

Internet Engineering Task Force (IETF)  
Request for Comments: 8791  
Updates: 8340  
Category: Standards Track  
ISSN: 2070-1721

A. Bierman  
YumaWorks  
M. Bjorklund  
Cisco  
K. Watsen  
Watsen Networks  
June 2020

## YANG Data Structure Extensions

Расширения структур данных YANG

### Аннотация

Этот документ описывает механизмы YANG для определения абстрактных структур данных с помощью YANG.

### Статус документа

Документ относится к категории Internet Standards Track.

Документ является результатом работы IETF<sup>1</sup> и представляет согласованный взгляд сообщества IETF. Документ прошёл открытое обсуждение и был одобрен для публикации IESG<sup>2</sup>. Дополнительную информацию о стандартах Internet можно найти в разделе 2 в RFC 7841.

Информация о текущем статусе документа, найденных ошибках и способах обратной связи доступна по ссылке <https://www.rfc-editor.org/info/rfc8791>.

### Авторские права

Copyright (c) 2020. Авторские права принадлежат IETF Trust и лицам, указанным в качестве авторов документа. Все права защищены.

К документу применимы права и ограничения, указанные в BCP 78 и IETF Trust Legal Provisions и относящиеся к документам IETF (<http://trustee.ietf.org/license-info>), на момент публикации данного документа. Прочтите упомянутые документы внимательно. Фрагменты программного кода, включённые в этот документ, распространяются в соответствии с упрощённой лицензией BSD, как указано в параграфе 4.e документа IETF Trust Legal Provisions, без каких-либо гарантий (как указано в Simplified BSD License).

## Оглавление

1. Введение.....	1
1.1. Терминология.....	2
1.1.1. NMDA.....	2
1.1.2. YANG.....	2
2. Определения.....	2
3. Структуры данных YANG в диаграммах деревьев YANG.....	2
4. Модуль расширения структур данных YANG.....	3
5. Взаимодействие с IANA.....	5
5.1. Реестр YANG Module.....	5
6. Вопросы безопасности.....	5
7. Литература.....	5
7.1. Нормативные документы.....	5
7.2. Дополнительная литература.....	6
Приложение А. Примеры.....	6
A.1. structure.....	6
A.2. augment-structure.....	6
A.3. Кодирование XML.....	7
A.4. Кодирование JSON.....	7
A.5. Пример structure с определением структуры не верхнего уровня.....	7
Адреса авторов.....	8

## 1. Введение

Имеется потребность в стандартных механизмах, позволяющих определять абстрактные данные, не предназначенные для реализации в качестве конфигурации или рабочего состояния. Для этого было задано расширение yang-data в RFC 8040 [RFC8040], но его функциональность ограничена.

Расширение yang-data предполагалось применять для моделирования протокольных сообщений или их частей, как в определении errors модуля YANG ietf-restconf [RFC8040] или в содержимом файла. Однако протоколы часто разделены на уровни так, что заголовок и части данных могут расширяться в дополнительных (внешних) документах. Операторы YANG, моделирующие протокол, должны поддерживать расширяемость, которая уже имеется в протоколе.

<sup>1</sup>Internet Engineering Task Force - комиссия по решению инженерных задач Internet.

<sup>2</sup>Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

Этот документ определяет новый оператор расширения YANG structure, похожий на yang-data из [RFC8040], но более гибкий. Не принимается никаких допущения о том, что структура данных YANG может использоваться лишь как абстракция верхнего уровня и она может быть вложена в другую структуру данных.

Документ также определяет оператор расширения YANG augment-structure, позволяющий дополнять абстрактные структуры из внешних модулей и походить на имеющийся в YANG оператор augment. Отметим, что augment нельзя применять для дополнения структур данных YANG, поскольку от компиляторов YANG и иных инструментов не требуется понимать расширение structure.

