

## СТАНДАРТ ВОИС ST.97

### РЕКОМЕНДАЦИИ ПО ОБРАБОТКЕ ДАННЫХ ОБ ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ С ИСПОЛЬЗОВАНИЕМ JSON (JAVASCRIPT OBJECT NOTATION)

*Одобрено Комитетом по Стандартам ВОИС (КСВ)  
на своей десятой сессии 25 ноября 2022*

#### СОДЕРЖАНИЕ

СТАНДАРТ ВОИС ST.97 .....	1
СОДЕРЖАНИЕ .....	1
СОДЕРЖАНИЕ (ТАБЛИЦЫ) .....	1
1 ВВЕДЕНИЕ .....	2
2 ОПРЕДЕЛЕНИЯ И ТЕРМИНОЛОГИЯ .....	2
3 ОБЩИЕ ОБОЗНАЧЕНИЯ .....	2
3.1 Правила составления идентификаторов .....	3
3.2 Пример структуры данных JSON .....	3
4 ЦЕЛЬ .....	3
5 ОБЩИЕ ПРАВИЛА ПРОЕКТИРОВАНИЯ JSON .....	4
5.1 Обзор .....	4
5.2 Соглашения об именах JSON .....	4
6 ПРАВИЛА ПРОЕКТИРОВАНИЯ СХЕМЫ JSON (JSON SCHEMA) .....	5
6.1 Обзор .....	5
6.2 Модульность .....	6
6.3 Документация .....	11
6.4 Имя файла .....	12
6.5 Структурирование свойств схемы JSON .....	13
7 ПРАВИЛА ПРОЕКТИРОВАНИЯ КОНСТРУКЦИЙ JSON SCHEMA .....	14
7.1 Обзор .....	14
7.2 Свойства .....	14
7.3 Определения .....	14
7.4 Определения типов .....	14
7.5 Базовые (primitive) типы JSON .....	15
7.6 Списки кодов .....	15
7.7 Массивы .....	15
7.8 Объекты .....	16
8 ИДЕНТИФИКАТОРЫ СХЕМЫ JSON .....	16
8.1 Обзор .....	16
9 ПРАВИЛА ПРОЕКТИРОВАНИЯ ЭКЗЕМПЛЯРА JSON (JSON INSTANCE) .....	16
9.1 Порядок свойств .....	16
9.2 Проверка экземпляра JSON .....	17
10 ССЫЛКИ .....	17
Стандарты ВОИС .....	17
Стандарты и отраслевые спецификации .....	17
ПРИЛОЖЕНИЕ I - ПРАВИЛА ПРЕОБРАЗОВАНИЯ ИЗ ST.96 XSD В JSON СХЕМА И РЕКОМЕНДАЦИИ ПО ИСПОЛЬЗОВАНИЮ .....	18
ДОПОЛНЕНИЕ - ИНСТРУМЕНТ ПРЕОБРАЗОВАНИЯ XSD В JSON SCHEMA .....	45
ПРИЛОЖЕНИЕ II - СХЕМА JSON .....	46
ПРИЛОЖЕНИЕ III - ПРИМЕРЫ JSON .....	47
ПРИЛОЖЕНИЕ IV - СПИСОК АКРОНИМОВ И СОКРАЩЕНИЙ .....	48
ПРИЛОЖЕНИЕ V – ПРЕДСТАВЛЕНИЕ ТЕРМИНОВ .....	51

## СОДЕРЖАНИЕ (ТАБЛИЦЫ)

Таблица 1. Элементы документации заголовка схемы JSON .....	12
Таблица 2: Преобразование простых типов данных XSD .....	20
Таблица 3 Преобразование ограничивающих аспектов XSD (X — числовое значение ограничения) .....	41

### 1 ВВЕДЕНИЕ

Данный стандарт содержит рекомендации по проектированию, созданию или обновлению ресурсов нотации объектов JavaScript (JSON) для использования при регистрации, обработке, обмене или публикации всех типов данных интеллектуальной собственности (ИС). В настоящем стандарте также рассматриваются правила преобразования схем расширяемого языка разметки (XML) Стандарта ВОИС ST.96 (XSD) в схемы JSON, отвечающие вышеупомянутым рекомендациям.

Настоящий Стандарт предназначен для:

- Предоставления рекомендаций по проектированию и разработке наилучших методов представления данных ИС в формате JSON ;
- Обеспечения согласованности путем предоставления схем и экземпляров JSON на основе стандарта ВОИС ST.96 для обмена данными об ИС;
- Рекомендаций принципов проектирования для расширения предоставленных схем JSON или создания новых совместимых схем JSON; и
- Повышения эффективности обмена данными, способствуя переиспользованию ресурсов JSON ведомствами интеллектуальной собственности (ИПО), а также публично доступных данных.

### 2 ОПРЕДЕЛЕНИЯ И ТЕРМИНОЛОГИЯ

Для целей Стандарта используются следующие термины:

- Термин «ресурсы JSON» предназначен для обозначения любых компонентов, используемых для создания и работы с реализацией JSON в соответствии с настоящим Стандартом;
- Термины «объект/object», «тип объекта/object type», «свойство/property», «член/member», «имя свойства/property name», «значение свойства/property value», «тип свойства/property type», «ключевое слово/keyword» и «определение/definition» в настоящем Стандарте должны пониматься также как в документе JSON Schema Core, версия draft-2020-12<sup>1</sup>;
- Термин «конструкция/construct» в данном Стандарте следует интерпретировать как «строительный блок/building block», из которого строятся схемы JSON;
- Термин «глобальное определение» используется для определения, на которое могут ссылаться другие определения в той же схеме или определения в других схемах; и
- В данном Стандарте «ДОЛЖЕН», «НЕ ДОЛЖЕН», «СЛЕДУЕТ», «НЕ СЛЕДУЕТ», «РЕКОМЕНДУЕТСЯ», «МОЖЕТ» и «ФАКУЛЬТАТИВНО» следует понимать, как описано в RFC 2119<sup>2</sup>. Незаглавные формы этих слов используются в обычном английском (русском в переводе) смысле.

### 3 ОБЩИЕ ОБОЗНАЧЕНИЯ

В настоящем стандарте используются следующие обозначения:

- `<>`: указывает описательный термин-заполнитель, который при реализации будет заменен конкретным значением;
- `" "`: указывают, что текст, заключенный в кавычки, должен использоваться дословно;
- `{ }`: указывает, что элементы являются необязательными в реализации; и

<sup>1</sup>Версия схемы JSON может быть изменена, поскольку она не получила статус RFC; он не был принят рабочей группой IETF. Эта версия Стандарта основана на последней версии, т. е. 2020-12, доступной по адресу <https://json-schema.org/draft/2020-12/json-schema-core.html>.

<sup>2</sup> <https://www.ietf.org/rfc/rfc2119.txt>

- Шрифтом Courier New font записываются ключевые слова JSON, имена свойств JSON, а также элементы и атрибуты XSD.

### 3.1 Правила составления идентификаторов

Все правила проектирования являются нормативными. Правила проектирования обозначаются префиксом [JXX-*nn*].

- Значение «JXX» — это префикс следующих категорий типов правил:
  - (a) JGD для общих правил проектирования;
  - (b) JSD для правил проектирования схемы JSON;
  - (c) JCD для построения правил проектирования; и
  - (d) JID для правил проектирования экземпляров
- Значение «*nn*» указывает следующее доступное число в последовательности определенного типа правила. Следует отметить, что цифра не означает позицию правила, в частности, для нового правила. Новое правило будет помещено в соответствующий контекст. Например, идентификатор правила [JGD-10] определяет десятое общее правило проектирования. Правило [JGD-10] можно поместить между правилами [JGD-05] и [JGD-06] вместо [JGD-09], если это наиболее подходящее место для этого правила.
- Идентификатор удаленного правила будет сохранен до замены описания правила текстом «Удалено».

### 3.2 Пример структуры данных JSON

Примеры структур данных JSON отображаются в текстовых полях с использованием шрифта фиксированной ширины. Образец синтаксиса структуры данных JSON выделяется для удобства чтения.

