СИСТЕМА МЕЖВЕДОМСТВЕННОГО

ЭЛЕКТРОННОГО ВЗАИМОДЕЙСТВИЯ

ПРИЛОЖЕНИЕ №

к протоколу заседания подкомиссии

по использованию информационных технологий

при предоставлении государственных и муниципальных услуг

Правительственной комиссии

по использованию информационных технологий

для улучшения качества жизни и условий ведения

предпринимательской деятельности

от 22 ноября 2017 г. № 540пр

ОДОБРЕНО

подкомиссией по использованию информационных технологий

при предоставлении государственных и муниципальных услуг

Правительственной комиссии

по использованию информационных технологий

для улучшения качества жизни и условий ведения

предпринимательской деятельности
(протокол от 22 ноября 2017 г. №540пр)

**Требования к XML-схемам, регистрируемым в СМЭВ**



2017

# Содержание

[Содержание 2](#_Toc446427281)

[1. Введение 4](#_Toc446427282)

[1.1. Назначение документа 4](#_Toc446427283)

[1.2. Термины и определения 4](#_Toc446427284)

[2. Требования к использованию конструкций языка XSD 5](#_Toc446427285)

[2.1. Указание целевого пространства имен в схемах 5](#_Toc446427286)

[2.2. Использование атрибута mixed content 5](#_Toc446427287)

[2.3. Использование кириллицы 6](#_Toc446427288)

[2.4. Конструкции xs:any и xs:anyAttribute 6](#_Toc446427289)

[2.5. Тип anyType 7](#_Toc446427290)

[2.6. Объявление элементов без указания типа 7](#_Toc446427291)

[2.7. Конструкция xs:list 8](#_Toc446427292)

[2.8. Форма элементов 8](#_Toc446427293)

[2.9. Конструкция xs:redefine 9](#_Toc446427294)

[2.10. Кодировка XML-схем 9](#_Toc446427295)

[2.11. Запрет на использование спец.символов в атрибутах схем 9](#_Toc446427296)

[3. Ограничения, не относящиеся к конструкциям языка XSD 10](#_Toc446427297)

[3.1. Уникальность пространств имен 10](#_Toc446427298)

[3.2. Пространства имен запроса и ответа 10](#_Toc446427299)

[3.3. Назначение версий схем 11](#_Toc446427300)

[3.4. Транслитерация и перевод в наименованиях 12](#_Toc446427301)

[3.5. Правила именования элементов и атрибутов 14](#_Toc446427302)

[3.6. Структурирование данных 17](#_Toc446427303)

[3.7. Встраивание XML-фрагментов как строк 19](#_Toc446427304)

[3.8. Структурные зависимости в схемах 22](#_Toc446427305)

[3.9. Тип xs:date 24](#_Toc446427306)

[3.10. Конструкция CDATA 24](#_Toc446427307)

[3.11. Элементы верхнего уровня 25](#_Toc446427308)

[3.12. Одинаковые декларации в нескольких схемах 26](#_Toc446427309)

[3.13. Схемы, не отражающие логическую структуру данных 27](#_Toc446427310)

[3.14. Передача в СМЭВ элементов оформления 28](#_Toc446427311)

[3.15. Передача блока структурированных сведений как вложения 29](#_Toc446427312)

[3.16. Вложения 29](#_Toc446427313)

[4. Стиль написания XML схем. Рекомендации 31](#_Toc446427314)

[4.1. Использование типа xs:string 31](#_Toc446427315)

[4.2. Определение простых типов 31](#_Toc446427316)

[4.3. Значения по умолчанию в XSD 31](#_Toc446427317)

[4.4. Избыточная вложенность 31](#_Toc446427318)

[4.5. Избыточная информация в наименованиях элементов 33](#_Toc446427319)

[5. Шаблоны типовых данных и наборов данных 35](#_Toc446427320)

[5.1. Данные физического лица 35](#_Toc446427321)

[5.2. Типы для вложений 47](#_Toc446427322)

[5.3. Реквизиты юридических, физических лиц и ИП 48](#_Toc446427323)

[5.4. Контакты 51](#_Toc446427324)

[5.5. ОКАТО. Тип OKATOType 52](#_Toc446427325)

[5.6. Кадастровый номер. Тип CadastralNumberType 52](#_Toc446427326)

[5.7. Серия свидетельства о рождении, паспорта СССР. Тип SovietStyleDocumentSeriesType 52](#_Toc446427327)

[5.8. Строковые типы 53](#_Toc446427328)

# Введение

## Назначение документа

Настоящий документ описывает требования и рекомендации по разработке XML-схем, регистрируемых в системе межведомственного электронного взаимодействия (далее – СМЭВ). В рамках документа рассматриваются следующие вопросы:

1. Обеспечение однозначности трактовки XML-схем. Использование ряда конструкций языка XSD существенно повышает неопределенность структуры контента; такие конструкции выводятся из употребления в СМЭВ пунктами 2.4, 2.5, 2.6 настоящего документа.
2. Обеспечение возможности контроля качества XML-документов (XML validation), с использованием стандартных технологий обработки XML-документов, и имеющихся промышленных решений. Проблемы в данной области порождаются, главным образом, неграмотной политикой назначения пространств имен (пункты 2.1, 3.1, 3.3), искусственным выводом контента из-под действия валидаторов (пункты 3.6, 3.7, 3.10, 0), а также написанием схем, не в полной мере отражающих логическую структуру данных (пункты 0, 0).
3. Упрощение анализа XML-схем, публикуемых в СМЭВ, потребителями данных; упрощение написания кода, обрабатывающего документы. Удобство для потребителя данных является одним из приоритетов в настоящем документе, поэтому целый ряд пунктов настоящего документа направлен на обеспечение данного требования: 2.1, 2.2, 2.7, 2.8, 3.2, 3.4, 0, 3.9, 3.11, 3.12 и другие.
4. Обеспечение возможности автоматической генерации программного кода на основе XML-схем. Пункты 2.2, 2.3, 2.9 устраняют ситуации, в которых автогенерация кода невозможна, либо затруднительна.

## Термины и определения

В документе используются следующие термины и определения:

|  |  |
| --- | --- |
| target namespace | XSD-атрибут, используемый при написании XML-схем для указания целевого пространства имен |
| XML | eXtensible Markup Language — расширяемый язык разметки  |
| XSD | (XML Schema definition) – язык описания структуры XML-документа |
| URI | Unique Resource Identificator – уникальный идентификатор ресурса |
| Оператор СМЭВ  | Министерство связи и массовых коммуникаций Российской Федерации (в соответствии с постановлением Правительства РФ N 697 от 08.09.2010)  |
| СМЭВ  | Система межведомственного электронного взаимодействия  |
| ОИВ  | Органы исполнительной власти  |

# Требования к использованию конструкций языка XSD

## Указание целевого пространства имен в схемах

Схемы без указания «target namespace» запрещены.

Неправильно:

<xs:schema
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 elementFormDefault="qualified"
 attributeFormDefault="unqualified"
 >

 <xs:complexType name="pIdentityPersonDocumentType">
 ...

Правильно:

<xs:schema
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:tns="urn://x-artefacts-some-oiv-ru/smev-interaction/smth/1.0.0"
 **targetNamespace="urn://x-artefacts-some-oiv-ru/smev-interaction/smth/1.0.0"**
 elementFormDefault="qualified"
 attributeFormDefault="unqualified"
 >
 <xs:complexType name="pIdentityPersonDocumentType">
 ...

## Использование атрибута mixed content

Запрещено использовать mixed content (<http://www.w3.org/TR/2006/REC-xml11-20060816/#sec-mixed-content>). Для структурированных типов разрешен только element content (<http://www.w3.org/TR/2006/REC-xml11-20060816/#sec-element-content>).

Неправильно:

<xs:complexType **mixed="true"**>
 ...
</xs:complexType>

Пример XML-фрагмента с mixed content:

<ns1:Agreement>Заявитель, <ns2:FamilyName>Ростовский</ns2:FamilyName><ns2:FirstName>Иван</ns2:FirstName><ns2:Patronymic>Семенович</ns2:Patronymic>, подписал согласие на обработку персональных данных</ns1:Agreement>

## Использование кириллицы

Рекомендуется использовать символы кириллицы в XML-схемах в аннотациях и в комментариях, а также для задания значений перечислимых типов и регулярных выражений.

Использование кириллических имен элементов и атрибутов возможно в исключительных случаях при наличии корневого элемента схемы в латинице.

Неправильно:

<xs:element name="**ЗапросВыпискиИзКакогоТоРеестра**">
 <xs:complexType>
 ...
 </xs:complexType>
</xs:element>

Правильно:

<xs:element name="**SomeRegistryExсerptionRequest**">
 <xs:complexType>
 ...
 </xs:complexType>
</xs:element>

Правильно:

<xs:simpleType name="string-rus-50">
 <xs:annotation>
 <xs:documentation>
 Русский текст, длиной до 50 символов.
 </xs:documentation>
 </xs:annotation>
 <xs:restriction base="xs:string">
 <xs:maxLength value="50"/>

 <xs:pattern value="**[а-яА-ЯеЁ\-\s]+**"/>
 </xs:restriction>
</xs:simpleType>

## Конструкции xs:any и xs:anyAttribute

Запрещено использование конструкций «xs:any» и «xs:anyAttribute» без указания целевого пространства имен.

Неправильно:

<xs:complexType name="SomeType">
 <xs:sequence>
 **<xs:any/>** ...
 </xs:sequence>
</xs:complexType>

При указании целевого пространства имен запрещено использование атрибута «noNamespace».

Неправильно:

<xs:complexType name="SomeType">
 <xs:sequence>
 <xs:any **noNamespace="##targetNamespace"**/>
 ...
 </xs:sequence>
</xs:complexType>

При указании целевого пространства имен в атрибуте «namespace» запрещено использование значений «##other», «##local».

Неправильно:

<xs:complexType name="SomeType">
 <xs:sequence>
 <xs:any **namespace="##other"**/>
 ...
 </xs:sequence>
</xs:complexType>

Правильно:

<xs:complexType name="SomeType">
 <xs:sequence>
 <xs:any **namespace="http://my.company.ru/something"**/>
 ...
 </xs:sequence>
</xs:complexType>

## Тип anyType

Использование встроенного типа «xs:anyType» запрещено.

## Объявление элементов без указания типа

Согласно спецификации <http://www.w3.org/TR/2012/REC-xmlschema11-1-20120405/>, если для элемента явно не указан тип, то по умолчанию он имеет тип xs:anyType. Поскольку использование типа xs:anyType запрещено настоящим документом, объявлять элементы без указания типа также не разрешается.

Неправильно:

<xs:element name="SomeNotice"/>

Правильно:

<xs:element name="SomeNotice" **type="tns:string-100"**/>

## Конструкция xs:list

Использование конструкции «xs:list» запрещено. Она должна быть заменена на «complex type».