## 1.1. Терминология

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не нужно** (SHALL NOT), **следует** (SHOULD), **не следует** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **не рекомендуется** (NOT RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе должны интерпретироваться в соответствии с BCP 14 [RFC2119] [RFC8174] тогда и только тогда, когда они выделены шрифтом, как показано здесь.

### **YANG data structure - структура данных YANG**

Структура данных, определённая с применением оператора structure.

#### 1.1.1. NMDA

Термины, определённые в архитектуре хранилища данных сетевого управления (Network Management Datastore Architecture или NMDA) [RFC8342]:

- configuration - конфигурация;
- operational state - рабочее состояние.

#### 1.1.2. YANG

Термины, определённые в [RFC7950]:

- absolute-schema-nodeid;
- container - контейнер;
- data definition statement - оператор определения данных;
- data node - узел данных;
- leaf - лист;
- leaf-list - лист-список;
- list - список.

## 2. Определения

Структуры данных YANG определяются оператором расширения structure, определённым в модуле YANG ietf-yang-structure-ext. Аргументом structure является имя структуры данных. Принадлежность структур данных к одному идентификатору пространства имён определена в параграфе 6.2.1 [RFC7950]. В частности, там сказано (седьмой пункт):

Все узлы типа leaf, leaf-list, list, container, choice, rpc, action, notification, anydata и anyxml, определённые (напрямую или с помощью оператора uses) в родительском узле или на верхнем уровне модуля и его submodule, используют общее пространство имён.

Это значит, что структуры данных, определённые оператором structure, не могут иметь имя, совпадающее с именем братского (sibling) узла из обычного оператора определения данных YANG или других операторов structure в том же модуле YANG.

Это не означает, что однажды определённая структура данных YANG должна применяться как протокольное сообщение или иная структура данных верхнего уровня.

Структура данных YANG кодируется как узел anydata. Это означает, что имя структуры кодируется как контейнер (container), а дочерние экземпляры как потомки этого узла. Например, структуру

```
module example-errors {
  ...

  sx:structure my-error {
    leaf error-number {
      type int;
    }
  }
}
```

можно представить в JSON как

```
"example-errors:my-error": {
  "error-number": 131
}
```

## 3. Структуры данных YANG в диаграммах деревьев YANG

Структура данных YANG может быть выведена в диаграмме дерева YANG [RFC8340]. Этот документ обновляет RFC 8340 [RFC8340], определяя два новых раздела в диаграмме дерева для модуля.

1. Структуры данных YANG смещены на 2 пробела и указываются ключевым словом `structure`, за которым следует имя структуры данных YANG и двоеточие (:).
2. Дополнения структуры данных YANG, смещены на 2 пробела и указываются ключевым словом `augment-structure`, за которым следует имя целевой структуры расширения и двоеточие (:).

Новые разделы с учётом пробелов имеют вид

```

structure <structure-name>:
  +--<node>
    +--<node>
      | +--<node>
      +--<node>
structure <structure-name>:
  +--<node>

augment-structure <structure-name>:
  +--<node>
    +--<node>
      | +--<node>
      +--<node>
augment-structure <structure-name>:
  +--<node>

```

Узлы в структурах данных YANG выводятся по правилам параграфа 2.6 в [RFC8340] и не имеют флагов (<flags>).