## 4 ЦЕЛЬ

Данный Стандарт предназначен для описания ресурсов JSON, которые будут использоваться для подачи, публикации, обработки и обмена данными ИС и сопутствующей информацией. Настоящий стандарт предназначен для предоставления рекомендаций для ВИС и других организаций, которые имеют дело с данными и документами, касающимися патентов, товарных знаков, промышленных образцов, географических указаний и/или работ, не защищенных авторскими правами.

Этот стандарт призван предоставить руководство для ВИС и других организаций, которые создают или изменяют данные об ИС в виде ресурсов JSON. Соответствие этому стандарту требуется для обмена данными между ВИС с использованием ресурсов JSON, таких как схемы, примеры, сообщения и полезные данные для интерфейсов прикладного программирования (API). Правила проектирования и правила трансформации написаны с учетом правил и соглашений по правилам проектирования стандарта ВОИС ST.96. Однако правила проектирования и соглашения ST.96 не сопоставляются один в один с правилами и соглашениями проектирования JSON, и поэтому правила проектирования ST.96 дублируются и в некоторых случаях слегка изменены, где это применимо.

Настоящий Стандарт включает следующие Приложения:

- Приложение I : Правила преобразования XML-схем ST.96 в схемы JSON, которые содержат Дополнение: инструмент преобразования для преобразования XSD ST.96 в схемы JSON;
- Приложение II : Схемы JSON, преобразованные из XML-схем Стандарта ВОИС ST.96 версии 5.0 <sup>3</sup>;
- Приложение III : Примеры случаев JSON ;
- Приложение IV : Список акронимов и аббревиатур; и
- Приложение V : Представления терминов.

Настоящий Стандарт не включает:

- (a) Проблемы архитектуры программного обеспечения; и
- (b) Языки реализации.

<sup>3</sup> Преобразованные схемы JSON имеют такие же имена тегов и структуру данных, как определено в ST.96, Приложение III, включая XML-компоненты смешанного содержимого и внешние стандарты, т. е. MathML и Oasis Table, для взаимодействия с данными в формате ST.96.

## 5 ОБЩИЕ ПРАВИЛА ПРОЕКТИРОВАНИЯ JSON

### 5.1 Обзор

Раздел содержит общие высокоуровневые правила и рекомендации по проектированию JSON, которые применяются ко всему обмену данными JSON и усилиям по разработке JSON, а не к конкретному языку программирования, выполняющему маршаллинг/демаршаллинг данных в/из JSON. Общие правила и рекомендации, перечисленные ниже, обеспечивают общую основу для разработки схемы JSON, экземпляра JSON и структуры данных JSON для всех данных, включая данные ИС и данные, не относящиеся к ИС, например данные смешанного содержимого. Уровни вложенности СЛЕДУЕТ сводить к минимуму при создании новых схем JSON, экземпляров JSON и разработке структур данных JSON, которые не доступны в стандарте ВОИС ST.96 или которым не обязательно быть совместимым со ST.96.

### 5.2 Соглашения об именах JSON

Эти соглашения необходимы для обеспечения согласованности, единообразия и полноты именования и определения всех ресурсов JSON.

Соглашения об именах JSON основаны на рекомендациях и принципах, описанных в документе [ISO 11179](#), часть 5<sup>4</sup> — Принципы именования и идентификации. Имена объектов и имена свойств состоят из следующих терминов:

- *Object Class (Класс объекта)* относится к действию или объекту в бизнес-контексте и представляет собой логическую группировку или агрегацию данных (в логической модели данных), к которой принадлежит свойство. Класс объекта выражается термином класса объекта.
- *Property Term (Свойства термина)* идентифицирует характеристики класса объекта.
- *Qualifier Term (Термин-квалификатор)* — это слово или слова, которые помогают определить и отличить элемент данных от других связанных элементов данных и могут быть присоединены к термину класса объекта или термину свойства, если необходимо сделать имя уникальным.
- *Representation Term (Представление термина)* классифицирует формат элемента данных в соответствии с наборами типов. Следует использовать Представления терминов, перечисленные в Приложении V.

- [JGD-01] Имена типов объектов и свойств ДОЛЖНЫ состоять из слов английского языка с использованием основных вариантов написания английского языка, приведенных в Оксфордском словаре английского языка. Единственными допустимыми исключениями являются акронимы, аббревиатуры и другие сокращения слов, перечисленные в Приложении IV.
- [JGD-02] Для типа объекта и имен свойств СЛЕДУЕТ выбирать только существительные, прилагательные и глаголы в настоящем времени, за исключением акронимов, сокращений и других сокращений слов, перечисленных в Приложении IV.
- [JGD-03] Символы, используемые в именах свойств, ДОЛЖНЫ содержаться в следующем наборе: 'a-z, A-Z и 0-9'.
- [JGD-04] Максимальная длина имен типов объектов и свойств ДОЛЖНА быть не более 35 символов.
- [JGD-05] Имена типов объектов и свойств ДОЛЖНЫ быть лаконичными и не требующими пояснений.
- [JGD-06] Тип объекта и имена свойств ДОЛЖНЫ использовать соглашение о наименованиях в так называемом нижнем верблюжьем регистре (lowerCamelCase LCC). Например, "currencyCode": "EUR".
- [JGD-07] Имена типов объектов ДОЛЖНЫ использовать соглашение о наименованиях LCC и иметь суффикс Type. Например, applicantType.
- [JGD-08] Акронимы и сокращения, перечисленные в Приложении IV, ДОЛЖНЫ всегда использоваться вместо полного расширенного имени.
- [JGD-09] Акронимы и аббревиатуры ДОЛЖНЫ появляться, как указано в Приложении IV для имен свойств и типов объектов.
- [JGD-10] Термин класса объекта ДОЛЖЕН всегда иметь одно и то же семантическое значение во всем конкретном домене ИС, таком как патенты, товарные знаки, промышленные образцы, географические указания или авторское право, и МОЖЕТ состоять из более чем одного слова. Например, contactInformation.

<sup>4</sup> <https://www.iso.org/standard/60341.html>

- [JGD-11] Термин свойства в имени ДОЛЖЕН быть уникальным в контексте класса объектов, но МОЖЕТ повторно использоваться в разных классах объектов.
- [JGD-12] Термин-квалификатор МОЖЕТ быть присоединен к термину класса объекта или термину свойства, если необходимо сделать имя уникальным.
- [JGD-13] Когда имя содержит термин класса объекта, термин свойства и термин представления, термин класса объекта ДОЛЖЕН предшествовать термину свойства, а термин свойства ДОЛЖЕН предшествовать термину представления. Термин квалификатора ДОЛЖЕН предшествовать связанному термину класса объекта или термину свойства. Например, `claimTotalQuantity`.
- [JGD-14] Если термин свойства заканчивается тем же словом, что и термин представления (или эквивалентным словом), тогда термин представления ДОЛЖЕН быть удален.
- [JGD-15] Там, где требуется термин представления, Условия представления в Приложении V ДОЛЖНЫ использоваться для терминов представления в именах основных компонентов.
- [JGD-16] В домене ИС все типы объектов и имена свойств ДОЛЖНЫ быть уникальными.
- [JGD-17] Слово (слова) в имени СЛЕДУЕТ использовать в единственном числе, если само понятие не во множественном числе. Например, `totalMarkSeries`.
- [JGD-18] К имени свойства или типа объекта, которое содержит набор контекстуально связанных компонентов, СЛЕДУЕТ добавлять суффикс «Bag». Например, `emailAddressBag` представляет коллекцию элементов `emailAddress`.
- [JGD-19] Соединительные слова, такие как «and», «of» и «the», НЕ СЛЕДУЕТ использовать в именах типов объектов и свойств, если только они не являются частью деловой терминологии.
- [JGD-20] Тип объекта и имена свойств НЕ ДОЛЖНЫ быть переведены, изменены или заменены для каких-либо целей.
- [JGD-21] Тип объекта и имена свойств НЕ ДОЛЖНЫ ссылаться на номера статей и правил. Например, `PCTRule702C` для PCT.
- [JGD-22] Уровни вложенности СЛЕДУЕТ сводить к минимуму при создании новых схем JSON, экземпляров JSON и фрагментов структур данных JSON, которые недоступны в стандарте BOIS ST.96.
- [JGD-23] Для нового типа объекта и имен свойств, которые не определены в Стандарте BOIS ST.96 или не предполагается, что окажутся совместимы с именем компонента, которое будет определено в ST.96, СЛЕДУЕТ использовать встроенные описательные термины или имена, а не использовать общие или короткие типы объектов или имена свойств, которые являются объектами с одним свойством. Например, вместо `"inventor":{"fullName":"Thomas Edison"}` предпочтительнее использовать `"inventorFullName":"Thomas Edison"`.

