



www.rapida.ru

Часть 3. Общие Технические требования

Раздел 5. Формат справочника получателей

Код документа: EXT.2.03.03.00005.004
Версия: 004
Статус: Действующий
Дата: 01.06.2017

Оглавление

История изменений документа	2
ОБЩИЕ СВЕДЕНИЯ	3
ОПИСАНИЕ ФОРМАТА XML ФАЙЛА	3
Блоки категорий. Элемент PART	3
Блоки получателей. Элемент TSP	3
Блоки параметров получателей. Элемент FIELD	4
Составные поля FIELD	5
Примечания	5
Приложения:	7
Пример XML справочника	7
Схема XML справочника:	9

История изменений документа

Версия	Дата	Внесенные изменения
001	25.03.2010	Начальная версия
002	26.05.2010	Добавлен раздел "Примечания"
003	15.05.2010	Добавлено описание составных полей
004	01.06.2017	Внесены правки в соответствии с реорганизацией 28 апреля 2017 КИВИ Банк (АО) в форме присоединения к нему ООО НКО «Рапида». Изменена структура и нумерация документа

ОБЩИЕ СВЕДЕНИЯ

КИВИ Банк (АО) в рамках взаимодействия с платежными и служебными сервисами Рапиды по запросу (описанному в документе *Протокол приема платежей по продукту "Платежи"*) формирует XML файл со справочником получателей их параметров необходимых для проведения запросов на платежи.

ОПИСАНИЕ ФОРМАТА XML ФАЙЛА

Главный элемент справочника – **rapida** сопровождаемый атрибутом **version**, указывающим версию дату и время документа.
<rapida version="20100315 17:29:43:170">

Блоки категорий. Элемент PART.

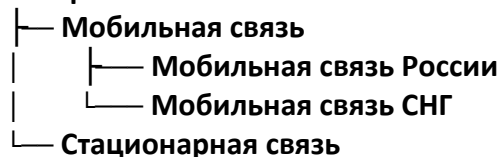
Справочник содержит в себе описание дерева категорий получателей.

Каждый узел дерева категорий получателей определяется элементом **part**, элементы **part** допускают неограниченную вложенность.

Элемент **part** сопровождается обязательным атрибутом **name** определяющим название категории.

Подобное дерево категорий получателей:

Телефония



Описывается следующей структурой:

```
<rapida version="20100315 17:29:43:170">
<part name="Телефония">
  . . .
  <part name="Мобильная связь">
    <part name="Мобильная связь России">
      . . .
    </part>
    <part name="Мобильная связь СНГ">
      . . .
    </part>
  </part>
  <part name="Стационарная связь">
    . . .
  </part>
  . . .
</part>
</rapida>
```

Блоки получателей. Элемент TSP.

Каждый элемент **part** описывающий категорию получателей, может содержать любое количество элементов **tsp** описывающих самих получателей, принадлежащих этой категории.

Элемент **tsp** сопровождается атрибутами:

- **name** (обязательный)- Название получателя;
- **code** (обязательный)- Код получателя (ТСП)
- **dest** (необязательный)- Название услуги;
- **note** (необязательный)- Альтернативное название получателя, или другое дополнительное описание относящееся к получателю;
- **state** (необязательный) признак справочника для альтернативного продукта.

Например:

```
<tsp name="МГТС" dest="абонент. плата и оплата внутрizonовых соединений"  
code="400">  
</tsp>
```

Блоки параметров получателей. Элемент FIELD.

Каждый элемент **tsp** описывающий получателя, содержит необходимое количество элементов **field** описывающих параметры платежа и данного получателя.

Элемент **field** сопровождается атрибутами:

- **name** (обязательный)- Название параметра;
- **code** (обязательный)- Код параметра;
- **state** (необязательный)- Наличие не равного 0 атрибута, указывает на его не обязательность при формировании платежного запроса.

Элемент **field** может содержать следующие неповторяющиеся элементы:

- **mask** (необязательный)- Маска для автоматизации ввода параметра;
- **reg** (необязательный)- Регулярное выражение для проверки параметра;
- **hint** (необязательный)- Текстовое описание, подсказка.