Неправильно:

<xs:simpleType name="SomeListType">
 <**xs:list** itemType="xs:int"/>
</xs:simpleType>

Правильно:

<xs:complexType name="SomeListType">
 <**xs:sequence**>
 <**xs:element** name="item" type="xs:int" maxOccurs="unbounded"/>
 </xs:sequence>
</xs:complexType>

## Форма элементов

Запрещено объявление элементов в «unqualified»-форме.

Неправильно:

<xs:schema
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:tns="http://my.company.ru/something"
 targetNamespace="http://my.company.ru/something"
 elementFormDefault="**unqualified**"
 >

Неправильно (значение по умолчанию для атрибута «elementFormDefault –"unqualified"»):

<xs:schema
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:tns="http://my.company.ru/something"
 targetNamespace="http://my.company.ru/something"
 >

Правильно:

<xs:schema
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:tns="http://my.company.ru/something"
 targetNamespace="http://my.company.ru/something"
 elementFormDefault="**qualified**"
 >

## Конструкция xs:redefine

В спецификации XML schema v.1.1 конструкция xs:redefine объявлена не рекомендованной к использованию (deprecated). Не гарантируется, что она будет поддерживаться в следующих версиях спецификации. По этой причине, а также ввиду того, что эта конструкция не даёт никаких значимых выгод, при этом провоцируя плохой стиль написания схем, её использование запрещено.

## Кодировка XML-схем

Для регистрации в СМЭВ XML-схемы должны предоставляться в кодировке
UTF-8.

## Запрет на использование спец.символов в атрибутах схем

В атрибутах элементов схемы запрещается использовать символы переноса строк и переноса каретки:

line feed |\#xd;|&#13;| \r|

carriage return|&#xa;|&#10;| \n|

# Ограничения, не относящиеся к конструкциям языка XSD

## Уникальность пространств имен

Целевое пространство имен (target namespace) любой схемы, используемой в СМЭВ, должно быть глобально уникально.

Чтобы облегчить соблюдение этого требования, в СМЭВ каждому ОИВ – поставщику данных должен присваиваться базовый URI. Все схемы, регистрируемые в СМЭВ этим поставщиком данных, должны иметь target namespace, начинающиеся с базового URI этого поставщика. Таким образом, ответственность за уникальность базовых URI несет оператор СМЭВ, а поставщик данных отвечает за уникальность target namespace в области действия своего базового URI.

## Пространства имен запроса и ответа

В схемах СМЭВ для каждого вида запрашиваемых сведений объявляется корневой элемент для документа – запроса, и корневой элемент для документа – ответа. Настоящий документ требует, чтобы корневой элемент запроса и корневой элемент ответа одного типа сведений находились в одном пространстве имен.

Неправильно:

<xs:schema
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:tns="urn://x-artifacts-some-foiv-ru/**requests**/1.0.0"
 targetNamespace="urn://x-artifacts-some-foiv-ru/**requests**/1.0.0"
 elementFormDefault="qualified"
 >

 <xs:element name="SomeRegistryExerptionRequest">
 ...

<xs:schema
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:tns="urn://x-artifacts-some-foiv-ru/cumulis/**responses**/1.0.0"
 targetNamespace="urn://x-artifacts-some-foiv-ru/**responses**/1.0.0"
 elementFormDefault="qualified"
 >

 <xs:element name="SomeRegistryExerption">
 ...

Правильно:

<xs:schema
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:tns="urn://x-artifacts-some-foiv-ru/**smev-interaction**/1.0.0"
 targetNamespace="urn://x-artifacts-some-foiv-ru/**smev-interaction**/1.0.0"
 elementFormDefault="qualified"
 >

 <xs:element name="SomeRegistryExerptionRequest">
 ...

 <xs:element name="SomeRegistryExerption">
 ...

При необходимости разделения схемы в силу ее сложности и/или объемности, рекомендуется сделать схему с одним «target namespace» из нескольких XML-документов, с использованием операции «xs:include».

## Назначение версий схем

В XML-схемах, регистрируемых в СМЭВ, версия схемы должна быть отражена в ее «target namespace». Версия должна иметь формат X.Y.Z, где:

* X – меняется при значительных изменениях в структуре и содержании схемы.
* Y – меняется при незначительной переработке структуры и содержания схемы, при стилистических изменениях.
* Z – меняется при незначительных стилистических изменениях, исправлении опечаток и/или грамматических ошибок.

Запрещено использовать какие-либо другие механизмы для обозначения версий.

Неправильно:

<xs:schema
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns="http://ws.some.vendor/some-foiv/sendRequest"
 targetNamespace="http://ws.some.vendor/some-foiv/sendRequest"
 elementFormDefault="qualified"
 >
 <xs:element name="Document">
 <xs:complexType>
 ...
 **<xs:attribute name="VersionForm"** use="required">
 <xs:simpleType>
 <xs:restriction base="xs:string">
 <xs:minLength value="1"/>
 <xs:maxLength value="5"/>
 <xs:enumeration value="4.01"/>
 </xs:restriction>
 </xs:simpleType>
 </xs:attribute>
 ...

Правильно:

<xs:schema
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns="http://ws.some.vendor/some-foiv/types/**4.0.1**"
 targetNamespace="http://ws.some.vendor/some-foiv/types/**4.0.1**"
 elementFormDefault="qualified"
 >
 <xs:element name="HeadcountRequest">
 <xs:complexType>
 ...

## Транслитерация и перевод в наименованиях

При формировании наименования рекомендуется осуществлять подбор слова или словосочетания из английского языка, соответствующего тому или иному используемому понятию. Все слова в наименовании метаданных рекомендуется использовать полностью, без сокращений.

Наименования, обозначающие общепринятые аббревиатуры, имеющие повсеместное хождение, подлежат транслитерации на латиницу (например, ИНН, КПП, СНИЛС). В остальных случаях при выборе наименований элементов и атрибутов запрещена транслитерация русских слов и словосочетаний. Лишь в исключительных случаях, если в английском языке отсутствует слово или словосочетание, достаточно однозначно определяющее описываемое понятие или допускающее большое количество вариантов обратного перевода, допустимо использовать транслитерацию на латинский алфавит.

Порядок записи слов в наименовании, в которых используется два или более слова, должен соответствовать правилам английского языка. Слова должны записываться подряд, без пробела и других знаков между ними.

Наименования метаданных должны записываться строчными буквами, кроме аббревиатур, записываемых полностью прописными (заглавными) буквами. Если используется два или более слова, то каждое слово должно начинаться с прописной (заглавной) буквы. Допустимо также написание первого слова строчными буквами.

В наименования простых и составных типов (simpleType, complexType) для обозначения их отличия от элементов (element) рекомендуется добавлять суффикс «Type»

Неправильно (транслитерация):

ZaprosVipiskyIsReestra

Неправильно (нарушен порядок слов):

RequestExсerptionRegistry

Правильно:

RegistryExсerptionRequest

Неправильно (не использованы общепринятые сокращения):

SocialSecurityNumber, PhysicalPersonTaxpayerID

Правильно:

SNILS, physicalPersonINN

Правильно:

OGRNType, pIdentityPersonDocumentType

## Правила именования элементов и атрибутов

Имя элемента или атрибута должно отражать смысл, который вкладывается в соответствующее понятие предметной области.

Неправильно:

<xs:schema
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:tns="http://somefoiv.gov.ru/requests/1.0.0"
 targetNamespace="http://somefoiv.gov.ru/requests/1.0.0" elementFormDefault="qualified"
 attributeFormDefault="unqualified">

 <xs:element **name="Document"**>
 <xs:annotation>
 <xs:documentation>
 Запрос по сведениям о среднесписочной численности работников
 </xs:documentation>
 </xs:annotation>

Правильно:

<xs:schema
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:tns ="http://somefoiv.gov.ru/requests/1.0.0"
 targetNamespace="http://somefoiv.gov.ru/requests/1.0.0" elementFormDefault="qualified"
 attributeFormDefault="unqualified">

 <xs:element **name="AverageHeadcountRequest"**>
 <xs:annotation>
 <xs:documentation>
 Запрос по сведениям о среднесписочной численности работников
 </xs:documentation>
 </xs:annotation>

Запрещены объявления элементов, в которых смысл отражен значением атрибута или вложенного элемента.

Неправильно:

<xs:element name="DataItem" type="tns:DataItemType"/>

<xs:complexType **name="DataItemType"**>
 <xs:simpleContent>
 <xs:extension base="xs:string">
<!—
Здесь смысл элемента выражен не его именем,
а значением атрибута, придуманного специально для этой цели.
-->
 **<xs:attribute name="meaning" type="tns:MeaningType"/>**
 </xs:extension>
 </xs:simpleContent>
</xs:complexType>

<xs:simpleType name="MeaningType">
 <xs:restriction base="xs:string">
 <xs:enumeration value="FamilyName"/>
 <xs:enumeration value="FirstName"/>
 <xs:enumeration value="Partonymic"/>
 </xs:restriction>
</xs:simpleType>

Соответствующий XML-фрагмент:

<ns1:DataItem meaning="FamilyName">Петров</ns1:DataItem>

Правильно:

<xs:element **name="FamilyName"** type="tns:string-rus-50"/>

<xs:simpleType name="string-rus-50">
 <xs:restriction base="xs:string">
 <xs:maxLength value="50"/>
 <xs:pattern value="[а-яА-ЯеЕ\-\s]+"/>
 </xs:restriction>
</xs:simpleType>

Соответствующий XML-фрагмент:

<ns1:FamilyName>Петров</ns1:FamilyName>

Запрещено использовать в качестве имен элементов числовые коды. Также рекомендуется не использовать аббревиатуры, специфичные для определенного ведомства.

Неправильно:

<xs:complexType **name="UD"**> <!-- Не общепринятая аббревиатура -->
 <xs:annotation>
 <xs:documentation>Установочные данные</xs:documentation>
 </xs:annotation>
 <xs:sequence>
 <xs:element **name="F100"** type="tns:string-rus-50"> <!-- Числовой код -->
 <xs:annotation>
 <xs:documentation>Фамилия</xs:documentation>
 </xs:annotation>
 </xs:element>
 <xs:element **name="F101"** type="tns:string-rus-50"> <!-- Числовой код -->
 <xs:annotation>
 <xs:documentation>Имя</xs:documentation>
 </xs:annotation>
 </xs:element>
 ...

Правильно (аннотации опущены):

<xs:complexType **name="PhysicalPersonBasicData"**>
 <xs:sequence>
 <xs:element **name="FamilyName"** type="tns:string-rus-50"/>
 <xs:element **name="FirstName"** type="xs:string-rus-50"/>
 ...