## 6 ПРАВИЛА ПРОЕКТИРОВАНИЯ СХЕМЫ JSON (JSON SCHEMA)

### 6.1 Обзор

Схема JSON описывает структуру экземпляра JSON, который выражает ограничения на структуру и содержимое документа. Данный стандарт должен быть приведен в соответствие с отраслевой спецификацией схемы JSON. Последней версией, доступной на момент публикации настоящего Стандарта, является проект 2020-12<sup>5</sup>, и данная версия Стандарта соответствует данному проекту спецификации.

- [JSD-01] Схемы JSON ДОЛЖНЫ соответствовать спецификациям схемы JSON под названием: JSON Schema Core (базовая схема JSON), версия 2020-12, доступной по адресу <https://json-schema.org/latest/json-schema-core.html>, и JSON Schema Validation (Проверка схемы JSON), версия 2020-12, доступной по адресу <https://json-schema.org/latest/json-schema-validation.html>.
- [JSD-02] Схемы JSON ДОЛЖНЫ указывать, что они соответствуют версии 2020-12 схемы JSON, используя ключевое слово `$schema` со значением «<https://json-schema.org/draft/2020-12/schema>».

#### Пример: указание версии схемы JSON

```
"$schema": "https://json-schema.org/draft/2020-12/schema"
```

Схема должна использовать кодировку UTF-8 для максимальной совместимости.

<sup>5</sup> <https://json-schema.org/draft/2020-12/schema>

[JSD-03] Схемы JSON ДОЛЖНЫ использовать набор символов ISO/IEC 10646 — UCS — Unicode. Кодировка UTF-8 ДОЛЖНА использоваться для кодирования символов Unicode.

## 6.2 Модульность

Модульность позволяет создавать компоненты схемы для поддержки гибкости дизайна и повторного использования. При проектировании рекомендуется избегать определения всех свойств и логических компонентов в единой монолитной схеме JSON, что препятствует возможности совместного использования и повторного использования отдельных свойств или логических компонентов, определенных как группа в схеме.

Ниже приведена схема, которая **не** придерживается принципа модульности. Подобное НЕ рекомендуется настоящим Стандартом.

### applicationNumber.json (неверный пример составного документа схемы)

```
"$id" : "applicationNumber.json",
"$schema" : "https://json-schema.org/draft/2020-12/schema",
"type" : "object",
"additionalProperties" : false,
"properties" : {
  "applicationNumber" : {
    "$ref" : "#/$defs/applicationNumber"
  }
},
"required" : [ "applicationNumber" ],
"$defs" : {
  "applicationNumber" : {
    "description" : "Description: Numbers used by IPOs in order to identify each application received; Version: V5_0",
    "type" : "object",
    "additionalProperties" : false,
    "properties" : {
      "ipOfficeCode" : {
        "anyOf" : [ {
          "type" : "string",
          "enum" : [ "AD", "AE", "AF", "AG", "AI", "AL", "AM", "AO", "AP", "AR", "AT", "AU", "AW", "AZ", "BA", "BB", "BD", "BE",
"BF", "BG", "BH", "BI", "BJ", "BM", "BN", "BO", "BQ", "BR", "BS", "BT", "BV", "BW", "BX", "BY", "BZ", "CA", "CD", "CF", "CG", "CH",
"CI", "CK", "CL", "CM", "CN", "CO", "CR", "CU", "CV", "CW", "CY", "CZ", "DE", "DJ", "DK", "DM", "DO", "DZ", "EA", "EC", "EE", "EG",
"EH", "EM", "EP", "ER", "ES", "ET", "EU", "FI", "FJ", "FK", "FO", "FR", "GA", "GB", "GC", "GD", "GE", "GG", "GH", "GI", "GL", "GM",
"GN", "GQ", "GR", "GS", "GT", "GW", "GY", "HK", "HN", "HR", "HT", "HU", "IB", "ID", "IE", "IL", "IM", "IN", "IQ", "IR", "IS", "IT",
"JE", "JM", "JO", "JP", "KE", "KG", "KH", "KI", "KM", "KN", "KP", "KR", "KW", "KY", "KZ", "LA", "LB", "LC", "LI", "LK", "LR", "LS",
"LT", "LU", "LV", "LY", "MA", "MC", "MD", "ME", "MG", "MK", "ML", "MM", "MN", "MO", "MP", "MR", "MS", "MT", "MU", "MV", "MW", "MX",
"MY", "MZ", "NA", "NE", "NG", "NI", "NL", "NO", "NP", "NZ", "OA", "OM", "PA", "PE", "PG", "PH", "PK", "PL", "PW", "PY",
"QA", "QZ", "RO", "RS", "RU", "RW", "SA", "SB", "SC", "SD", "SE", "SG", "SH", "SI", "SK", "SL", "SM", "SN", "SO", "SR", "SS", "ST",
"SV", "SX", "SY", "SZ", "TC", "TD", "TG", "TH", "TJ", "TL", "TM", "TN", "TO", "TR", "TT", "TV", "TW", "TZ", "UA", "UG", "US", "UY",
"UZ", "VA", "VC", "VE", "VG", "VN", "VO", "WS", "XN", "XU", "XV", "XX", "YE", "ZA", "ZM", "ZW" ],
          "description" : "Description: This code list is inline with WIPO Standard ST.3 (two-letter codes for the representation
of states, other entities and organizations) published on September, 2019.; Version: V5_0; AD: Andorra; AE: United Arab Emirates; AF:
Afghanistan; AG: Antigua And Barbuda; AI: Anguilla; AL: Albania; AM: Armenia; AO: Angola; AP: African Regional Intellectual Property
Organization (ARIPO); AR: Argentina; AT: Austria; AU: Australia; AW: Aruba; AZ: Azerbaijan; BA: Bosnia and Herzegovina; BB: Barbados;
BD: Bangladesh; BE: Belgium; BF: Burkina Faso; BG: Bulgaria; BH: Bahrain; BI: Burundi; BJ: Benin; BM: Brunei Darussalam;
BO: Bolivia (Plurinational State of); BQ: Bonaire, Sint Eustatius and Saba; BR: Brazil; BS: Bahamas; BT: Bhutan; BV: Bouvet Island;
BW: Botswana; BX: Benelux Office for Intellectual Property (BOIP); BY: Belarus; BZ: Belize; CA: Canada; CD: Democratic Republic of
the Congo; CF: Central African Republic; CG: Congo; CH: Switzerland; CI: Côte D'Ivoire; CK: Cook Islands; CL: Chile; CM: Cameroon;
CN: China; CO: Colombia; CR: Costa Rica; CU: Cuba; CV: Cabo Verde; CW: Curaçao; CY: Cyprus; CZ: Czech Republic; DE: Germany; DJ:
Djibouti; DK: Denmark; DM: Dominica; DO: Dominican Republic; DZ: Algeria; EA: Eurasian Patent Organization (EAPO); EC: Ecuador; EE:
Estonia; EG: Egypt; EH: Western Sahara; EM: European Union Intellectual Property Office (EUIPO); EP: European Patent Office (EPO);
ER: Eritrea; ES: Spain; ET: Ethiopia; EU: European Union; FI: Finland; FJ: Fiji; FK: Falkland Islands (Malvinas); FO: Faroe Islands;
FR: France; GA: Gabon; GB: United Kingdom; GC: Patent Office of the Cooperation Council for the Arab States of the Gulf (GCC Patent
Office); GD: Grenada; GE: Georgia; GG: Guernsey; GH: Ghana; GI: Gibraltar; GL: Greenland; GM: Gambia; GN: Guinea; GQ: Equatorial
Guinea; GR: Greece; GS: South Georgia and South Sandwich Islands; GT: Guatemala; GW: Guinea-Bissau; GY: Guyana; HK: Hong Kong, China;
HN: Honduras; HR: Croatia; HT: Haiti; HU: Hungary; IB: International Bureau of the World Intellectual Property Organization (WIPO);
ID: Indonesia; IE: Ireland; IL: Israel; IM: Isle of Man; IN: India; IQ: Iraq; IR: Iran, Islamic Republic of; IS: Iceland; IT: Italy;
JE: Jersey; JM: Jamaica; JO: Jordan; JP: Japan; KE: Kenya; KG: Kyrgyzstan; KH: Cambodia; KI: Kiribati; KM: Comoros; KN: Saint Kitts
and Nevis; KP: Democratic People's Republic of Korea; KR: Republic of Korea; KW: Kuwait; KY: Cayman Islands; KZ: Kazakhstan; LA: Lao
People's Democratic Republic; LB: Lebanon; LC: Saint Lucia; LI: Liechtenstein; LK: Sri Lanka; LR: Liberia; LS: Lesotho; LT:
Lithuania; LU: Luxembourg; LV: Latvia; LY: Libya; MA: Morocco; MC: Monaco; MD: Republic of Moldova; ME: Montenegro; MG: Madagascar;
MK: North Macedonia; ML: Mali; MM: Myanmar; MN: Mongolia; MO: Macao, China; MP: Northern Mariana Islands; MR: Mauritania; MS:
Montserrat; MT: Malta; MU: Mauritius; MV: Maldives; MW: Malawi; MX: Mexico; MY: Malaysia; MZ: Mozambique; NA: Namibia; NE: Niger; NG:
Nigeria; NI: Nicaragua; NL: Netherlands; NO: Norway; NP: Nepal; NR: Nauru; NZ: New Zealand; OA: African Intellectual Property
Organization (OAPI); OM: Oman; PA: Panama; PE: Peru; PG: Papua New Guinea; PH: Philippines; PK: Pakistan; PL: Poland; PT: Portugal;
PW: Palau; PY: Paraguay; QA: Qatar; QZ: Community Plant Variety Office (European Community) (CPVO); RO: Romania; RS: Serbia; RU:
Russian Federation; RW: Rwanda; SA: Saudi Arabia; SB: Solomon Islands; SC: Seychelles; SD: Sudan; SE: Sweden; SG: Singapore; SH:
Saint Helena, Ascension and Tristan da Cunha; SI: Slovenia; SK: Slovakia; SL: Sierra Leone; SM: San Marino; SN: Senegal; SO: Somalia;
SR: Suriname; SS: South Sudan; ST: Sao Tome and Principe; SV: El Salvador; SX: Sint Maarten (Dutch part); SY: Syrian Arab Republic;
SZ: Eswatini; TC: Turks and Caicos Islands; TD: Chad; TG: Togo; TH: Thailand; TJ: Tajikistan; TL: Timor-Leste; TM: Turkmenistan; TN:
Tunisia; TO: Tonga; TR: Turkey; TT: Trinidad and Tobago; TV: Tuvalu; TW: Taiwan, Province of China; TZ: United Republic of Tanzania;
UA: Ukraine; UG: Uganda; US: United States of America; UY: Uruguay; UZ: Uzbekistan; VA: Holy See; VC: Saint Vincent and the
Grenadines; VE: Venezuela (Bolivian Republic of); VG: British Virgin Islands; VN: Viet Nam; VU: Vanuatu; WO: World Intellectual
Property Organization (WIPO) (International Bureau of); WS: Samoa; XN: Nordic Patent Institute (NPI); XU: International Union for the
Protection of New Varieties of Plants (UPOV); XV: Visegrad Patent Institute (VPI); XX: Unknown states, other entities or
organizations; YE: Yemen; ZA: South Africa; ZM: Zambia; ZW: Zimbabwe"
        ], {