Например:

```
<field name="Номер мобильного телефона" code="188">  
  <mask>!\ (999\) 000-00-00;0;_</mask>  
  <reg>^[0-9]{10}$</reg>  
  <hint>Введите десятизначный номер телефона</hint>  
</field>
```

Элемент **field** может содержать повторяющийся элемент **def**, указывающий, что клиенту предлагается выбрать один из возможных вариантов. Одиночный тег **def** описывает значение параметра "по умолчанию".

Элемент **def** сопровождается атрибутами:

- **value** (обязательный)- значение варианта выбора, или значения по умолчанию;
- **note** (необязательный)- название значения варианта выбора.

Например:

```
<field name="Код вида платежа (1-2 цифры)" code="592">
  <reg>^\d{1,2}$</reg>
  <hint>1: ЮТК - услуги местной телефонной связи\n2: ЮТК - услуги
Ростелеком (МГ\МН связь)\n3: ЮТК - услуги МТТ (МГ\МН связь)\n4:
Голдентелеком\n5: Эквант\n33: Транстелеком</hint>
  <def value="1" note="ЮТК - услуги местной телефонной связи"/>
  <def value="2" note="ЮТК - услуги Ростелеком (МГ\МН связь)"/>
  <def value="3" note="ЮТК - услуги МТТ (МГ\МН связь)"/>
  <def value="4" note="Голдентелеком"/>
  <def value="5" note="Эквант"/>
  <def value="33" note="Транстелеком"/>
</field>
```

Составные поля FIELD

Количество элементов **field** не может превышать 10. Поэтому у отдельных ТСП элемент **field** может содержать вложенные элементы **field**.

В этом случае поле является составным, состоящим из субполей. Требование к каждому из которых, указано во вложенном элементе **field**. Элемент **code** первого вложенного элемента **field** всегда совпадает с соответствующим элементом основного поля. В последующих субполях элемент **code** определяет порядок следования конкретного субполя в основном составном поле.

Например:

```
<field name="Карта" code="123">
  <reg>^\d{1,2}, (0|1|2|3)$</reg>
  <field name="Вес" code="123">
    <reg>^\d{1,2}$</reg>
  </field>
  <field name="Масть" code="124">
    <reg>^(0|1|2|3)$</reg>
    <def value="0" note="Пики"/>
    <def value="1" note="Трефы"/>
    <def value="0" note="Бубны"/>
    <def value="1" note="Червы"/>
  </field>
</field>
```

При формировании основного поля, в качестве разделителя составных полей выступает символ «,» (запятая). Например субполе «Вес» установлено в 6 а поле «Масть» в 1, то составное поле, согласно пункту 1.3. документа «Раздел 3. Протокол приема платежей по продукту "Платежи" », будет выглядеть следующим образом:

123+6,1

Примечания

Элемент **tsp**, содержащий атрибут **state** отличный от нуля определяет получателя не доступного для платежей.

Элемент **tsp**, содержащий элемент **field** с атрибутом **name** равным константе "#", или не содержащий вложенных элементов, определяет получателя не доступного для платежей.

Любой элемент может содержать атрибут и/или вложенный элемент, не описанный настоящим документом. В этом случае он должен игнорироваться. Или его функциональное назначение может определяться отдельным документом.

Приложения:

Пример XML справочника

```
<?xml version="1.0" encoding="windows-1251" ?>
<rapida version="20100315 17:29:43:170">
  <part name="Телефония" img="3">
    <part name="Междугородная телефонная связь" img="3">
      <tsp name="Ростелеком Москва" dest="оплата междугородной/международной связи по счету" code="316">
        <field name="Номер счета" code="16">
          <reg>^\d{11}$</reg>
        </field>
        <field name="Дата выставления счета" code="53">
          <reg>^\d{6}$</reg>
          <hint>в формате ДДММГГ</hint>
        </field>
        <field name="Тип счета" code="293">
          <reg>^\d{3}$</reg>
        </field>
        <field name="Номер телефона" code="17">
          <reg>^\d{12}$</reg>
          <hint>Указан в счете</hint>
        </field>
      </tsp>
    </part>
  <part name="Мобильная связь" img="7">
    <tsp name="ОАО &quot;Мобильные ТелеСистемы&quot;" dest="МТС Единая сеть приема платежей" code="115">
      <field name="Номер мобильного телефона" code="307">
        <reg>^[0-9]{10}$</reg>
      </field>
    </tsp>
    <tsp name="БИЛЛАЙН" note="ОАО &quot;ВымпелКом&quot;" code="101">
      <field name="Номер мобильного телефона" code="188">
        <mask>! \ (999 \ ) 000 -00 -00 ; 0 ; _ </mask>
        <reg>^[0-9]{10}$</reg>
      </field>
    </tsp>
  </part>
</rapida>
```



```
</tsp>  
<tsp name="Мегафон" note="ОАО &quot;Мегафон&quot;" code="111">  
  <field name="Номер мобильного телефона" code="150">  
    <mask>!\\(999\\)000-00-00;0;_</mask>  
    <reg>^[0-9]{10}$</reg>  
  </field>  
</tsp>  
</part>  
</part>  
</rapida>
```