## Структурирование данных

Запрещено структурировать данные не XML средствами. Пример неправильной реализации:

<ResponseDocument xmlns="http://finstat.ru/gosuslugi/RegisterSignResponse">
<Code>0</Code>
<Value>**Наименование=Закрытое акционерное общество "Джон Смит и внуки"#ОКПО=00011111#ОКОПФ=11#ОКФС=11#ОКВЭД=11.11.1#ИНН=1234567890#Код единицы измерения=123#11104=0#11204=0#11304=0#11354=0#11404=0#11454=0#11504=0#11904=0#12104=0#12114=0#12124=0#12134=0#12144=0#12154=0#12164=0#12174=0#12204=0#12304=0#12314=0#12404=0#12414=0#12504=0#12604=0#12704=0#12904=0#13004=0#14104=0#14114=0#14204=0#14254=0#14304=0#14314=0#14324=0#14334=0#14704=0#14804=0#14904=0#15104=0#15154=0#15204=0#15904=0#16104=0#16204=0#16214=0#16224=0#16234=0#16244=0#16254=0#16304=0#16404=0#16504=0#16604=0#16904=0#17004=0#20103=0#20203=0#20293=0#20303=0#20403=0#20503=0#20603=0#20703=0#20803=0#20903=0#21003=0#21403=0#21413=0#21423=0#21503=0#21903=0#22003=0#31003=0#31005=0#31105=0#31213=0#31223=0#31233=0#31253=0#31255=0#31313=0#31323=0#31333=0#31353=0#31355=0#31403=0#31405=0#32004=0#32103=0#32105=0#32203=0#32205=0#41503=0#41603=0#41703=0#53104=0#53106=0#53204=0#54104=0#54106=0#57103=0#57203=0#57303=0#57403=0#57503=0#57603=0#57653=0#57663=0#57673=0#59103=0#59204=0#59206=0#62603=0#63603=0**</Value>
</ResponseDocument>

Это ответ на запрос о бухгалтерском балансе юридического лица. Баланс представлен в одном строковом поле «Value» в формате с символом-разделителем. Настоящим документом такое структурирование запрещено; логическая структура данных должна быть выражена XML-схемой.

Таким образом, в XML-схеме простые типы могут быть использованы только для атомарных элементов данных.

В контексте межведомственного обмена, элемент данных является атомарным, если выполняется любое из условий:

1. Элемент данных появляется в запросе, и не существует бизнес-процесса оказания государственной услуги или выполнения государственной функции, в котором значение этого элемента данных невозможно получить из первичных документов без вычислений.
2. Элемент данных появляется в ответе на запрос, и не существует бизнес-процесса оказания государственной услуги или выполнения государственной функции, в котором значение этого элемента данных требует дальнейшего разбора.

Неправильно:

<xs:element name="ApplicantID" **type="xs:string"**>
 <xs:annotation>
 **<xs:documentation>
 Идентификатор заявителя формируется путем конкатенации фамилии,
 имени, отчества, даты рождения в формате yyyy-mm-dd, места рождения.
 </xs:documentation>** </xs:annotation>
</xs:element>

Правильно:

<xs:element name="ApplicantID" **type="tns:PersonIdType"**/>

<xs:complexType name="PersonIdType">
 <xs:sequence>
 <xs:element name="FamilyName" type="tns:string-rus-50"/>
 <xs:element name="FirstName" type="tns:string-rus-50"/>
 <xs:element name="Partonymic" type="tns:string-rus-50"/>
 <xs:element name="BirthDate" type="xs:date"/>
 <xs:element name="BirthPlace" type="tns:string-rus-200"/>
 </xs:sequence>
</xs:complexType>

## Встраивание XML-фрагментов как строк

Запрещено встраивание XML-фрагментов в элементы простых типов и в атрибуты, например, с использованием механизмов XML escaping, CDATA и base64binary.

Неправильно:

<xs:schema
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:tns="http://qwe"
 targetNamespace="http://qwe"
 elementFormDefault="qualified"
 >

<xs:element name="SomeRequest">
 <xs:complexType>
 <xs:sequence>
 <xs:element name="ApplicantFamilyName" type="tns:string-rus-50"/>
 <xs:element name="ApplicantFirstName" type="tns:string-rus-50"/>
 <xs:element name="ApplicantPatronymic" type="tns:string-rus-50"/>
 <xs:element name="OriginalApplicationContent" **type="xs:string"**>
 <xs:annotation>
 **<xs:documentation>
 XML-документ; блок структурированных сведений
 заявки на предоставление госуслуги,
 на основании которой сделан этот запрос.
 </xs:documentation>** </xs:annotation>
 </xs:element>
 </xs:sequence>
 </xs:complexType>
</xs:element>
...
</xs:schema>

Соответствующий XML-фрагмент:

<ns1:SomeRequest xmlns:ns1="http://qwe">
 <ns1: ApplicantFamilyName>Иванов</ns1:VictimFamilyName>
 <ns1: ApplicantFirstName>Федор</ns1:VictimFirstName>
 <ns1: ApplicantPatronymic>Борисович</ns1:VictimPatronymic>
 <ns1:OriginalApplicationContent>**<![CDATA[
 <ns1:SomeApplication xmlns:ns1="http://asd">
 <ns1:FamilyName>Иванов</ns1:FamilyName>
 <ns1:FirstName>Федор</ns1:FirstName>
 <ns1:Patronymic>Борисович</ns1:Patronymic>
 <ns1:ApplicationReason>В соответствии с решением суда</ns1:ApplicationReason>
 </ns1:SomeApplication>
 ]]>**</ns1:OriginalApplicationContent>
</ns1:SomeRequest>

Правильно (схема запроса):

<xs:schema
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:tns="http://qwe"
 **xmlns:app="http://asd"**
 targetNamespace="http://qwe"
 elementFormDefault="qualified"
 >

**<xs:import namespace="http://asd" schemaLocation="asd-1.0.0.xsd"/>**

<xs:element name="SomeRequest">
 <xs:complexType>
 <xs:sequence>
 <xs:element name="ApplicantFamilyName" type="tns:string-rus-50"/>
 <xs:element name="ApplicantFirstName" type="tns:string-rus-50"/>
 <xs:element name="ApplicantPatronymic" type="tns:string-rus-50"/>
 <xs:element name="OriginalApplicationContent">
 <xs:annotation>
 <xs:documentation>
 XML-документ; блок структурированных сведений
 заявки на предоставление госуслуги,
 на основании которой сделан этот запрос.
 </xs:documentation>
 </xs:annotation>
 **<xs:complexType>
 <xs:sequence>
 <xs:element ref="app:SomeApplication"/>
 </xs:sequence>
 </xs:complexType>** </xs:element>
 </xs:sequence>
 </xs:complexType>
</xs:element>
...
</xs:schema>

Правильно (схема исходной заявки):

<xs:schema
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:tns="http://asd"
 targetNamespace="http://asd"
 elementFormDefault="qualified"
 >

<xs:element name="SomeApplication">
 <xs:complexType>
 <xs:sequence>
 <xs:element name="FamilyName" type="tns:string-rus-50"/>
 <xs:element name="FirstName" type="tns:string-rus-50"/>
 <xs:element name="Patronymic" type="tns:string-rus-50"/>
 <xs:element name="ApplicationReason" type="tns:string-rus-2000"/>
 </xs:sequence>
 </xs:complexType>
</xs:element>
...
</xs:schema>

Правильно (соответствующий XML-фрагмент):

<ns1:SomeRequest xmlns:ns1="http://qwe">
 <ns1: ApplicantFamilyName>Иванов</ns1:VictimFamilyName>
 <ns1: ApplicantFirstName>Федор</ns1:VictimFirstName>
 <ns1: ApplicantPatronymic>Борисович</ns1:VictimPatronymic>
 <ns1:OriginalApplicationContent>
 **<ns2:SomeApplication xmlns:ns2="http://asd">
 <ns2:FamilyName>Иванов</ns1:FamilyName>
 <ns2:FirstName>Федор</ns1:FirstName>
 <ns2:Patronymic>Борисович</ns1:Patronymic>
 <ns2:ApplicationReason> В соответствии с решением суда </ns1:ApplicationReason>
 </ns2:SomeApplication>** </ns1:OriginalApplicationContent>
</ns1:SomeRequest>

## Структурные зависимости в схемах

В данном разделе рассматриваются зависимости между элементами данных, имеющие место согласно логике предметной области, но не выраженные в XML-схемах.

### Зависимость обязательности элемента от значения другого элемента

Проблема зависимости обязательности элемента от значения другого элемента не всегда ее имеет корректное решение, поэтому подобные зависимости не запрещены. Однако необходимо устранять данную проблему, если имеется возможность избавиться от подобной зависимости путем изменения структуры схемы. Пример изменения структуры схемы:

Неправильно:

**<xs:element name="DoesHaveInternationalPassport" type="xs:boolean"/>**
<xs:element name="InternationalPassport"
 type="tns:InternationalPassportType" minOccurs="0"/>

Правильно:

<xs:element name="InternationalPassport"
 type="tns:InternationalPassportType" minOccurs="0"/>

Элемент «DoesHaveInternationalPassport» избыточен и должен быть исключен.

Для случаев, когда путём реструктуризации схемы проблема кросс-зависимостей не решается, СМЭВ предоставляет возможность включить в любую XML-схему Schematron-правила, в которой подобные зависимости можно описать явно.

### Согласование choice и значения элемента

Согласования choice и значений элементов запрещены, поскольку согласованный элемент всегда является избыточным.

Неправильно:

<simpleType name="PersonTypeType">
 <xs:restriction base="xs:string">
 **<xs:enumeration value="Физическое лицо"/>
 <xs:enumeration value="Юридическое лицо"/>** </xs:restriction>
</simpleType>
...
<xs:complexType name="SomeRequestType">
 <xs:sequence>
 <xs:element name="ApplicantType" type="tns:PersonTypeType"/>
 **<xs:choice>
 <xs:element name="ApplicantPP" type="tns:PhysicalPersonType"/>
 <xs:element name="ApplicantLP" type="tns:LegalPersonType"/>
 </xs:choice>** <xs:element name="ApplicationDetails" ...