```

```
{
  "type": "string",
  "enum": [ "AN", "CS", "DL", "DD", "DT", "RH", "SU", "YD", "YU" ],
  "description": "Version: V5_0; AN: Netherlands Antilles; CS: Czechoslovakia; DL: German Democratic Republic; DD: German Democratic Republic; DT: Federal Republic of Germany; RH: Southern Rhodesia; SU: Soviet Union; YD: Democratic Yemen; YU: Yugoslavia/Serbia and Montenegro"
},
{
  "st13ApplicationNumber": {
    "type": "string",
    "pattern": "\\d{2}\\d{4}\\d{9}",
    "description": "Description: Application number format recommended in WIPO Standard ST.13. The sequence of indispensable elements in the application number format is IP type (2 digits), year designation (4 digits) and serial number (9 digits).; Version: V5_0"
  },
  "applicationNumberText": {
    "type": "string",
    "description": "Description: Free format of application number; Version: V5_0"
  }
},
{
  "oneOf": [ {
    "required": [ "st13ApplicationNumber" ]
  }, {
    "required": [ "applicationNumberText" ]
  } ]
}
}
```

Предпочтительный подход к проектированию заключается в разбиении данных на набор небольших компонентов, представленных модулями схемы, что показано в новой схеме applicationNumber ниже. Эта схема JSON построена на более мелких модулях схемы JSON, индивидуально определенных в их собственных схемах.

#### applicationNumber.json (пример модульной схемы)

```
{
  "$id": "applicationNumber.json",
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "applicationNumber": {
      "$ref": "#/$defs/applicationNumber"
    }
  },
  "required": [ "applicationNumber" ],
  "$defs": {
    "applicationNumber": {
      "$ref": "applicationNumberType.json#/$defs/applicationNumberType",
      "description": "Description: Numbers used by IPOs in order to identify each application received; Version: V5_0"
    }
  }
}
```

**applicationNumberType.json (пример модульной схемы, продолжение)**

```
{
  "$id" : "applicationNumberType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$defs" : {
    "applicationNumberType" : {
      "description" : "Version: V5_0",
      "type" : "object",
      "additionalProperties" : false,
      "properties" : {
        "ipOfficeCode" : {
          "$ref" : "ipOfficeCode.json#/$defs/ipOfficeCode"
        },
        "st13ApplicationNumber" : {
          "$ref" : "st13ApplicationNumber.json#/$defs/st13ApplicationNumber"
        },
        "applicationNumberText" : {
          "$ref" : "applicationNumberText.json#/$defs/applicationNumberText"
        }
      },
      "oneOf" : [ {
        "required" : [ "st13ApplicationNumber" ]
      }, {
        "required" : [ "applicationNumberText" ]
      } ]
    }
  }
}
```

**ipOfficeCode.json (пример модульной схемы, продолжение)**

```
{
  "$id" : "ipOfficeCode.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "type" : "object",
  "additionalProperties" : false,
  "properties" : {
    "ipOfficeCode" : {
      "$ref" : "#/$defs/ipOfficeCode"
    }
  },
  "required" : [ "ipOfficeCode" ],
  "$defs" : {
    "ipOfficeCode" : {
      "$ref" : "extendedWIPOST3CodeType.json#/$defs/extendedWIPOST3CodeType",
      "description" : "Description: Two-letter alphabetic codes which represent the names of states, other entities and intergovernmental organizations the legislation of which provides for the protection of IP rights or which organizations are acting in the framework of a treaty in the field of IP; Version: V5_0"
    }
  }
}
```



