs:sequence>
        <xs:element name="Value" type="ValueType" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="name" type="xs:string" use="required"/>
      <xs:attribute name="reg" type="xs:string" use="optional"/>
      <xs:attribute name="min" type="xs:anySimpleType" use="optional"/>
      <xs:attribute name="max" type="xs:anySimpleType" use="optional"/>
      <xs:attribute name="value" type="xs:anySimpleType" use="optional"/>
      <xs:attribute name="type" use="optional">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:pattern value="money|string|digital|date|list|group|check|choice"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:extension>
```

```
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="ValueType" mixed="true">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="title" type="xs:string" use="optional"/>
      <xs:anyAttribute namespace="##any"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
<xs:element name="Response">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Result" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="Description" type="xs:string" minOccurs="1" maxOccurs="1"/>
      <xs:element name="ErrCode" type="xs:intrger" minOccurs="0" maxOccurs="1"/>
      <xs:element name="Info">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Name" type="xs:string"/>
            <xs:element name="PID" type="xs:string"/>
            <xs:element name="Date" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Data" minOccurs="0" maxOccurs="1">
        <xs:complexType>
          <xs:choice>
            <xs:element ref="Subscriber" minOccurs="0" maxOccurs="unbounded"/>
          </xs:choice>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Subscriber">
  <xs:complexType mixed="false">
    <xs:complexContent mixed="false">
      <xs:extension base="SubscriberType"/>
```

```
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
</xs:schema>
```