Правильно:

<xs:complexType name="SomeRequestType">
 <xs:sequence>
 **<xs:choice>
 <xs:element name="ApplicantPP" type="tns:PhysicalPersonType"/>**

 **<xs:element name="ApplicantLP" type="tns:LegalPersonType"/>**

 **</xs:choice>** <xs:element name="ApplicationDetails" ...

### Согласованные элементы choice

Два элемента choice считаются согласованными, когда в одном из них есть хотя бы один вариант, при котором не все варианты второго элемента choice имеют смысл в рамках предметной области. Например:

<xs:choice>
 <xs:element name="ApplicantPP" type="tns:PhysicalPersonType">

 <xs:annotation>
 <xs:documentation>Заявитель – физическое лицо</xs:documentation>
 </xs:annotation>

 </xs:element>
 <xs:element name="ApplicantLP" type="tns:LegalPersonType">

 <xs:annotation>
 <xs:documentation>Заявитель – юридическое лицо</xs:documentation>
 </xs:annotation>
 </xs:element>
</xs:choice>
<xs:choice>
 <xs:element name="PostalAddress" type="tns:AddressType">

 <xs:annotation>
 <xs:documentation>Почтовый адрес</xs:documentation>
 </xs:annotation>

 </xs:element>
 <xs:element name="ActualAddress" type="tns: AddressType">

 <xs:annotation>
 <xs:documentation>Фактический адрес</xs:documentation>
 </xs:annotation>
 </xs:element>

 <xs:element **name="LegalAddress"** type="tns: AddressType">

 <xs:annotation>
 <xs:documentation>Юридический адрес</xs:documentation>
 </xs:annotation>
 </xs:element>

</xs:choice>

В приведенном выше примере, в случае если в первом choice выбран вариант «Заявитель – физическое лицо», то во втором choice нельзя выбирать вариант «Юридический адрес», т.к. у физических лиц нет юридических адресов. Таким образом, третий вариант второго choice не имеет смысла в рамках предметной области. Поэтому данные choice являются согласованными.

Согласованные элементы choice запрещены. Они должны быть устранены за счет переработки дизайна схемы.

### Согласованная обязательность элементов

Запрещено использование схем, в которых есть элементы, являющиеся дочерними элементами одного и того же элемента, и при этом не пустое значение одного из них влечет обязательность заполнения остальных, и наоборот. Такие элементы должны быть объединены либо в необязательную последовательность (sequence), либо в элемент структурированного типа.

Неправильно:

<xs:complexType name="SomeType">
 <xs:sequence>
 <xs:element name="FamilyName" type="tns:string-rus-50"/>
 <xs:element name="FirstName" type="tns:string-rus-50"/>
 <xs:element name="Patronymic" type="tns:string-rus-50"
 minOccurs="0"/>
 <xs:element name="BirthDate" type="xs:date"/>
 **<xs:element name="DeathDate" type="xs:date"**

 **minOccurs="0"/>
 <xs:element name="CauseOfDeath" type="tns:string-rus-500"**

 **minOccurs="0"/>** </xs:sequence>
</xs:complexType>

Правильно:

<xs:complexType name="SomeType">
 <xs:sequence>
 <xs:element name="FamilyName" type="tns:string-rus-50"/>
 <xs:element name="FirstName" type="tns:string-rus-50"/>
 <xs:element name="Patronymic" type="tns:string-rus-50"
 minOccurs="0"/>
 <xs:element name="BirthDate" type="xs:date"/>
 **<xs:sequence minOccurs="0">
 <xs:element name="DeathDate" type="xs:date"/>
 <xs:element name="CauseOfDeath" type="tns:string-rus-500"/>
 </xs:sequence>** </xs:sequence>
</xs:complexType>

## Тип xs:date

Для дат используется только тип «xs:date», которому соответствуют даты в формате ISO.

## Конструкция CDATA

Использование конструкции CDATA запрещено.

## Элементы верхнего уровня

В схемах, регистрируемых в СМЭВ, на верхнем уровне схемы (top level, http://www.w3.org/TR/2008/REC-xml-20081126/, http://www.w3.org/TR/2012/REC-xmlschema11-1-20120405/) запрещено описывать элементы и атрибуты, которые используются только внутри этой же схемы.

Неправильно:

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:smev-request="http://smev.gosuslugi.ru/request/rev111111"
 targetNamespace="http://smev.gosuslugi.ru/request/rev111111"
 elementFormDefault="qualified" attributeFormDefault="unqualified"
 version="2.4.4">

 <xs:element name="AppliedDocuments" type="smev-request:AppliedDocumentsType"/>

 <xs:complexType name="AppliedDocumentsType">
 <xs:sequence>
 <xs:element ref="smev-request:AppliedDocument"
 minOccurs="0" maxOccurs="unbounded"/>
 </xs:sequence>
 </xs:complexType>

 <xs:complexType name="AppliedDocumentType">
 <xs:sequence>
 <xs:element ref="smev-request:CodeDocument" minOccurs="0"/>
 <xs:element ref="smev-request:Name"/>
 <xs:element ref="smev-request:Number" minOccurs="0"/>
 <xs:element ref="smev-request:URL"/>
 <xs:element ref="smev-request:Type"/>
 <xs:element ref="smev-request:DigestValue" minOccurs="0"/>
 </xs:sequence>
 <xs:attribute ref="smev-request:ID" use="optional"/>
 </xs:complexType>

 <xs:element name="AppliedDocument"
 type="smev-request:AppliedDocumentType"/>
 **<xs:element name="CodeDocument" type="xs:string"/>
 <xs:element name="Name" type="xs:string"/>
 <xs:element name="Number" type="xs:string"/>
 <xs:element name="URL" type="xs:string"/>
 <xs:element name="Type" type="xs:string"/>
 <xs:element name="DigestValue" type="xs:base64Binary"/>
 <xs:attribute name="ID"/>
 <xs:simpleType><xs:restriction base="xs:ID"/></xs:simpleType>
 </xs:attribute>**</xs:schema>

Реализация в соответствии с требованиями текущего подраздела:

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:smev-request="http://smev.gosuslugi.ru/request/rev111111"
 targetNamespace="http://smev.gosuslugi.ru/request/rev111111"
 elementFormDefault="qualified"
 version="2.4.4">

 <xs:element name="AppliedDocuments" type="smev-request:AppliedDocumentsType"/>

 <xs:element name="AppliedDocument"
 type="smev-request:AppliedDocumentType"/>

 <xs:complexType name="AppliedDocumentsType">
 <xs:sequence>
 <xs:element ref="smev-request:AppliedDocument"
 minOccurs="0" maxOccurs="unbounded"/>
 </xs:sequence>
 </xs:complexType>

 <xs:complexType name="AppliedDocumentType">
 <xs:sequence>
 <xs:element name="CodeDocument" type="xs:string" minOccurs="0"/>
 <xs:element name="Name" type="xs:string"/>
 <xs:element name="Number" type="xs:string" minOccurs="0"/>
 <xs:element name="URL" type="xs:string"/>
 <xs:element name="Type" type="xs:string"/>
 <xs:element name="DigestValue" type="xs:string" minOccurs="0"/>
 </xs:sequence>
 <xs:attribute name="ID" type="xs:ID" use="optional"/>
 </xs:complexType>

</xs:schema>

## Одинаковые декларации в нескольких схемах

Рекомендуется не использовать идентичные объявления типов в нескольких схемах одного и того же ОИВ. Необходимо выделять общие типы либо в отдельный XML-документ, включаемый через «xs:include», либо в отдельную схему со своим «target namespace», включаемую через «xs:import».

## Схемы, не отражающие логическую структуру данных

Не должны регистрироваться в СМЭВ схемы, структура которых не соответствует логической структуре данных.

Неправильно:

<xs:complexType name="SomeType">
 <xs:sequence>
 <xs:element name="ApplicantFamilyName" type="tns:string-rus-50"/>
 <xs:element name="ApplicantFirstName" type="tns:string-rus-50"/>
 <xs:element name="ApplicantPatronymic" type="tns:string-rus-50"
 minOccurs="0"/>
 <xs:element name="ApplicantPassportSeries" type="xs:string"

 minOccurs="0"/>
 <xs:element name="ApplicantPassportNumber" type="xs:string"

 minOccurs="0"/>
 <xs:element name="ApplicantBirthDate" type="xs:date" minOccurs="0"/>
 <xs:element name="AgentFamilyName" type="tns:string-rus-50"/>
 <xs:element name="AgentFirstName" type="tns:string-rus-50"/>
 <xs:element name="AgentPatronymic" type="tns:string-rus-50"

 minOccurs="0"/>
 ...
 </xs:sequence>
</xs:complexType>

Правильно:

<xs:complexType name="SomeType">
 <xs:sequence>
 <xs:element name="ApplicantInfo">
 <xs:complexType>

 <xs:sequence>

 <xs:element name="FamilyName" type="tns:string-rus-50"/>
 <xs:element name="FirstName" type="tns:string-rus-50"/>
 <xs:element name="Patronymic" type="tns:string-rus-50"

 minOccurs="0"/>
 <xs:element name="PersonIdentityDocument" minOccurs="0">
 <xs:complexType>

 <xs:sequence>

 <xs:element name="PassportSeries" type="xs:string"/>
 <xs:element name="PassportNumber" type="xs:string"/>

 </xs:sequence>

 </xs:complexType>

 </xs:element>
 <xs:element name="BirthDate" type="xs:date" minOccurs="0"/>

 </xs:sequence>

 </xs:complexType>

 </xs:element>
 <xs:element name="AgentInfo">
 <xs:complexType>

 <xs:sequence>

 <xs:element name="FamilyName" type="tns:string-rus-50"/>
 <xs:element name="FirstName" type="tns:string-rus-50"/>
 <xs:element name="Patronymic" type="tns:string-rus-50"

 minOccurs="0"/>
 </xs:sequence>

 </xs:complexType>

 </xs:element>
 ...
 </xs:sequence>
</xs:complexType>

## Передача в СМЭВ элементов оформления

В СМЭВ для запросов и ответов в блоке структурированных сведений запрещена передача данных, не являющихся входными или выходными параметрами запросов/ответов, в первую очередь – элементов оформления, таких как шапки и подвалы выписок. Если по бизнес-процессу требуется распечатка документа в установленной форме на стороне потребителя данных, и потребителей много, а установленная форма сложная, передающая сторона, кроме XML-контента формирует PDF-документ, и пересылает его как вложение.

## Передача блока структурированных сведений как вложения

Настоящий документ требует, чтобы схема сообщения СМЭВ (запроса, заявки или ответа) описывала действительные входные или выходные параметры и данные. Запрещена практика, при которой в схеме для СМЭВ поставщик данных описывает пустой блок структурированных сведений, а параметры предметной области требует передавать как вложение.

## Вложения

Документы, приложенные к заявкам, запросам и ответам, должны передаваться только в конверте СМЭВ (см. схему urn://x-artefacts-smev-gov-ru/services/message-exchange/types/1.2), либо через механизмы Файлового хранилища СМЭВ3.х.

Запрещается передача приложенных документов внутри XML-структур, представляющих блок структурированных сведений заявок, запросов и ответов (document embedding). XML-документы должны ссылаться на приложенные документы, используя идентификаторы вложений (см. {urn://x-artefacts-smev-gov-ru/services/message-exchange/types/basic/1.2:AttachmentContent}).