**extendedWIPOST3CodeType.json (Продолжение примера модульной схемы)**

```
{
  "$id" : "extendedWIPOST3CodeType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$defs" : {
    "extendedWIPOST3CodeType" : {
      "description" : "Version: V5_0",
      "anyOf" : [ {
        "$ref" : "wipoST3CodeType.json#/$defs/wipoST3CodeType"
      }, {
        "$ref" : "wipoFormerST3CodeType.json#/$defs/wipoFormerST3CodeType"
      } ]
    }
  }
}
```

**wipoST3CodeType.json (Продолжение примера модульной схемы)**

```
{
  "$id" : "wipoST3CodeType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$defs" : {
    "wipoST3CodeType" : {
      "description" : "Description: This code list is inline with WIPO Standard ST.3 (two-letter codes for the representation of states, other entities and organizations) published on September, 2019.; Version: V5_0; AD: Andorra; AE: United Arab Emirates; AF: Afghanistan; AG: Antigua And Barbuda; AI: Anguilla; AL: Albania; AM: Armenia; AO: Angola; AP: African Regional Intellectual Property Organization (ARIPO); AR: Argentina; AT: Austria; AU: Australia; AW: Aruba; AZ: Azerbaijan; BA: Bosnia and Herzegovina; BB: Barbados; BD: Bangladesh; BE: Belgium; BF: Burkina Faso; BG: Bulgaria; BH: Bahrain; BI: Burundi; BJ: Benin; BM: Bermuda; BN: Brunei Darussalam; BO: Bolivia (Plurinational State of); BQ: Bonaire, Sint Eustatius and Saba; BR: Brazil; BS: Bahamas; BT: Bhutan; BV: Bouvet Island; BW: Botswana; BX: Benelux Office for Intellectual Property (BOIP); BY: Belarus; BZ: Belize; CA: Canada; CD: Democratic Republic of the Congo; CF: Central African Republic; CG: Congo; CH: Switzerland; CI: Côte D'Ivoire; CK: Cook Islands; CL: Chile; CM: Cameroon; CN: China; CO: Colombia; CR: Costa Rica; CU: Cuba; CV: Cabo Verde; CW: Curaçao; CY: Cyprus; CZ: Czech Republic; DE: Germany; DJ: Djibouti; DK: Denmark; DM: Dominica; DO: Dominican Republic; DZ: Algeria; EA: Eurasian Patent Organization (EAPo); EC: Ecuador; EE: Estonia; EG: Egypt; EH: Western Sahara; EM: European Union Intellectual Property Office (EUIPO); EP: European Patent Office (EPO); ER: Eritrea; ES: Spain; ET: Ethiopia; EU: European Union; FI: Finland; FJ: Fiji; FK: Falkland Islands (Malvinas); FO: Faroe Islands; FR: France; GA: Gabon; GB: United Kingdom; GC: Patent Office of the Cooperation Council for the Arab States of the Gulf (GCC Patent Office); GD: Grenada; GE: Georgia; GG: Guernsey; GH: Ghana; GI: Gibraltar; GL: Greenland; GM: Gambia; GN: Guinea; GQ: Equatorial Guinea; GR: Greece; GS: South Georgia and South Sandwich Islands; GT: Guatemala; GW: Guinea-Bissau; GY: Guyana; HK: Hong Kong, China; HN: Honduras; HR: Croatia; HT: Haiti; HU: Hungary; IB: International Bureau of the World Intellectual Property Organization (WIPO); ID: Indonesia; IE: Ireland; IL: Israel; IM: Isle of Man; IN: India; IQ: Iraq; IR: Iran, Islamic Republic of; IS: Iceland; IT: Italy; JE: Jersey; JM: Jamaica; JO: Jordan; JP: Japan; KE: Kenya; KG: Kyrgyzstan; KH: Cambodia; KI: Kiribati; KM: Comoros; KN: Saint Kitts and Nevis; KP: Democratic People's Republic of Korea; KR: Republic of Korea; KW: Kuwait; KY: Cayman Islands; KZ: Kazakhstan; LA: Lao People's Democratic Republic; LB: Lebanon; LC: Saint Lucia; LI: Liechtenstein; LK: Sri Lanka; LR: Liberia; LS: Lesotho; LT: Lithuania; LU: Luxembourg; LV: Latvia; LY: Libya; MA: Morocco; MC: Monaco; MD: Republic of Moldova; ME: Montenegro; MG: Madagascar; MK: North Macedonia; ML: Mali; MM: Myanmar; MN: Mongolia; MO: Macao, China; MP: Northern Mariana Islands; MR: Mauritania; MS: Montserrat; MT: Malta; MU: Mauritius; MV: Maldives; MW: Malawi; MX: Mexico; MY: Malaysia; MZ: Mozambique; NA: Namibia; NE: Niger; NG: Nigeria; NI: Nicaragua; NL: Netherlands; NO: Norway; NP: Nepal; NR: Nauru; NZ: New Zealand; OA: African Intellectual Property Organization (OAPI); OM: Oman; PA: Panama; PE: Peru; PG: Papua New Guinea; PH: Philippines; PK: Pakistan; PL: Poland; PT: Portugal; PW: Palau; PY: Paraguay; QA: Qatar; QZ: Community Plant Variety Office (European Community) (CPVO); RO: Romania; RS: Serbia; RU: Russian Federation; RW: Rwanda; SA: Saudi Arabia; SB: Solomon Islands; SC: Seychelles; SD: Sudan; SE: Sweden; SG: Singapore; SH: Saint Helena, Ascension and Tristan da Cunha; SI: Slovenia; SK: Slovakia; SL: Sierra Leone; SM: San Marino; SN: Senegal; SO: Somalia; SR: Suriname; SS: South Sudan; ST: Sao Tome and Principe; SV: El Salvador; SX: Sint Maarten (Dutch part); SY: Syrian Arab Republic; SZ: Eswatini; TC: Turks and Caicos Islands; TD: Chad; TG: Togo; TH: Thailand; TJ: Tajikistan; TL: Timor-Leste; TM: Turkmenistan; TN: Tunisia; TO: Tonga; TR: Turkey; TT: Trinidad and Tobago; TV: Tuvalu; TW: Taiwan, Province of China; TZ: United Republic of Tanzania; UA: Ukraine; UG: Uganda; US: United States of America; UY: Uruguay; UZ: Uzbekistan; VA: Holy See; VC: Saint Vincent and the Grenadines; VE: Venezuela (Bolivian Republic of); VG: British Virgin Islands; VN: Viet Nam; VU: Vanuatu; WO: World Intellectual Property Organization (WIPO) (International Bureau of); WS: Samoa; XN: Nordic Patent Institute (NPI); XU: International Union for the Protection of New Varieties of Plants (UPOV); XZ: Visegrad Patent Institute (VPI); XX: Unknown states, other entities or organizations; YE: Yemen; ZA: South Africa; ZM: Zambia; ZW: Zimbabwe",
      "type" : "string",
      "enum" : [ "AD", "AE", "AF", "AG", "AI", "AL", "AM", "AO", "AP", "AR", "AT", "AU", "AW", "AZ", "BA", "BB", "BD", "BE", "BF", "BG", "BH", "BI", "BJ", "BM", "BN", "BO", "BQ", "BR", "BS", "BT", "BV", "BW", "BX", "BY", "BZ", "CA", "CD", "CF", "CG", "CH", "CI", "CK", "CL", "CM", "CN", "CO", "CR", "CU", "CV", "CW", "CY", "CZ", "DE", "DJ", "DK", "DM", "DO", "DZ", "EA", "EC", "EE", "EG", "EH", "EM", "EP", "ER", "ES", "ET", "EU", "FI", "FJ", "FK", "FO", "FR", "GA", "GB", "GC", "GD", "GE", "GG", "GH", "GI", "GL", "GM", "GN", "GQ", "GR", "GS", "GT", "GW", "GY", "HK", "HN", "HR", "HT", "HU", "IB", "ID", "IE", "IL", "IM", "IN", "IQ", "IR", "IS", "IT", "JE", "JM", "JO", "JP", "KE", "KG", "KH", "KI", "KM", "KN", "KP", "KR", "KW", "KY", "KZ", "LA", "LB", "LC", "LI", "LK", "LR", "LS", "LT", "LU", "LV", "LY", "MA", "MC", "MD", "ME", "MG", "MK", "ML", "MM", "MN", "MO", "MP", "MR", "MS", "MT", "MU", "MV", "MW", "MX", "MY", "MZ", "NA", "NE", "NG", "NI", "NL", "NO", "NP", "NR", "NZ", "OA", "OM", "PA", "PE", "PG", "PH", "PK", "PL", "PT", "PW", "PY", "QA", "QZ", "RO", "RS", "RW", "SA", "SB", "SC", "SD", "SE", "SG", "SH", "SI", "SK", "SL", "SM", "SN", "SO", "SR", "SS", "ST", "SV", "SX", "SY", "SZ", "TC", "TD", "TG", "TH", "TJ", "TL", "TM", "TN", "TO", "TR", "TT", "TV", "TW", "TZ", "UA", "UG", "US", "UY", "UZ", "VA", "VC", "VE", "VG", "VN", "VU", "WO", "WS", "XN", "XU", "XZ", "XX", "YE", "ZA", "ZM", "ZW" ]
    }
  }
}
```