## 4. Модуль расширения структур данных YANG

```

<CODE BEGINS> file "ietf-yang-structure-ext@2020-06-17.yang"
module ietf-yang-structure-ext {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-yang-structure-ext";
  prefix sx;

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";
  contact
    "WG Web: <https://datatracker.ietf.org/wg/netmod/>
    WG List: <mailto:netmod@ietf.org>

    Author: Andy Bierman
           <mailto:andy@yumaworks.com>

    Author: Martin Bjorklund
           <mailto:mbj+ietf@4668.se>

    Author: Kent Watsen
           <mailto:kent+ietf@watsen.net>";
  description
    "Этот модуль содержит концептуальные спецификации YANG для
    определения абстрактных структур данных.

    Ключевые слова ДОЛЖНО, НЕДОПУСТИМО, ТРЕБУЕТСЯ, НУЖНО, НЕ НУЖНО,
    СЛЕДУЕТ, НЕ СЛЕДУЕТ, РЕКОМЕНДУЕТСЯ, НЕ РЕКОМЕНДУЕТСЯ, МОЖНО,
    НЕОБЯЗАТЕЛЬНО в этом документе трактуются в соответствии с
    ВСП 14 (RFC 2119) (RFC 8174) тогда и только тогда, когда они
    указаны заглавными буквами, как показано здесь.

    Авторские права (c) 2020 IETF принадлежат IETF Trust и
    лицам, указанным как авторы. Все права защищены.
    Распространение и применение модуля в исходной или двоичной
    форме с изменениями или без таковых разрешено в соответствии с
    лицензией Simplified BSD License, изложенной в параграфе 4.c
    IETF Trust's Legal Provisions Relating to IETF Documents
    (https://trustee.ietf.org/license-info).
    Эта версия модуля YANG является частью RFC 8791
    (https://www.rfc-editor.org/info/rfc8791), где правовые
    аспекты приведены более полно.";
  revision 2020-06-17 {
    description
      "Исходный выпуск.";
    reference
      "RFC 8791: YANG Data Structure Extensions.";
  }

  extension structure {
    argument name {
      yin-element true;
    }
    description
      "Это расширение служит для задания структуры данных YANG,
      представляющей концептуальные данные, определённые в YANG.
      Оно предназначено для описания иерархических данных, независимо
      от контекста протокола и формата кодирования сообщений. Операторы
      определения данных в structure задают базовый синтаксис для
      конкретной структуры данных YANG, именем которой служит аргумент
      оператора расширения structure."

```

Отметим, что это расширение не задаёт тип носителя. Спецификация, использующая расширение, ДОЛЖНА задать правила кодирования сообщений, включая тип носителя для содержимого, если это применимо.

Обязательный параметр `name` указывает определяемую структуру данных YANG.

Это расширение действительно лишь как оператор верхнего уровня, т. е. как субоператор в `module` или `submodule`.

Субоператоры этого расширения ДОЛЖНЫ следовать указанным ниже правилам ABNF, заданным в RFC 7950:

```
*must-stmt
[status-stmt]
[description-stmt]
[reference-stmt]
*(typedef-stmt / grouping-stmt)
*data-def-stmt
```

Структура данных YANG, определяемая этим оператором расширения, кодируется как узел `anydata`. Это значит, что имя структуры кодируется как `container` с дочерними операторами, кодируемыми как потомки этого узла.

Имя модуля и пространство имён модуля YANG, использующего оператор расширения, назначаются каждому оператору определения данных, происходящему из структуры данных YANG.

Элемент документа XPath является самым оператором расширения так, что дочерние узлы элемента представляются субоператорами `data-def-stmt` в этом расширении. Этот концептуальный документ является контекстом для указанных ниже операторов YANG.

```
- must-stmt
- when-stmt
- path-stmt
- min-elements-stmt
- max-elements-stmt
- mandatory-stmt
- unique-stmt
- ordered-by
- instance-identifier data type
```

Указанные ниже субоператоры `data-def-stmt` ограничены при использовании в операторе расширения `structure`:

```
- в list-stmt не требуется включать key-stmt;
- config-stmt игнорируется при наличии.
```

```
";
```

```
}
```

```
extension augment-structure {
  argument path {
    yin-element true;
  }
  description
```

```
"Это расширение служит для дополнения структур данных YANG,
определённых оператором structure. Оно предназначено для описания
иерархических данных, независимо от контекста протокола и формата
кодирования сообщений.
```

Оператор имеет почти такую же структуру, как `augment-stmt`. Операторы определения данных внутри этого оператора задают семантику и базовый синтаксис дополнительных данных, добавляемых в конкретную структуру данных YANG, указанную аргументом `path`.