Неправильно:

<xs:complexType name="SomeDocumentType">
 <xs:sequence>
 <xs:element name="Series" type="tns:string-4-digits"/>
 <xs:element name="Number" type="tns:string-6-digits"/>
 <xs:element name="IssueDate" type="xs:date"/>
 <xs:element name="Issuer" type="tns:string-rus-200"/>
 <xs:element name="Scan" maxOccurs="10">
 <xs:complexType>
 <xs:sequence>
 <xs:element name="mimeType" type="xs:string"/>
 **<xs:element name="content" type="xs:base64Binary"/>**
 </xs:sequence>
 </xs:complexType>
 </xs:element>
 </xs:sequence>
</xs:complexType>

В схеме urn://x-artefacts-smev-gov-ru/supplementary/commons/1.2 объявлен элемент AttachmentRef, рекомендуется для ссылок на вложения использовать его.

Правильно:

<xs:complexType name="SomeDocumentType">
 <xs:sequence>
 <xs:element name="Series" type="tns:string-4-digits"/>
 <xs:element name="Number" type="tns:string-6-digits"/>
 <xs:element name="IssueDate" type="xs:date"/>
 <xs:element name="Issuer" type="tns:string-rus-200"/>
 <xs:element name="Scan" maxOccurs="10">
 <xs:complexType>
 <xs:sequence>
 <xs:element name="mimeType" type="xs:string"/>
 **<xs:element xmlns:sc=”urn://x-artefacts-smev-gov-ru/supplementary/commons/1.2” ref="sc:AttachmentRef"/>** </xs:sequence>
 </xs:complexType>
 </xs:element>
 </xs:sequence>
</xs:complexType>

# Стиль написания XML схем. Рекомендации

## Использование типа xs:string

Настоятельно не рекомендуется объявлять элементы и атрибуты с помощью типа «xs:string». Рекомендуется использовать производные типы с ограничением максимальной длины строки.

## Определение простых типов

Не рекомендуется использовать анонимные простые типы, т.к. их использование усложняет сопровождение схем. Следует объявлять именованные простые типы и ссылаться на них.

Не рекомендуется:

<xs:element name="NotQuiteGood">
 **<xs:simpleType>
 <xs:restriction base="xs:string">
 <xs:maxLength value="20"/>
 </xs:restriction>
 </xs:simpleType>**</xs:element>

Рекомендуется:

<xs:element name="ThisLooksBetter" **type="tns:string-20"**/>

<xs:simpleType name="string-20">
 <xs:restriction base="xs:string">
 <xs:maxLength value="20"/>
 </xs:restriction>
</xs:simpleType>

## Значения по умолчанию в XSD

Не следует использовать атрибут «form» в объявлениях элементов. Рекомендуется использовать атрибут «elementFormDefault="qualified"» в объявлении схемы.

Не следует использовать атрибут «form» в объявлениях атрибутов. Атрибут «attributeFormDefault» также не рекомендуется использовать, т.к. по умолчанию он имеет значение «unqualified», что и требуется.

Не следует указывать явно «minOccurs="1"» и «maxOccurs="1"», т.к. данные значения являются значениями по умолчанию для указанных атрибутов.

## Избыточная вложенность

Не следует увеличивать вложенность XML-структур без необходимости. Следует стремиться к тому, чтобы структура XML-документа четко отражала логическую структуру данных.

Не рекомендуется:

<xs:complexType name="FIOType">
 <xs:sequence>
 <xs:element name="FamilyName" type="tns:string-rus-50"/>
 <xs:element name="FirstName" type="tns:string-rus-50"/>
 <xs:element name="Patronymic" type="tns:string-rus-50"/>
 </xs:sequence>
</xs:complexType>

<xs:complexType name="PhysicalPersonBasicData">
 <xs:sequence>
 **<xs:element name="FIO" type="tns:FIOType"/>** <xs:element name="BirthDate" type="xs:date"/>
 <xs:element name="BirthPlace" type="xs:string"/>
 </xs:sequence>
</xs:complexType>

Соответствующий XML-фрагмент:

<ns1:Applicant>
 **<ns1:FIO>** <!-- Избыточный уровень вложенности -->
 **<ns1:FamilyName>Иванов</ns1:FamilyName>
 <ns1:FirstName>Федор</ns1:FirstName>
 <ns1:Patronymic>Борисович</ns1:Patronymic>
 </ns1:FIO>** <ns1:BirthDate>1980-05-05</ns1:BirthDate>
 <ns1:BirthPlace>г. Балашиха Московской области</ns1:BirthPlace>
</ns1:Applicant>

Рекомендуется:

 <xs:group name="FIOGroup">
 <xs:sequence>
 <xs:element name="FamilyName" type="tns:string-rus-50"/>
 <xs:element name="FirstName" type="tns:string-rus-50"/>
 <xs:element name="Patronymic" type="tns:string-rus-50"/>
 </xs:sequence>
 </xs:group>

 <xs:complexType name="PhysicalPersonBasicData">
 <xs:sequence>
 **<xs:group ref="tns:FIOGroup"/>** <xs:element name="BirthDate" type="xs:date"/>
 <xs:element name="BirthPlace" type="xs:string"/>
 </xs:sequence>
 </xs:complexType>

Соответствующий XML-фрагмент:

<ns1:Applicant>
 <ns1:FamilyName>Иванов</ns1:FamilyName>
 <ns1:FirstName>Фёдор</ns1:FirstName>
 <ns1:Patronymic>Борисович</ns1:Patronymic>
 <ns1:BirthDate>1980-05-05</ns1:BirthDate>
 <ns1:BirthPlace>г. Балашиха Московской области</ns1:BirthPlace>
</ns1:Applicant>

## Избыточная информация в наименованиях элементов

Наименования элементов рекомендуется назначать с учетом структуры XML-документа. Рекомендуется упрощать наименования элементов, если структура XML-документа позволяет это сделать.

Например:

<xs:complexType name="SomeApplicationType">
 <xs:sequence>
 <xs:element name="**Beneficiary**">
 <xs:complexType>
 <xs:sequence>
 <xs:element name="**Beneficiary**FamilyName" type="tns:string-rus-50"/>
 <xs:element name="**Beneficiary**FirstName" type="tns:string-rus-50"/>
 <xs:element name="**Beneficiary**Patronymic" type="tns:string-rus-50"/>
 </xs:sequence>
 </xs:complexType>
 </xs:element>
 <xs:element name="**Trustee**">
 <xs:complexType>
 <xs:sequence>
 <xs:element name="**Trustee**FamilyName" type="tns:string-rus-50"/>
 <xs:element name="**Trustee**FirstName" type="tns:string-rus-50"/>
 <xs:element name="**Trustee**Patronymic" type="tns:string-rus-50"/>
 </xs:sequence>
 </xs:complexType>
 </xs:element>
 </xs:sequence>
</xs:complexType>

Рекомендуется:

 <xs:complexType name="SomeApplicationType">
 <xs:sequence>
 <xs:element name="**Beneficiary**">
 <xs:complexType>
 <xs:sequence>
 <xs:element name="FamilyName" type="tns:string-rus-50"/>
 <xs:element name="FirstName" type="tns:string-rus-50"/>
 <xs:element name="Patronymic" type="tns:string-rus-50"/>
 </xs:sequence>
 </xs:complexType>
 </xs:element>
 <xs:element name="**Trustee**">
 <xs:complexType>
 <xs:sequence>
 <xs:element name="FamilyName" type="tns:string-rus-50"/>
 <xs:element name="FirstName" type="tns:string-rus-50"/>
 <xs:element name="Patronymic" type="tns:string-rus-50"/>
 </xs:sequence>
 </xs:complexType>
 </xs:element>
 </xs:sequence>
</xs:complexType>

# Шаблоны типовых данных и наборов данных

В данном разделе приводится описание групп и типов данных, содержащихся в файле smev-supplementary-commons-1.2.xsd. В данном файле собраны группы и типы данных, структура и форматно-логический контроль которых не зависят от специфики описываемой предметной области, и которые повсеместно используются при разработке схем. Файл smev-supplementary-commons-1.2.xsd расположен в пространстве имен по адресу urn://x-artefacts-smev-gov-ru/supplementary/commons/1.2.

## Данные физического лица

### Фамилия, имя, отчество физического лица. Группа PhysicalPersonQualifiedName-ModelGroup

Группа PhysicalPersonQualifiedName-ModelGroup содержит Фамилию, Имя и Отчество физического лица. Все элементы имеют тип rus-100.

|  |
| --- |
| *<xs:group name="PhysicalPersonQualifiedName-ModelGroup">**<xs:annotation>* *<xs:documentation>ФИО физического лица.</xs:documentation>**</xs:annotation>**<xs:sequence>* *<xs:element name="FamilyName" type="tns:rus-100">* *<xs:annotation><xs:documentation>Фамилия</xs:documentation></xs:annotation>* *</xs:element>* *<xs:element name="FirstName" type="tns:rus-100">* *<xs:annotation><xs:documentation>Имя</xs:documentation></xs:annotation>* *</xs:element>* *<xs:element name="Patronymic" type="tns:rus-100" minOccurs="0">* *<xs:annotation><xs:documentation>Отчество</xs:documentation>* *</xs:annotation>* *</xs:element>**</xs:sequence>**</xs:group>* |

### Установочные данные: фамилия, имя, отчество, дата и место рождения. Группа PhysicalPersonIdentificationInformation-ModelGroup

Группа PhysicalPersonIdentificationInformation-ModelGroup в дополнение к предыдущей группе (PhysicalPersonQualifiedName-ModelGroup, см. раздел 6.1.1) содержит дату и место рождения.

|  |
| --- |
| *<xs:group name="PhysicalPersonIdentificationInformation-ModelGroup">* *<xs:annotation><xs:documentation>Установочные данные физического лица: ФИО + дата и место рождения.</xs:documentation></xs:annotation>* *<xs:sequence>* *<xs:group ref="tns:PhysicalPersonQualifiedName-ModelGroup"/>* *<xs:element name="BirthDate" type="xs:date">* *<xs:annotation><xs:documentation>Дата рождения</xs:documentation></xs:annotation>* *</xs:element>* *<xs:element name="BirthPlace" type="tns:rus-and-num-500">* *<xs:annotation><xs:documentation>Место рождения</xs:documentation></xs:annotation>* *</xs:element>* *</xs:sequence>**</xs:group>* |

### Пол. Тип GenderType

Выпадающий список со значениями «Мужской» и «Женский». Используется для указания пола физического лица.

|  |
| --- |
| *<xs:simpleType name="GenderType">* *<xs:annotation>* *<xs:documentation>Пол.</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:enumeration value="Male"/>* *<xs:enumeration value="Female"/>* *</xs:restriction>**</xs:simpleType>* |

### Документы, удостоверяющие личность

Описание документов, удостоверяющих личность, реализовано следующим образом. Все документы разделены на три группы:

* основные документы, удостоверяющие личность (подходят для любых юридических действий):
* паспорт РФ;
* паспорт иностранного гражданина;
* вид на жительство в РФ.
* документы, удостоверяющие личность, ограниченного действия (используются в качестве ДУЛ только в определенных ситуациях, например, при выезде за границы РФ):
* загранпаспорт;
* военный билет;
* паспорт моряка;
* паспорт СССР;
* свидетельство о рождении;
* водительское удостоверение.
* временные документы, удостоверяющие личность (выдаются в качестве документа, заменяющего ДУЛ, на время оформления/восстановления ДУЛ):
* справка об освобождении;
* справка об утере паспорта;
* справка о регистрации по форме 9;
* временное удостоверение личности гражданина РФ по форме 2П.