**st13ApplicationNumber.json (Пример модульной схемы, продолжение)**

```
{
  "$id" : "st13ApplicationNumber.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "type" : "object",
  "additionalProperties" : false,
  "properties" : {
    "st13ApplicationNumber" : {
      "$ref" : "#/$defs/st13ApplicationNumber"
    }
  },
  "required" : [ "st13ApplicationNumber" ],
  "$defs" : {
    "st13ApplicationNumber" : {
      "$ref" : "st13ApplicationNumberType.json#/$defs/st13ApplicationNumberType",
      "description" : "Description: Application number format recommended in WIPO Standard ST.13. The sequence of indispensable elements in the application number format is IP type (2 digits), year designation (4 digits) and serial number (9 digits).; Version: V5_0"
    }
  }
}
```

**wipoFormerST3CodeType.json (Пример модульной схемы, продолжение)**

```
{
  "$id" : "wipoFormerST3CodeType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$defs" : {
    "wipoFormerST3CodeType" : {
      "description" : "Version: V5_0; AN: Netherlands Antilles; CS: Czechoslovakia; DL: German Democratic Republic; DD: German Democratic Republic; DT: Federal Republic of Germany; RH: Southern Rhodesia; SU: Soviet Union; YD: Democratic Yemen; YU: Yugoslavia/ Serbia and Montenegro",
      "type" : "string",
      "enum" : [ "AN", "CS", "DL", "DD", "DT", "RH", "SU", "YD", "YU" ]
    }
  }
}
```

**st13ApplicationNumberType.json (Пример модульной схемы, продолжение)**

```
{
  "$id" : "st13ApplicationNumberType.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "$defs" : {
    "st13ApplicationNumberType" : {
      "description" : "Version: V5_0",
      "type" : "string",
      "pattern" : "\\d{2}\\d{4}\\d{9}"
    }
  }
}
```

**applicationNumberText.json (Пример модульной схемы, продолжение)**

```
{
  "$id" : "applicationNumberText.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "type" : "object",
  "additionalProperties" : false,
  "properties" : {
    "applicationNumberText" : {
      "$ref" : "#/$defs/applicationNumberText"
    }
  },
  "required" : [ "applicationNumberText" ],
  "$defs" : {
    "applicationNumberText" : {
      "type" : "string",
      "description" : "Description: Free format of application number; Version: V5_0"
    }
  }
}
```

В JSON схемах следует использовать ключевое слово "\$defs" для создания глобальных определений свойств и их содержимого, которые можно использовать повторно, как показано в приведенном выше примере. Это примерно эквивалентно созданию объявлений глобальных элементов и именованных типов в схеме XML.

- [JSD-04] В JSON схемах СЛЕДУЕТ использовать ключевое слово "\$defs", которое включает многократно используемое определение для каждого свойства и типа свойства.
- [JSD-05] Разработчики ДОЛЖНЫ использовать существующие схемы JSON, определенные в Приложении II к настоящему проекту стандарта, где это применимо, прежде, чем создавать новые схемы JSON.
- [JSD-06] Разработчикам СЛЕДУЕТ создавать новые схемы JSON только после того, как будет установлено, что ни одна из существующих схем JSON не описывает данную конструкцию адекватно.

### 6.3 Документация

Схемам JSON следует быть самоинформативными. Разработчикам следует стремиться давать значимые имена для конструкций JSON. Кроме того, схеме JSON следует иметь документацию, описывающую схему и конструкции JSON.

Чтобы способствовать повторному использованию, оставляя его общим, в схеме JSON не следует предоставлять документацию по конкретным деталям реализации системы.

- [JSD-07] В документации НЕ СЛЕДУЕТ описывать детали реализации или другую информацию, не имеющую прямого отношения к объяснению конструкции.

Заголовок схемы JSON позволяет разработчику схемы легко определить назначение, использование и содержимое схемы. Эта информация очень полезна, когда разработчику схемы необходимо выбрать схему для использования в качестве шаблона при создании другой схемы.

- [JSD-08] В схемы JSON СЛЕДУЕТ включать документацию заголовка схемы JSON с использованием ключевого слова "description" (описание).
- [JSD-09] Элементы, перечисленные в таблице 1 ниже, СЛЕДУЕТ включать в раздел заголовка всех схем JSON.

Таблица 1. Элементы документации заголовка схемы JSON

Название элемента заголовка	Описание	Обязательно/необязательно
Description (Описание)	Простое текстовое описание информации, описываемой схемой	Требуется, за исключением схем JSON, происходящих из простых XSD Type ST.96, для которых не предоставлено описание, например <code>DateTime</code> .
Version (Версия)	Основной и дополнительный номер версии схемы	Необходимый
SchemaCreatedDate	Дата создания схемы	Необязательный
SchemaLastModifiedDate	Дата последнего изменения схемы	Необязательный
SchemaContactPoint	Название организации, к которой можно обратиться с вопросами о схеме	Необязательный
SchemaReleaseNoteURL	Место, где публикуются заметки о выпуске схемы	Необязательный

[JSD-10] Элементы документации заголовка, такие как Опубликовано и номер версии выше, СЛЕДУЕТ отделять точкой с запятой, с допустимыми пробелами после точки с запятой, и приводить значение, связанное с ключевым словом "description" (описание). Если значение для элемента заголовка недоступно, следует включить только метку, как показано в следующем примере:

Пример заголовка документации для <code>applicationBody</code> (схема уровня документа)
<pre>"description" : "Description: Body of a patent application; Version: V5_0; SchemaCreatedDate: 2012-07-13; SchemaLastModifiedDate: 2021-10-01; SchemaContactPoint: xml.standards@wipo.int; SchemaReleaseNoteURL: <a href="http://www.wipo.int/standards/XMLSchema/ST96/V5_0/ReleaseNotes.pdf">http://www.wipo.int/standards/XMLSchema/ST96/V5_0/ReleaseNotes.pdf</a>"</pre>

Пример заголовка документации для <code>ipOfficeCode</code> (схема не на уровне документа)
<pre>"description" : "Description: Two-letter alphabetic codes which represent the names of states, other entities and intergovernmental organizations the legislation of which provides for the protection of IP rights or which organizations are acting in the framework of a treaty in the field of IP; Version: V5_0"</pre>

Пример заголовка документации для <code>appellatBodyCategoryType.json</code> (схема определений типа перечисления - Enumeration Type definitions Schema)
<pre>"description" : "Version: V5_0; Office appeal board: Appeal board within the IP office; Court: Court; Appeal Court: Second instance court; Supreme Court: Highest appellate court"</pre>

Пример заголовка документации для схемы определений типов Type definitions Schema
<pre>"description" : "Version: V5_0"</pre>

#### 6.4 Имя файла

Правила из ST.96 соблюдаются для имени файла схемы JSON, за исключением того, что они должны быть LCC.