Обязательный параметр `path` указывает концептуальный узел данных YANG, который будет дополнен и представлен строкой `absolute-schema-nodeid`, где первый узел указывает дополняемую структуру данных YANG, а остальные узлы строки указывают узел в структуре YANG для дополнения.

Это расширение действительно лишь как оператор верхнего уровня, т. е. как субоператор в `module` или `submodule`.

Субоператоры этого расширения ДОЛЖНЫ следовать указанным ниже правилам ABNF, заданным в RFC 7950:

```
[status-stmt]
[description-stmt]
[reference-stmt]
1*(data-def-stmt / case-stmt)
```

Имя модуля и пространство имён для модуля YANG, использующего оператор расширения назначаются документу данных экземпляра, соответствующему операторам определений в этом расширении.

Элемент документа XPath является самым оператором расширения так, что дочерние узлы элемента представляются субоператорами `data-def-stmt` в дополняемом операторе `structure`.

Узел контекста оператора `augment-structure` выводится так же, как в [RFC7950]. Этот концептуальный узел считается узлом контекста для перечисленных ниже операторов YANG.

- `must-stmt`
- `when-stmt`
- `path-stmt`
- `min-elements-stmt`
- `max-elements-stmt`
- `mandatory-stmt`
- `unique-stmt`
- `ordered-by`
- `instance-identifier data type`

Указанные ниже субоператоры ограничены при использовании внутри оператора расширения `augment-structure`.

- в `list-stmt` не требуется включать `key-stmt`;
- `config-stmt` игнорируется при наличии.

Пример

```

module foo {
  import ietf-yang-structure-ext { prefix sx; }

  sx:structure foo-data {
    container foo-con { }
  }
}

module bar {
  import ietf-yang-structure-ext { prefix sx; }
  import foo { prefix foo; }

  sx:augment-structure /foo:foo-data/foo:foo-con {
    leaf add-leaf1 { type int32; }
    leaf add-leaf2 { type string; }
  }
}
";
}
}
<CODE ENDS>

```

## 5. Взаимодействие с IANA

### 5.1. Реестр YANG Module

Агентство IANA зарегистрировало указанный ниже идентификатор URI в субреестре ns реестра IETF XML [RFC3688].

URI: urn:ietf:params:xml:ns:yang:ietf-yang-structure-ext

Registrant Contact: The IESG.

XML: N/A; запрошенный идентификатор URI относится к пространству имён XML.

Агентство IANA зарегистрировало указанный ниже модуль YANG в субреестре YANG Module Names [RFC6020] реестра YANG Parameters.

Name: ietf-yang-structure-ext

Namespace: urn:ietf:params:xml:ns:yang:ietf-yang-structure-ext

Prefix: sx

Reference: RFC 8791

## 6. Вопросы безопасности

Этот документ определяет расширения YANG, служащие для определения концептуальных структур данных YANG. Он не создаёт новых уязвимостей в дополнение к имеющимся в YANG 1.1 [RFC7950].

## 7. Литература

### 7.1. Нормативные документы

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", [RFC 7950](#), DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", [RFC 8040](#), DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, [RFC 8340](#), DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", [RFC 8342](#), DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.

[W3C.REC-xml-20081126] Bray, T., Paoli, J., Sperberg-McQueen, M., Maler, E., and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", World Wide Web Consortium Recommendation REC-xml-20081126, November 2008, <<http://www.w3.org/TR/2008/REC-xml-20081126>>.

## 7.2. Дополнительная литература

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, [RFC 3688](#), DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.