Описанное выше разделение реализовано следующим образом:

|  |
| --- |
| *<xs:element name="IdentityDocument" type="tns:AnyIdentityDocumentType">* *<xs:annotation>* *<xs:documentation>Документ, удостоверяющий личность</xs:documentation>* *</xs:annotation>**</xs:element>**…**<xs:complexType name="AnyIdentityDocumentType">* *<xs:annotation>* *<xs:documentation>Документ, удостоверяющий личность</xs:documentation>* *</xs:annotation>* *<xs:choice>* *<xs:group ref="tns:PrimaryIdentityDocument-ModelGroup"/>* *<xs:group ref="tns:WeakIdentityDocument-ModelGroup"/>* *<xs:group ref="tns:TemporaryIdentityDocument-ModelGroup"/>* *</xs:choice>**</xs:complexType>**…**<xs:element name="PrimaryIdentityDocument" type="tns:PrimaryIdentityDocumentType">* *<xs:annotation>* *<xs:documentation>Основной документ, удостоверяющий личность.</xs:documentation>* *</xs:annotation>**</xs:element>**…**<xs:complexType name="PrimaryIdentityDocumentType">* *<xs:annotation>* *<xs:documentation>Основной документ, удостоверяющий личность</xs:documentation>* *</xs:annotation>* *<xs:group ref="tns:PrimaryIdentityDocument-ModelGroup"/>**</xs:complexType>**…**<xs:group name="PrimaryIdentityDocument-ModelGroup">* *<xs:annotation>* *<xs:documentation>Основной документ, удостоверяющие личность. Подходит для любых юр. действий.</xs:documentation>* *</xs:annotation>* *<xs:choice>* *<xs:element ref="tns:PassportRF"/>* *<xs:element ref="tns:ForeignPassport"/>* *<xs:element ref="tns:ResidencePermitRF"/>* *</xs:choice>**</xs:group>**<xs:group name="WeakIdentityDocument-ModelGroup">* *<xs:annotation>* *<xs:documentation>Документы, удостоверяющие личность, ограниченного применения.</xs:documentation>* *</xs:annotation>* *<xs:choice>* *<xs:element ref="tns:InternationalPassportRF"/>* *<xs:element ref="tns:MilitaryPassport"/>* *<xs:element ref="tns:SailorPassport"/>* *<xs:element ref="tns:SovietPassport"/>* *<xs:element ref="tns:BirthCertificate"/>* *<xs:element ref="tns:DrivingLicenseRF"/>* *</xs:choice>**</xs:group>**<xs:group name="TemporaryIdentityDocument-ModelGroup">* *<xs:annotation>* *<xs:documentation>Документы, удостоверяющие личность, временные.</xs:documentation>* *</xs:annotation>* *<xs:choice>* *<xs:element ref="tns:ReleaseCertificate"/>* *<xs:element ref="tns:PassportLossCertificate"/>* *<xs:element ref="tns:Form9Certificate"/>* *<xs:element ref="tns:TemporaryIdentityCardRF"/>* *</xs:choice>**</xs:group>**…**<xs:element name="PassportRF" type="tns:PassportRFType">* *<xs:annotation>* *<xs:documentation>Паспорт РФ</xs:documentation>* *</xs:annotation>**</xs:element>**…**<xs:element name="ForeignPassport" type="tns:NotRestrictedDocumentType">* *<xs:annotation>* *<xs:documentation>Паспорт иностранного гражданина</xs:documentation>* *</xs:annotation>**</xs:element>**…**<xs:element name="ResidencePermitRF" type="tns:InternationalPassportRFType">* *<xs:annotation>* *<xs:documentation>Вид на жительство в РФ. Формат серии и номера совпадает с загранпаспортом.</xs:documentation>* *</xs:annotation>**</xs:element>**…**<xs:element name="InternationalPassportRF" type="tns:InternationalPassportRFType">* *<xs:annotation>* *<xs:documentation>Загранпаспорт</xs:documentation>* *</xs:annotation>**</xs:element>**…**<xs:element name="MilitaryPassport" type="tns:MilitaryPassportDocumentType">* *<xs:annotation>* *<xs:documentation>Военный билет</xs:documentation>* *</xs:annotation>**</xs:element>**…**<xs:element name="SailorPassport" type="tns:MilitaryPassportDocumentType">* *<xs:annotation>* *<xs:documentation>Паспорт моряка</xs:documentation>* *</xs:annotation>**</xs:element>**…**<xs:element name="SovietPassport" type="tns:SovietPassportType">* *<xs:annotation>* *<xs:documentation>Паспорт СССР</xs:documentation>* *</xs:annotation>**</xs:element>**…**<xs:element name="BirthCertificate" type="tns:SovietPassportType">* *<xs:annotation>* *<xs:documentation>Свидетельство о рождении</xs:documentation>* *</xs:annotation>**</xs:element>**…**<xs:element name="DrivingLicenseRF" type="tns:DrivingLicenseRFType">* *<xs:annotation>* *<xs:documentation>Водительское удостоверение РФ</xs:documentation>* *</xs:annotation>**</xs:element>**…**<xs:element name="ReleaseCertificate" type="tns:NotRestrictedDocumentType">* *<xs:annotation>* *<xs:documentation>Справка об освобождении</xs:documentation>* *</xs:annotation>**</xs:element>**…**<xs:element name="PassportLossCertificate" type="tns:NotRestrictedDocumentType">* *<xs:annotation>* *<xs:documentation>Справка об утере паспорта</xs:documentation>* *</xs:annotation>**</xs:element>**…**<xs:element name="Form9Certificate" type="tns:NotRestrictedDocumentType">* *<xs:annotation>* *<xs:documentation>Справка о регистрации по форме 9</xs:documentation>* *</xs:annotation>**</xs:element>**…**<xs:element name="TemporaryIdentityCardRF" type="tns:NotRestrictedDocumentType">* *<xs:annotation>* *<xs:documentation>Временное удостоверение личности гражданина РФ по форме 2П</xs:documentation>* *</xs:annotation>**</xs:element>* |

В качестве типов данных документов используются 6 типов:

* паспорт РФ: PassportRFType (см. раздел 6.1.5);
* паспорт иностранного гражданина: NotRestrictedDocumentType (см. раздел 6.1.6);
* вид на жительство в РФ: InternationalPassportRFType (см. раздел 6.1.10);
* загранпаспорт: InternationalPassportRFType (см. раздел 6.1.10);
* военный билет: MilitaryPassportDocumentType (см. раздел 6.1.9);
* паспорт моряка: MilitaryPassportDocumentType (см. раздел 6.1.9);
* паспорт СССР: SovietPassportType (см. раздел 6.1.7);
* свидетельство о рождении: SovietPassportType (см. раздел 6.1.7);
* водительское удостоверение: DrivingLicenseRFType (см. раздел 6.1.8);.
* справка об освобождении: NotRestrictedDocumentType (см. раздел 6.1.6);
* справка об утере паспорта: NotRestrictedDocumentType (см. раздел 6.1.6);
* справка о регистрации по форме 9: NotRestrictedDocumentType (см. раздел 6.1.6);
* временное удостоверение личности гражданина РФ по форме 2П: NotRestrictedDocumentType (см. раздел 6.1.6).

Ниже приведено описание типов данных документов.

### Паспорт РФ. Тип PassportRFType

Тип PassportRFType используется для описания реквизитов паспорта гражданина РФ, действующего на территории РФ с 1 октября 1997 года.

|  |
| --- |
| *<xs:complexType name="PassportRFType">* *<xs:annotation>* *<xs:documentation>Паспорт РФ</xs:documentation>* *</xs:annotation>* *<xs:sequence>* *<xs:element name="Series" type="tns:digits-4">* *<xs:annotation>* *<xs:documentation>Серия</xs:documentation>* *</xs:annotation>* *</xs:element>* *<xs:element name="Number" type="tns:digits-6">* *<xs:annotation>* *<xs:documentation>Номер</xs:documentation>* *</xs:annotation>* *</xs:element>* *<xs:element name="IssueDate" type="xs:date">* *<xs:annotation>* *<xs:documentation>Дата выдачи</xs:documentation>* *</xs:annotation>* *</xs:element>* *<xs:element name="Issuer" type="tns:rus-and-num-200">* *<xs:annotation>* *<xs:documentation>Кем выдан</xs:documentation>* *</xs:annotation>* *</xs:element>* *</xs:sequence>**</xs:complexType>* |

### Документ с нерегламентированными реквизитами. Тип NotRestrictedDocumentType

Тип NotRestrictedDocumentType используется для описания документов с нерегламентированными реквизитами, таких как паспорт иностранного гражданина, справка об освобождении, справка об утере паспорта, справка о регистрации по форме 9, временное удостоверение личности гражданина РФ по форме 2П.

|  |
| --- |
| *<xs:complexType name="NotRestrictedDocumentType">* *<xs:annotation>* *<xs:documentation>Документ с нерегламентированными серией и номером</xs:documentation>* *</xs:annotation>* *<xs:sequence>* *<xs:element name="Series" type="tns:string-20" minOccurs="0">* *<xs:annotation>* *<xs:documentation>Серия</xs:documentation>* *</xs:annotation>* *</xs:element>* *<xs:element name="Number">* *<xs:annotation>* *<xs:documentation>Номер</xs:documentation>* *</xs:annotation>* *<xs:simpleType>* *<xs:restriction base="xs:string">* *<xs:pattern value="[0-9а-яА-ЯA-Za-z]{1,25}"/>* *</xs:restriction>* *</xs:simpleType>* *</xs:element>* *<xs:element name="IssueDate" type="xs:date">* *<xs:annotation>* *<xs:documentation>Дата выдачи</xs:documentation>* *</xs:annotation>* *</xs:element>* *<xs:element name="Issuer" type="tns:rus-and-num-200">* *<xs:annotation>* *<xs:documentation>Кем выдан</xs:documentation>* *</xs:annotation>* *</xs:element>* *</xs:sequence>**</xs:complexType>* |

### Паспорт СССР, свидетельство о рождении. Тип SovietPassportType

Тип SovietPassportType используется для описания реквизитов паспорта СССР и свидетельства о рождении.