Схема XML справочника:

```
<?xml version="1.0" encoding="windows-1251"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="rapida">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="part" maxOccurs="unbounded"/>
        <xs:element ref="tsp"/>
      </xs:sequence>
      <xs:attribute name="version" use="required" type="xs:string"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="part">
    <xs:annotation>
      <xs:documentation>Элемент описывает ветвь дерева получателей</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:choice>
        <xs:element ref="tsp" maxOccurs="unbounded"/>
        <xs:sequence>
          <xs:element ref="part" maxOccurs="unbounded"/>
          <xs:element ref="tsp" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:choice>
      <xs:attribute name="name" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string"/>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="img">
        <xs:simpleType>
          <xs:restriction base="xs:string"/>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>
```

```
<xs:element name="tsp">
  <xs:annotation>
    <xs:documentation>Элемент описывает получателя</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="field" maxOccurs="10"/>
      <xs:element ref="xfield" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="type">
      <xs:simpleType>
        <xs:restriction base="xs:byte">
          <xs:enumeration value="1"/>
          <xs:enumeration value="2"/>
          <xs:enumeration value="3"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="state" type="xs:byte"/>
    <xs:attribute name="note" type="xs:string"/>
    <xs:attribute name="name" type="xs:string" use="required"/>
    <xs:attribute name="getbal" type="xs:string"/>
    <xs:attribute name="dest" type="xs:string"/>
    <xs:attribute name="code" type="xs:int" use="required"/>
    <xs:attribute name="bik" type="xs:string"/>
    <xs:anyAttribute namespace="##any"/>
    <xs:anyAttribute />
  </xs:complexType>
</xs:element>
<xs:element name="field">
  <xs:annotation>
    <xs:documentation>Элемент описывает параметр получателя</xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:choice minOccurs="0">
      <xs:element ref="def"/>
      <xs:element ref="hint"/>
      <xs:sequence>
```

```
<xs:element ref="reg"/>
<xs:choice minOccurs="0">
  <xs:element ref="def" maxOccurs="unbounded"/>
  <xs:sequence>
    <xs:element ref="hint"/>
    <xs:choice minOccurs="0">
      <xs:element ref="def" maxOccurs="unbounded"/>
      <xs:element ref="field" maxOccurs="unbounded"/>
    </xs:choice>
  </xs:sequence>
</xs:choice>
</xs:sequence>
<xs:sequence>
  <xs:element ref="mask"/>
  <xs:sequence minOccurs="0">
    <xs:element ref="reg"/>
    <xs:element ref="hint" minOccurs="0"/>
  </xs:sequence>
</xs:sequence>
</xs:choice>
<xs:attribute name="state">
  <xs:simpleType>
    <xs:restriction base="xs:byte">
      <xs:enumeration value="1"/>
      <xs:enumeration value="4"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="name" type="xs:string" use="required"/>
<xs:attribute name="code" use="required">
  <xs:simpleType>
    <xs:restriction base="xs:short"/>
  </xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>
<xs:element name="reg" type="xs:string"/>
<xs:element name="mask" type="xs:string"/>
```

```
<xs:element name="hint" type="xs:string"/>
<xs:element name="def">
  <xs:complexType>
    <xs:attribute name="value" type="xs:string" use="required"/>
    <xs:attribute name="note" type="xs:string" use="optional"/>
    <xs:anyAttribute namespace="##any"/>
  </xs:complexType>
</xs:element>
<xs:element name="xfield">
  <xs:annotation>
    <xs:documentation>Элемент описывает расширенный параметр получателя</xs:documentation>
  </xs:annotation>
  <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:anyAttribute namespace="##any"/>
  </xs:complexType>
</xs:element>
</xs:schema>
```