## Приложение А. Примеры

### A.1. structure

Этот пример показывает простую адресную книгу, которую можно сохранить.

```

module example-module {
  yang-version 1.1;
  namespace "urn:example:example-module";
  prefix exm;

  import ietf-yang-structure-ext {
    prefix sx;
  }

  sx:structure address-book {
    list address {
      key "last first";
      leaf last {
        type string;
        description "Фамилия";
      }
      leaf first {
        type string;
        description "Имя";
      }
      leaf street {
        type string;
        description "Улица";
      }
      leaf city {
        type string;
        description "Город";
      }
      leaf state {
        type string;
        description "Штат";
      }
    }
  }
}

```

Ниже представлена диаграмма дерева для модуля.

```

module: example-module
  structure address-book:
    +-- address* [last first]
       +-- last      string
       +-- first     string
       +-- street?   string
       +-- city?     string
       +-- state?    string

```

### A.2. augment-structure

Этот пример добавляет листья county и zipcode в книгу адресов.

```

module example-module-aug {
  yang-version 1.1;
  namespace "urn:example:example-module-aug";
  prefix exma;

  import ietf-yang-structure-ext {
    prefix sx;
  }
  import example-module {
    prefix exm;
  }

  sx:augment-structure "/exm:address-book/exm:address" {
    leaf county {

```

```

    type string;
    description "Государство";
  }
  leaf zipcode {
    type string;
    description "Почтовый индекс";
  }
}

```

Ниже представлена диаграмма дерева для модуля.

```

module: example-module-aug

  augment-structure /exm:address-book/exm:address:
    +-- county?   string
    +-- zipcode?  string

```

### A.3. Кодирование XML

Этот пример показывает, как можно закодировать адресную книгу в XML [W3C.REC-xml-20081126].

```

<address-book xmlns="urn:example:example-module">
  <address>
    <last>Flintstone</last>
    <first>Fred</first>
    <street>301 Cobblestone Way</street>
    <city>Bedrock</city>
    <zipcode xmlns="urn:example:example-module-aug">70777</zipcode>
  </address>
  <address>
    <last>Root</last>
    <first>Charlie</first>
    <street>4711 Cobblestone Way</street>
    <city>Bedrock</city>
    <zipcode xmlns="urn:example:example-module-aug">70777</zipcode>
  </address>
</address-book>

```

### A.4. Кодирование JSON

Этот пример показывает, как можно закодировать адресную книгу в JSON.

```

"example-module:address-book": {
  "address": [
    {
      "city": "Bedrock",
      "example-module-aug:zipcode": "70777",
      "first": "Fred",
      "last": "Flintstone",
      "street": "301 Cobblestone Way"
    },
    {
      "city": "Bedrock",
      "example-module-aug:zipcode": "70777",
      "first": "Charlie",
      "last": "Root",
      "street": "4711 Cobblestone Way"
    }
  ]
}

```

### A.5. Пример structure с определением структуры не верхнего уровня

Приведённый ниже пример определяет структуру данных с информацией об ошибках, которую можно включить в элемент <error-info> для <rpc-error>.

```

module example-error-info {
  yang-version 1.1;
  namespace "urn:example:example-error-info";
  prefix exei;

  import ietf-yang-structure-ext {
    prefix sx;
  }

  sx:structure my-example-error-info {
    leaf error-code {
      type uint32;
    }
  }
}

```

Следующий пример показывает использование этой структуры в <rpc-error>.

```

<rpc-reply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <rpc-error>
    <error-type>protocol</error-type>
    <error-tag>operation-failed</error-tag>

```

```
<error-severity>error</error-severity>
<error-info>
  <my-example-error-info
    xmlns="urn:example:example-error-info">
    <error-code>42</error-code>
  </my-example-error-info>
</error-info>
</rpc-error>
</rpc-reply>
```

## Адреса авторов

**Andy Bierman**

YumaWorks

Email: [andy@yumaworks.com](mailto:andy@yumaworks.com)

**Martin Bjorklund**

Cisco

Email: [mbj+ietf@4668.se](mailto:mbj+ietf@4668.se)

**Kent Watsen**

Watsen Networks

Email: [kent+ietf@watsen.net](mailto:kent+ietf@watsen.net)

## Перевод на русский язык

Николай Малых

[nmalykh@protokols.ru](mailto:nmalykh@protokols.ru)