|  |
| --- |
| *<xs:complexType name="SovietPassportType">* *<xs:annotation>* *<xs:documentation>Паспорт CCCP или свидетельство о рождении РФ.</xs:documentation>* *</xs:annotation>* *<xs:sequence>* *<xs:element name="Series" type="tns:SovietStyleDocumentSeriesType">* *<xs:annotation>* *<xs:documentation>Серия</xs:documentation>* *</xs:annotation>* *</xs:element>* *<xs:element name="Number" type="tns:digits-6">* *<xs:annotation>* *<xs:documentation>Номер</xs:documentation>* *</xs:annotation>* *</xs:element>* *<xs:element name="IssueDate" type="xs:date">* *<xs:annotation>* *<xs:documentation>Дата выдачи</xs:documentation>* *</xs:annotation>* *</xs:element>* *<xs:element name="Issuer" type="tns:rus-and-num-200">* *<xs:annotation>* *<xs:documentation>Кем выдан</xs:documentation>* *</xs:annotation>* *</xs:element>* *</xs:sequence>**</xs:complexType>* |

### Водительское удостоверение. Тип DrivingLicenseRFType

Тип DrivingLicenseRFType используется для описания реквизитов водительского удостоверения, выданного в РФ.

|  |
| --- |
| *<xs:complexType name="DrivingLicenseRFType">* *<xs:annotation>* *<xs:documentation>Водительское удостоверение, выданное в РФ</xs:documentation>* *</xs:annotation>* *<xs:sequence>* *<xs:element name="Series">* *<xs:annotation>* *<xs:documentation>* *Серия.* *Две цифры и две русские заглавные буквы для водительского удостоверения, полученного до 1 марта 2011 г.,*  *или четыре цифры для водительского удостоверения, полученного после 1 марта 2011 г. Пример: 44АА или 4403* *</xs:documentation>* *</xs:annotation>* *<xs:simpleType>* *<xs:restriction base="xs:string">* *<xs:length value="4"/>* *<xs:pattern value="[0-9]{2}([0-9]{2})|([А-Я]{2})"/>* *</xs:restriction>* *</xs:simpleType>* *</xs:element>* *<xs:element name="Number" type="tns:digits-6">* *<xs:annotation>* *<xs:documentation>Номер</xs:documentation>* *</xs:annotation>* *</xs:element>* *<xs:element name="IssueDate" type="xs:date">* *<xs:annotation>* *<xs:documentation>Дата выдачи</xs:documentation>* *</xs:annotation>* *</xs:element>* *<xs:element name="Issuer" type="tns:rus-and-num-200">* *<xs:annotation>* *<xs:documentation>Кем выдан</xs:documentation>* *</xs:annotation>* *</xs:element>* *</xs:sequence>**</xs:complexType>* |

### Военный билет, паспорт моряка. Тип MilitaryPassportDocumentType

Тип MilitaryPassportDocumentType используется для описания реквизитов военного билета, а также паспорта моряка.

|  |
| --- |
| *<xs:complexType name="MilitaryPassportDocumentType">* *<xs:annotation>* *<xs:documentation>Военный билет или паспорт моряка</xs:documentation>* *</xs:annotation>* *<xs:sequence>* *<xs:element name="Series">* *<xs:annotation>* *<xs:documentation>Серия</xs:documentation>* *</xs:annotation>* *<xs:simpleType>* *<xs:restriction base="xs:string">* *<xs:length value="2"/>* *<xs:pattern value="[А-Я]{2}"/>* *</xs:restriction>* *</xs:simpleType>* *</xs:element>* *<xs:element name="Number" type="tns:digits-7">* *<xs:annotation>* *<xs:documentation>Номер</xs:documentation>* *</xs:annotation>* *</xs:element>* *<xs:element name="IssueDate" type="xs:date">* *<xs:annotation>* *<xs:documentation>Дата выдачи</xs:documentation>* *</xs:annotation>* *</xs:element>* *<xs:element name="Issuer" type="tns:rus-and-num-200">* *<xs:annotation>* *<xs:documentation>Кем выдан</xs:documentation>* *</xs:annotation>* *</xs:element>* *</xs:sequence>**</xs:complexType>* |

### Загранпаспорт, вид на жительство в РФ. Тип InternationalPassportRFType

Тип InternationalPassportRFType используется для описания реквизитов заграничного паспорта, удостоверяющего личность гражданина РФ за пределами РФ, а вида на жительство в РФ.

|  |
| --- |
| *<xs:complexType name="InternationalPassportRFType">* *<xs:annotation>* *<xs:documentation>Загранпаспорт, либо вид на жительство в РФ.</xs:documentation>* *</xs:annotation>* *<xs:sequence>* *<xs:element name="Series" type="tns:digits-2">* *<xs:annotation>* *<xs:documentation>Серия</xs:documentation>* *</xs:annotation>* *</xs:element>* *<xs:element name="Number" type="tns:digits-7">* *<xs:annotation>* *<xs:documentation>Номер</xs:documentation>* *</xs:annotation>* *</xs:element>* *<xs:element name="IssueDate" type="xs:date">* *<xs:annotation>* *<xs:documentation>Дата выдачи</xs:documentation>* *</xs:annotation>* *</xs:element>* *<xs:element name="Issuer" type="tns:rus-and-num-200">* *<xs:annotation>* *<xs:documentation>Кем выдан</xs:documentation>* *</xs:annotation>* *</xs:element>* *</xs:sequence>**</xs:complexType>* |

## Типы для вложений

### Ссылка на вложение. Тип AttachmentRefType

Тип AttachmentRefType используется для ссылки из содержательной части запроса (заявки, ответа) на вложение, находящееся в том же СМЭВ-сообщении.

|  |
| --- |
| *<xs:element name="AttachmentRef" type="tns:AttachmentRefType"/>**<xs:complexType name="AttachmentRefType">* *<xs:annotation>* *<xs:documentation>* *Ссылка из содержательной части запроса (заявки, ответа) на вложение, находящееся в том же СМЭВ-сообщении.* *</xs:documentation>* *</xs:annotation>* *<xs:attribute name="attachmentId" type="xs:string">* *<xs:annotation>* *<xs:documentation>* *Идентификатор вложения, на которое ссылаемся. Должен быть равен значению* *//{urn://x-artefacts-smev-gov-ru/smev-core/client-interaction/basic/1.0.0}AttachedFile[n]/Id/text()* *того вложения, на которое нужно сослаться.* *</xs:documentation>* *</xs:annotation>* *</xs:attribute>**</xs:complexType>* |

### Вложение в формате RFC-2046. Тип RFC2046MimeTypesType

Тип RFC2046MimeTypesType используется для описания вложений в формате RFC-2046.

|  |
| --- |
| *<xs:simpleType name="RFC2046MimeTypesType">* *<xs:annotation>* *<xs:documentation>Типы вложений, в формате RFC-2046.</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:pattern value="(text|image|audio|video|application)/[a-zA-Z0-9\-+.]\*"/>* *</xs:restriction>**</xs:simpleType>* |

## Реквизиты юридических, физических лиц и ИП

### СНИЛС. Тип SNILSType

Тип для описания СНИЛС физического лица. Длина ровно 11 символов, разрешается использовать только цифры.

|  |
| --- |
| *<xs:simpleType name="SNILSType">* *<xs:annotation>* *<xs:documentation>СНИЛС</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:length value="11"/>* *<xs:pattern value="\d{11}"/>* *</xs:restriction>**</xs:simpleType>* |

### ИНН физического лица. Тип PhysicalPersonINNType

Тип для описания ИНН физического лица (ИП). Длина ровно 12 символов, разрешается использовать только цифры.

|  |
| --- |
| *<xs:simpleType name="PhysicalPersonINNType">* *<xs:annotation>* *<xs:documentation>ИНН физического лица</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:length value="12"/>* *<xs:pattern value="\d{12}"/>* *</xs:restriction>**</xs:simpleType>* |

### ОГРН ИП. Тип OGRNSoleProprietorType

Тип для описания ОГРН индивидуальных предпринимателей. Длина ровно 15 символов, разрешается использовать только цифры. Первая цифра – 3.

|  |
| --- |
| *<xs:simpleType name="OGRNSoleProprietorType">* *<xs:annotation>* *<xs:documentation>ОГРН индивидуального предпринимателя</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:length value="15"/>* *<xs:pattern value="[3]{1}\d{14}"/>* *</xs:restriction>**</xs:simpleType>* |

### ИНН юридического лица. Тип LegalPersonINNType

Тип для описания ИНН юридических лиц. Длина ровно 10 символов, разрешается использовать только цифры.

|  |
| --- |
| *<xs:simpleType name="LegalPersonINNType">* *<xs:annotation>* *<xs:documentation>ИНН юридического лица</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:length value="10"/>* *<xs:pattern value="\d{10}"/>* *</xs:restriction>**</xs:simpleType>* |

### ОГРН. Тип OGRNCompanyType

Тип для описания ОГРН юридических лиц. Длина ровно 13 символов, разрешается использовать только цифры. Первая цифра – 1, 2 или 5.

|  |
| --- |
| *<xs:simpleType name="OGRNCompanyType">* *<xs:annotation>* *<xs:documentation>ОГРН юридического лица</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:length value="13"/>* *<xs:pattern value="[125]{1}\d{12}"/>* *</xs:restriction>**</xs:simpleType>* |

### КПП. Тип KPPTyp

Тип для описания КПП юридических лиц. Длина ровно 9 символов, разрешается использовать только цифры.

|  |
| --- |
| *<xs:simpleType name="KPPType">* *<xs:annotation>* *<xs:documentation>КПП</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:length value="9"/>* *<xs:pattern value="\d{9}"/>* *</xs:restriction>**</xs:simpleType>* |

## Контакты

### Индекс. Тип PostalCodeRFType

Тип для почтового индекса. Длина ровно 6 символов, разрешается использовать только цифры.

|  |
| --- |
| *<xs:simpleType name="PostalCodeRFType">* *<xs:annotation>* *<xs:documentation>* *Почтовый индекс, РФ.* *Намеренно указаны одновременно и length и pattern, чтобы облегчить JavaScript коду валидацию "на лету".* *</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:length value="6"/>* *<xs:pattern value="\d{6}"/>* *</xs:restriction>**</xs:simpleType>* |

### Телефонный номер. Тип PhoneNumberRFType

Тип для телефонных номеров в РФ. Длина ограничена 10 символами, разрешается использовать только цифры.

|  |
| --- |
| *<xs:simpleType name="PhoneNumberRFType">* *<xs:annotation>* *<xs:documentation>Номер телефона в РФ, в десятизначном формате</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:maxLength value="10"/>* *<xs:pattern value="\d{10}"/>* *</xs:restriction>**</xs:simpleType>* |

### Электронная почта. Тип EmailAddressType

Тип для адреса электронной почты.

|  |
| --- |
| *<xs:simpleType name="EmailAddressType">* *<xs:annotation>* *<xs:documentation>Адрес электронной почты</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:maxLength value="100"/>* *<xs:pattern value="[0-9a-zA-Z\_.\-]{2,50}[@]{1}[0-9a-zA-Z\_./-]{2,50}[.]{1}[a-zA-Z]{2,5}"/>* *</xs:restriction>**</xs:simpleType>* |