Имена файлов схем и имена схем часто являются парными. Имена файлов схемы основаны на соответствующих именах схемы. Например, имя файла `postalAddressType.json` происходит от имени схемы `postalAddressType`. Таким образом, соглашения об именовании файлов схемы связаны с правилами для соглашений об именовании JSON в этом стандарте.

Файл схемы МОЖЕТ иметь информацию о версии. Схема, находящаяся на стадии проекта, может быть изменена. Черновые схемы должны быть обозначены как таковые в имени файла схемы, поставив букву «D» и номер редакции.

- [JSD-11] Символы, используемые в именах файлов схемы, ДОЛЖНЫ принадлежать к следующему набору: a-z, A-Z, 0-9, знак подчеркивания “\_” и точка “.”.
- [JSD-12] Имя файла схемы ДОЛЖНО состоять из двух обязательных частей с одним разделителем и необязательной информации о версии с двумя дополнительными разделителями, например: `<имя компонента>{"_""V"<основной номер версии>""<дополнительный номер версии>"}`. `"<расширение файла> ;`. Например, `emailAddressType.json`, `languageCode.json`, `applicationBody_V1_0.json`.
- [JSD-13] Имя файла черновика (проекта) схемы ДОЛЖНО состоять из четырех обязательных частей с двумя разделителями и необязательной информации о версии с двумя дополнительными разделителями, например: `<имя компонента>{"_""V"<основной номер версии>""<дополнительный номер версии>"}` `_"D"<номер версии>""<расширение файла>`, например, `tradeApplication_V1_1_D1.json`. Если черновик схемы основан на существующей схеме и содержит информацию о версии в имени файла, основному и вспомогательному номеру версий в имени файла черновика схемы СЛЕДУЕТ совпадать с номерами, указанными в файле схемы, на котором основана черновик схемы. Если черновик схемы является новым, основному номеру версии в имени файла черновика схемы СЛЕДУЕТ совпадать с номером, указанным в соответствующем домене ИС, а вспомогательному номеру версии в черновике файла схемы СЛЕДУЕТ равняться нулю «0».

## 6.5 Структурирование свойств схемы JSON

JSON схемам следует иметь свойство `"type": "object"`, чтобы гарантировать, что JSON используется только для вложенных структур, а не для отдельных значений. Взято из примера `applicationNumber.json` ниже:

```
{
  "$id" : "applicationNumber.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "type" : "object",
  "additionalProperties" : false,
  "properties" : {
    "applicationNumber" : {
      "$ref" : "#/$defs/applicationNumber"
    }
  },
  "required" : [ "applicationNumber" ],
  "$defs" : {
    "applicationNumber" : {
      "$ref" : "applicationNumberType.json#/$defs/applicationNumberType",
      "description" : "Description: Numbers used by IPOs in order to identify each application received; Version: V5_0"
    }
  }
}
```

- [JSD-14] Самый внешний объект схемы ДОЛЖЕН иметь ключевое слово `"type"`, со значением `"object"`.
- [JSD-15] Самый внешний объект схемы ДОЛЖЕН иметь ключевое слово `"$defs"`, значение которого является свойством самого внешнего объекта схемы.
- [JSD-16] Самый внешний объект схемы ДОЛЖЕН иметь ключевое слово `"required"`, значение которого представляет собой массив, содержащий один элемент, т.е. свойство самого внешнего объекта схемы.

Можно использовать расширения схемы JSON (customizations). Если расширенный тип является типом объекта, то на него необходимо ссылаться, а не дублировать его свойства, чтобы повысить возможность повторного использования.

- [JSD-17] Расширения схемы JSON (customizations) для типов объектов ДОЛЖНЫ быть реализованы в виде ссылки на схему JSON расширенного типа.

## 7 ПРАВИЛА ПРОЕКТИРОВАНИЯ КОНСТРУКЦИЙ JSON SCHEMA

### 7.1 Обзор

В этом разделе устанавливаются правила для конструкций схемы JSON, в частности для массивов, объектов и базовых значений. Стандартизация имен для конструкций схемы имеет решающее значение для разработки надежной архитектуры данных.

### 7.2 Свойства

Свойства, также известные как члены, являются основными строительными блоками конструкции JSON.

[JSC-01] В определениях СЛЕДУЕТ использовать существующие схемы в максимально возможной степени.

[JSC-02] Несколько свойств, которые могут быть логически сгруппированы вместе, МОГУТ быть объявлены в одном файле схемы под глобальным определением.

### 7.3 Определения

Каждое свойство должно иметь глобальное определение, определенное в его схеме JSON. Это позволит повторно использовать имя свойства во многих родительских элементах и иметь согласованное определение для всех из них. См. свойство "applicationNumber" в приведенном ниже примере.

[JSC-03] Каждому свойству, указанному в ключевом слове `properties`, СЛЕДУЕТ ссылаться на глобальное определение, описанное в ключевом слове `"$defs"`. Этому глобальному определению СЛЕДУЕТ иметь то же имя, что и свойство.

#### Пример свойства, ссылающегося на глобальное определение

```
{
  "$id" : "applicationNumber.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "type" : "object",
  "additionalProperties" : false,
  "properties" : {
    "applicationNumber" : {
      "$ref" : "#/$defs/applicationNumber"
    }
  },
  "required" : [ "applicationNumber" ],
  "$defs" : {
    "applicationNumber" : {
      "$ref" : "applicationNumberType.json#/$defs/applicationNumberType",
      "description" : "Description: Numbers used by IPOs in order to identify each
application received; Version: V5_0"
    }
  }
}
```

Глобальное определение свойства следует составить из имени файла и описания свойства.

[JSC-04] Глобальному определению свойства СЛЕДУЕТ состоять из имени файла и описания (`"description"`) свойства.

Свойства должны иметь типы. Их можно определить напрямую, если они являются примитивными типами (кроме объектов), или обрабатывать их через ссылку на определение глобального свойства в другой схеме JSON.

[JSC-05] Свойство ДОЛЖНО иметь тип, указанный с помощью ключевого слова `"type"`, либо как непосредственно свойство, либо через ссылку на глобальное определение.

### 7.4 Определения типов

Схемы JSON могут определять повторно используемые определения типов, на которые ссылаются определения глобальных свойств. Эти глобальные определения типов должны состоять из ключевого слова `"type"`, ключевого слова `"properties"` (если тип - `"object"`) и любых других ограничений значений.

[JSC-06] Схема МОЖЕТ содержать определения глобальных типов для повторного использования моделей содержимого во многих свойствах.

**Повторно используемое определение типа**

```
{
  "$id" : "applicationNumber.json",
  "$schema" : "https://json-schema.org/draft/2020-12/schema",
  "type" : "object",
  "additionalProperties" : false,
  "properties" : {
    "applicationNumber" : {
      "$ref" : "#/$defs/applicationNumber"
    }
  },
  "required" : [ "applicationNumber" ],
  "$defs" : {
    "applicationNumber" : {
      "$ref" : "applicationNumberType.json#/$defs/applicationNumberType",
      "description" : "Description: Numbers used by IPOs in order to identify each
application received; Version: 5_0"
    }
  }
}
```

[JSC-07] Определения, которые представляют типы, ДОЛЖНЫ иметь имена, соответствующие соглашению LCC + суффикс "Type".

#### 7.5 Базовые (primitive) типы JSON

[JSC-08] Для свойства СЛЕДУЕТ использовать наиболее подходящий базовый тип JSON. К базовым типам относятся: "string" (строка), "number" (число), "integer" (целое), "object" (объект), "array" (массив), "boolean" (логический тип) и "null" (нулевой).

Например, если значением свойства будет целое число, следует использовать тип "integer", а не более общий "number" или еще более широкий "string". Для типа "string", если применимо, следует использовать встроенные форматы, например, "date-time" (дата-время) или "duration" (длительность).