## ОКАТО. Тип OKATOType

Тип для описания кода ОКАТО. Длина от 1 до 11 символов, разрешается использовать только цифры.

|  |
| --- |
| *<xs:simpleType name="OKATOType">* *<xs:annotation>* *<xs:documentation>Код ОКАТО</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:maxLength value="11"/>* *<xs:pattern value="\d{1,11}"/>* *</xs:restriction>**</xs:simpleType>* |

## Кадастровый номер. Тип CadastralNumberType

Тип CadastralNumberType используется для ввода кадастровых номеров объектов недвижимости.

|  |
| --- |
| *<xs:simpleType name="CadastralNumberType">* *<xs:annotation>* *<xs:documentation>Кадастровый номер земельного участка</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:maxLength value="44"/>* *<xs:pattern value="[0-9]{2,3}[/:][0-9/:]{1,40}"/>* *</xs:restriction>**</xs:simpleType>* |

## Серия свидетельства о рождении, паспорта СССР. Тип SovietStyleDocumentSeriesType

Используется для ввода строки, состоящей из римской цифры, тире и двух прописных букв кириллицы. Например, серия свидетельства о рождении.

|  |
| --- |
| *<xs:simpleType name="SovietStyleDocumentSeriesType">* *<xs:annotation>* *<xs:documentation>Серии документов, таких как свидетельства о рождении, советские паспорта, и т.д: римская цифра, тире, две прописные буквы кириллицы.</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:maxLength value="6"/>* *<xs:pattern value="[IVXLCDM]{1,3}[\-][А-Я]{2}"/>* *</xs:restriction>**</xs:simpleType>* |

## Строковые типы

### Семь цифр. Тип digits-7

Тип digits-7 используется для ввода строки, состоящей из семи цифр. Например, номера загранпаспорта.

|  |
| --- |
| *<xs:simpleType name="digits-7">* *<xs:annotation>* *<xs:documentation>Строка, состоящая из семи цифр. Например номер загранпаспорта.</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:length value="7"/>* *<xs:pattern value="\d{7}"/>* *</xs:restriction>**</xs:simpleType>* |

### Шесть цифр. Тип digits-6

Тип digits-6 используется для ввода строки, состоящей из шести цифр. Например, номера паспорта РФ.

|  |
| --- |
| *<xs:simpleType name="digits-6">* *<xs:annotation>* *<xs:documentation>Строка, состоящая из шести цифр. Например номер паспорта РФ.</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:length value="6"/>* *<xs:pattern value="\d{6}"/>* *</xs:restriction>**</xs:simpleType>* |

### Четыре цифры. Тип digits-4

Тип digits-4 используется для ввода строки, состоящей из четырех цифр. Например, серии паспорта РФ.

|  |
| --- |
| *<xs:simpleType name="digits-4">* *<xs:annotation>* *<xs:documentation>Строка, состоящая из четырёх цифр. Например серия паспорта РФ.</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:length value="4"/>* *<xs:pattern value="\d{4}"/>* *</xs:restriction>**</xs:simpleType>* |

### Две цифры. Тип digits-2

Тип digits-2 используется для ввода строки, состоящей из двух цифр. Например, серии заграничного паспорта.

|  |
| --- |
| *<xs:simpleType name="digits-2">* *<xs:annotation>* *<xs:documentation>Строка, состоящая из двух цифр. Например серия загранпаспорта.</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:length value="2"/>* *<xs:pattern value="\d{2}"/>* *</xs:restriction>**</xs:simpleType>* |

### Русскоязычный текст и цифры длиной до 500 символов. Тип rus-and-num-500

Тип rus-and-num-500 используется для ввода текста, содержащего кириллические буквы, дефисы, пробелы, точки, запятые и цифры, длиной до 500 символов. Например, описания места рождения.

|  |
| --- |
| *<xs:simpleType name="rus-and-num-500">* *<xs:annotation>* *<xs:documentation>Русский текст, длиной до 500 символов.</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:maxLength value="500"/>* *<xs:pattern value="[а-яА-ЯёЁ\-\s,.0-9]{1,500}"/>* *</xs:restriction>**</xs:simpleType>* |

### Русскоязычный текст и цифры длиной до 200 символов. Тип rus-and-num-200

Тип rus-and-num-200 используется для ввода текста, содержащего кириллические буквы, дефисы, пробелы, точки, запятые и цифры, длиной до 200 символов. Например, наименования органа, выдавшего документ.

|  |
| --- |
| *<xs:simpleType name="rus-and-num-200">* *<xs:annotation>* *<xs:documentation>Русский текст, длиной до 200 символов.</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:maxLength value="200"/>* *<xs:pattern value="[а-яА-ЯёЁ\-\s,.0-9]{1,200}"/>* *</xs:restriction>**</xs:simpleType>* |

### Русскоязычный текст и цифры длиной до 100 символов. Тип rus-and-num-100

Тип rus-and-num-100 используется для ввода текста, содержащего кириллические буквы, дефисы, пробелы, точки, запятые и цифры, длиной до 100 символов.

|  |
| --- |
| *<xs:simpleType name="rus-and-num-100">* *<xs:annotation>* *<xs:documentation>Русский текст, длиной до 100 символов.</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:maxLength value="100"/>* *<xs:pattern value="[а-яА-ЯёЁ\-\s,.0-9]{1,100}"/>* *</xs:restriction>**</xs:simpleType>* |

### Русскоязычный текст и цифры длиной до 50 символов. Тип rus-and-num-50

Тип rus-and-num-50 используется для ввода текста, содержащего кириллические буквы, дефисы, пробелы, точки, запятые и цифры, длиной до 50 символов.

|  |
| --- |
| *<xs:simpleType name="rus-and-num-50">* *<xs:annotation>* *<xs:documentation>Русский текст, длиной до 50 символов.</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:maxLength value="50"/>* *<xs:pattern value="[а-яА-ЯёЁ\-\s,.0-9]{1,50}"/>* *</xs:restriction>**</xs:simpleType>* |

### Русскоязычный текст длиной до 500 символов. Тип rus-500

Тип rus-500 используется для ввода текста, содержащего кириллические буквы, дефисы, пробелы, точки и запятые, длиной до 500 символов.

|  |
| --- |
| *<xs:simpleType name="rus-500">* *<xs:annotation>* *<xs:documentation>Русский текст, длиной до 500 символов, цифры не допускаются.</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:maxLength value="500"/>* *<xs:pattern value="[а-яА-ЯёЁ\-\s,.]{1,500}"/>* *</xs:restriction>**</xs:simpleType>* |

### Русскоязычный текст длиной до 200 символов. Тип rus-200

Тип rus-200 используется для ввода текста, содержащего кириллические буквы, дефисы, пробелы, точки и запятые, длиной до 200 символов.

|  |
| --- |
| *<xs:simpleType name="rus-200">* *<xs:annotation>* *<xs:documentation>Русский текст, длиной до 200 символов, цифры не допускаются.</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:maxLength value="200"/>* *<xs:pattern value="[а-яА-ЯёЁ\-\s,.]{1,200}"/>* *</xs:restriction>**</xs:simpleType>* |

### Русскоязычный текст длиной до 100 символов. Тип rus-100

Тип rus-100 используется для ввода текста, содержащего кириллические буквы, дефисы, пробелы, точки и запятые, длиной до 100 символов. Например, фамилии.

|  |
| --- |
| *<xs:simpleType name="rus-100">* *<xs:annotation>* *<xs:documentation>Русский текст, длиной до 100 символов, цифры не допускаются.</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:maxLength value="100"/>* *<xs:pattern value="[а-яА-ЯёЁ\-\s,.]{1,100}"/>* *</xs:restriction>**</xs:simpleType>* |

### Русскоязычный текст длиной до 50 символов. Тип rus-50

Тип rus-50 используется для ввода текста, содержащего кириллические буквы, дефисы, пробелы, точки и запятые, длиной до 50 символов.

|  |
| --- |
| *<xs:simpleType name="rus-50">* *<xs:annotation>* *<xs:documentation>Русский текст, длиной до 50 символов, цифры не допускаются.</xs:documentation>* *</xs:annotation>* *<xs:restriction base="xs:string">* *<xs:maxLength value="50"/>* *<xs:pattern value="[а-яА-ЯёЁ\-\s,.]{1,50}"/>* *</xs:restriction>**</xs:simpleType>* |

### Произвольный текст длиной до 500 символов. Тип string-500

Используется для ввода произвольного текста длиной до 500 символов.

|  |
| --- |
| *<xs:simpleType name="string-500">* *<xs:restriction base="xs:string">* *<xs:maxLength value="500"/>* *</xs:restriction>**</xs:simpleType>* |

### Произвольный текст длиной до 200 символов. Тип string-200

Используется для ввода произвольного текста длиной до 200 символов.

|  |
| --- |
| *<xs:simpleType name="string-200">* *<xs:restriction base="xs:string">* *<xs:maxLength value="200"/>* *</xs:restriction>**</xs:simpleType>* |

### Произвольный текст длиной до 100 символов. Тип string-100

Используется для ввода произвольного текста длиной до 100 символов.

|  |
| --- |
| *<xs:simpleType name="string-100">* *<xs:restriction base="xs:string">* *<xs:maxLength value="100"/>* *</xs:restriction>**</xs:simpleType>* |

### Произвольный текст длиной до 50 символов. Тип string-50

Используется для ввода произвольного текста длиной до 50 символов.

|  |
| --- |
| *<xs:simpleType name="string-50">* *<xs:restriction base="xs:string">* *<xs:maxLength value="50"/>* *</xs:restriction>**</xs:simpleType>* |

### Произвольный текст длиной до 20 символов. Тип string-20

Используется для ввода произвольного текста длиной до 20 символов. Например, серии документа с нерегламентированными реквизитами.

|  |
| --- |
| *<xs:simpleType name="string-20">* *<xs:restriction base="xs:string">* *<xs:maxLength value="20"/>* *</xs:restriction>**</xs:simpleType>* |

### Произвольный текст длиной до 10 символов. Тип string-10

Используется для ввода произвольного текста длиной до 10 символов.

|  |
| --- |
| *<xs:simpleType name="string-10">* *<xs:restriction base="xs:string">* *<xs:maxLength value="10"/>* *</xs:restriction>**</xs:simpleType>* |

## Электронная подпись в формате XMLDsig

Используется для передачи сведений об электронной подписи в формате XMLDsig

|  |
| --- |
| *<xs:complexType name="XMLDSigSignatureType">* *<xs:sequence>* *<xs:any namespace="*[http://www.w3.org/2000/09/xmldsig#](http://www.w3.org/2000/09/xmldsig)*" processContents="skip"> </xs:any>* *</xs:sequence>**</xs:complexType>* |