#### 7.6 Списки кодов

В некоторых случаях целесообразно ограничить значение пронумерованным списком кодов, которые являются стандартными и приемлемыми для целей обмена данными. Списки кодов — это средство для создания контролируемого словаря разрешенных значений для элемента данных (например, стандартный список кодов для кодов стран, кодов языков, кодов патентных ведомств и т.д.). Следует использовать перечни кодов, которые уже существуют в открытом доступе и поддерживаются соответствующими комитетами по стандартам, такими как ISO.

[JSC-09] Стандарт ВОИС ST.3 ДОЛЖЕН использоваться для представления МПВ (Международных патентных ведомств), государств, других субъектов, организаций, а также для приоритетных и определенных стран/организаций.

[JSC-10] Элементы кода ISO 3166-1-Alpha-2 (двухбуквенные коды стран) ДОЛЖНЫ использоваться для представления названий стран для адресации и гражданства.

[JSC-11] ISO 639-1 (двухбуквенные коды языков) ДОЛЖЕН использоваться для кодов языков.

[JSC-12] ISO 4217-Alpha (трехбуквенные коды валют) ДОЛЖЕН использоваться для кодов валют.

[JSC-13] Ключевое слово JSON enum (перечисление) СЛЕДУЕТ использовать для задания определения списков кодов.

[JSC-14] Символы, используемые в значениях перечисления, ДОЛЖНЫ быть ограничены следующим набором: {a-z, A-Z, 0-9, точка (.), запятая (,), пробелы, дефис (-) и знак подчеркивания (\_)}

#### 7.7 Массивы

Размер задается как количество элементов в массиве. Размер указывается в схеме с помощью ключевых слов minItems и maxItems (минимальное и максимальное количество элементов). Разработчикам схемы рекомендуется не указывать значения по умолчанию для количества появлений элементов (т.е. "minItems": 0), поскольку это может излишне загромождать схему.

[JSC-15] В схемах JSON СЛЕДУЕТ использовать ключевые слова `minItems` и `maxItems` для массивов, за исключением значения по умолчанию `minItems` (0).

Тип элементов в массиве должен быть определен с помощью ключевого слова `"items"`. Для простоты все элементы в массиве должны иметь один и тот же тип. Если требуется последовательность объектов разных типов, их следует определить как отдельные свойства объекта.

[JSC-16] Для каждого объекта типа "массив" ДОЛЖНО иметься ключевое слово `"items"`, и его значение ДОЛЖНО быть одиночным объектом схемы, а не массивом. Все элементы массива ДОЛЖНЫ иметь один и тот же тип.

Ключевое слово `"additionalItems"` нельзя использовать для массивов, так как оно не имеет значения, когда значением `"items"` является одиночный объект схемы.

[JSC-17] Ключевое слово `"additionalItems"` НЕ СЛЕДУЕТ использовать, когда `"items"` является одиночным объектом схемы.

## 7.8 Объекты

### 7.8.1 Свойство `"wildcards"` (подстановочные знаки)

Схемы JSON не должны допускать, чтобы произвольные свойства были частью экземпляра JSON, причем корректными, поскольку это может нарушить целостность обмена данными.

Требуется использовать ключевое слово `"AdditionalProperties"`, значение которого должно быть установлено в `"false"`. В противном случае в экземплярах будут разрешены неопределенные свойства.

[JSC-18] Схема JSON ДОЛЖНА иметь свойство `"AdditionalProperties"`, значение которого устанавливаться в `"false"` для всех объектов.

Использование ключевого слова `"patternProperties"` не допускается. Это ключевое слово позволяет отображать регулярные выражения в схемы. Например, оно допускает неявные определения схемы на основе имени свойства. Для определенного вида имени свойства применяется определенная схема.

[JSC-19] В схеме НЕ ДОЛЖНО использоваться ключевое слово `"patternProperties"`.

### 7.8.2 Порядок расположения свойств

Схема JSON не требует определенного порядка свойств объекта. Однако если для схемы JSON имеется соответствующая схема XML, рекомендуется, чтобы свойства были перечислены в том же порядке, что и в схеме XML, как в схеме JSON, так и в экземпляре.

[JSC-20] Схеме СЛЕДУЕТ использовать тот же порядок свойств, что и в соответствующей XML-схеме, если она существует.

## 8 ИДЕНТИФИКАТОРЫ СХЕМЫ JSON

### 8.1 Обзор

Идентификатор в схеме JSON предоставляет URI, который идентифицирует категорию информации на основе бизнес-домена (например корпоративного, патентного и товарных знаков). В данном проекте стандарта было принято решение использовать отношения «многие к одному» между несколькими идентификаторами и, возможно, сотнями конструкций JSON. Планируется, что группа связанных конструкций JSON с уникальными именами будет связана с определенным идентификатором (ID). Для идентификации следует использовать универсальный идентификатор ресурса (URI).

[JID-01] Идентификаторы ДОЛЖНЫ применяться в схемах с использованием ключевого слова `"$id"`.

## 9. ПРАВИЛА ПРОЕКТИРОВАНИЯ ЭКЗЕМПЛЯРА JSON (JSON INSTANCE)

Схема JSON определяет структуру и ограничения для экземпляра JSON. Чтобы улучшить качество и обеспечить надежность обмена данными (внутри и между офисами), экземпляры JSON должны соотноситься со схемой JSON в целях проверки корректности и соответствия.

### 9.1 Порядок свойств

Схема JSON не требует определенный порядок свойств объекта. Однако рекомендуется, чтобы свойства отображались в экземпляре в том же порядке, что и в схеме JSON.



[JIN-01] В экземпляре документа JSON СЛЕДУЕТ использовать тот же порядок свойств, что и в соответствующей схеме JSON, если она существует.

## 9.2 Проверка экземпляра JSON

Успешная проверка экземпляров JSON на корректность гарантирует, что их содержимое удовлетворяет всем требованиям, определенным в соответствующих схемах.

[JIN-02] Экземпляры документов JSON МОГУТ проверяться на соответствие схеме в процессе обработки.

[JIN-03] МОЖЕТ быть создана схема времени выполнения для удовлетворения требований к производительности приложения в среде выполнения. Например, составная схема для номера заявки может использоваться в качестве схемы времени выполнения.

[JIN-04] Все модификации, обновления, исправления и новые выпуски ДОЛЖНЫ быть сначала отправлены на одобрение Целевой группе XML4IP, прежде чем изменения могут быть включены в схему времени выполнения.

Ожидается, что экземпляр JSON соответствует схеме.

[JIN-05] Экземпляру JSON СЛЕДУЕТ соответствовать определенной схеме JSON, которая соответствует правилам, описанным в этом стандарте.

## 10. ССЫЛКИ

### Стандарты ВОИС

- Стандарт ВОИС [ST.96](#): Рекомендации по обработке информации по интеллектуальной собственности с использованием XML
- Стандарт ВОИС [ST.90](#): Рекомендации по обработке и обмену данными об интеллектуальной собственности с использованием веб-интерфейсов API (интерфейсов прикладного программирования)

### Стандарты и отраслевые спецификации

- Спецификация JSON: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>
- Ядро схемы JSON, проект 2020–2012 гг.: <http://json-schema.org/latest/json-schema-core.html>
- Проверка схемы JSON, проект 2020–2012 гг.: <http://json-schema.org/latest/json-schema-validation.html>
- Спецификация OpenAPI v3.1.0: <https://spec.openapis.org/oas/v3.1.0#schema>
- Стандарт ISO 21778 ECMA-404 – Синтаксис обмена данными JSON: <https://www.ecma-international.org/publications/standards/Ecma-404.htm>
- UBL-2.1 Универсальный деловой язык: <http://docs.oasis-open.org/ubl/UBL-2.1.html>
- Коды стран ISO 3166: <https://www.iso.org/iso-3166-country-codes.html>
- Коды языков ISO 639: <https://www.iso.org/iso-639-language-codes.html>
- Текущие коды ISO 4217: <https://www.iso.org/iso-4217-currency-codes.html>
- ISO 11179: <https://www.iso.org/standard/60341.html>
- RFC 2119: <https://www.ietf.org/rfc/rfc2119.txt>

[Приложение I следует]