

YANG Patch Media Type

Media-тип YANG Patch

Аннотация

В этом документе описан метод применения исправлений (patch) к конфигурационным базам, использующим язык моделирования данных YANG.

Статус документа

Документ относится к категории Internet Standards Track.

Документ является результатом работы IETF¹ и представляет согласованный взгляд сообщества IETF. Документ прошёл открытое обсуждение и был одобрен для публикации IESG². Дополнительную информацию о стандартах Internet можно найти в разделе 2 в RFC 7841.

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке <http://www.rfc-editor.org/info/rfc8072>.

Авторские права

Авторские права (Copyright (c) 2017) 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.....	2
1.1. Терминология.....	2
1.1.1. NETCONF.....	2
1.1.2. HTTP.....	2
1.1.3. YANG.....	2
1.1.4. RESTCONF.....	2
1.1.5. YANG Patch.....	3
1.1.6. Примеры.....	3
1.1.7. Обозначения в диаграмме дерева.....	3
2. YANG Patch.....	3
2.1. Целевой ресурс.....	4
2.2. Запрос yang-patch.....	4
2.3. Отклик yang-patch-status.....	4
2.4. Целевой узел данных.....	5
2.5. Операции редактирования.....	5
2.6. Обработка отклика об успешном редактировании.....	5
2.7. Обработка ошибок.....	5
2.8. Свойство RESTCONF ":yang-patch".....	5
3. Модуль YANG.....	5
4. Взаимодействие с IANA.....	10
4.1. Регистрация нового URI и модуля YANG.....	10
4.2. Типы носителя.....	10
4.2.1. application/yang-patch+xml.....	10
4.2.2. application/yang-patch+json.....	10
4.3. RESTCONF Capability URN.....	11
5. Вопросы безопасности.....	11
6. Литература.....	11
6.1. Нормативные документы.....	11
6.2. Дополнительная литература.....	12
Приложение А. Пример модуля YANG.....	12
А.1. Примеры YANG Patch.....	13
А.1.1. Ошибка при добавлении ресурса.....	13

¹Internet Engineering Task Force - комиссия по решению инженерных задач Internet.

²Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

А.1.2. Успешное добавление ресурсов.....	14
А.1.3. Вставка списка.....	15
А.1.4. Перемещение списка.....	15
А.1.5. Редактирование ресурса хранилища данных.....	16
Благодарности.....	17
Адреса авторов.....	17

1. Введение

Нужны стандартные механизмы для наложения правок (patch) в хранилища данных [RFC6241], которые содержат концептуальные данные, соответствующие схеме YANG [RFC7950]. Требуется подход на основе упорядоченного списка правок (ordered 'edit' list), чтобы предоставить разработчикам клиентов RESTCONF более чёткий контроль клиентов RESTCONF за процедурой редактирования, нежели даёт механизм простого исправления (plain patch), заданный в [RFC8040].

В этом документе определён media-тип для механизма редактирования на основе YANG, который можно применять в методе HTTP PATCH [RFC5789]. Тип YANG Patch предназначен для поддержки протокола RESTCONF, заданного в [RFC8040]. Документ задаёт лишь применение типа YANG Patch с протоколом RESTCONF.

Возможно использование YANG Patch и с другими протоколами, но этот вопрос выходит за рамки документа. Для любого протокола, поддерживающего тип YANG Patch, недопустимо применение правок, если документ с исправлениями невозможно применить целиком. Можно применять YANG Patch с хранилищами данных, отличающимися от хранилищ конфигурации, но этот вопрос выходит за рамки документа.

1.1. Терминология

Ключевые слова **должно** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не следует** (SHALL NOT), **следует** (SHOULD), **не нужно** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **не рекомендуется** (NOT RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с [RFC2119].

1.1.1. NETCONF

Ниже приведены термины, определённые в [RFC6241].

- configuration data - данные конфигурации;
- datastore - хранилище данных;
- configuration datastore - хранилище конфигурации;
- protocol operation - операция протокола;
- running configuration datastore - хранилище рабочей конфигурации;
- state data - данные состояния;
- user - пользователь.

1.1.2. HTTP

Ниже приведены термины, определённые в [RFC7230].

- header field - поле заголовка;
- message-body - тело сообщения;
- query - запрос;
- request URI - URI запроса.

Ниже приведены термины, определённые в [RFC7231].

- method - метод;
- request - запрос;
- resource - ресурс.

1.1.3. YANG

Ниже приведены термины, определённые в [RFC7950].

- container - контейнер;
- data node - узел данных;
- leaf - лист;
- leaf-list - лист-список;
- list - список.

1.1.4. RESTCONF

Ниже приведены термины, определённые в [RFC8040].

- application/yang-data+xml;
- application/yang-data+json;

- data resource - ресурс данных;
- datastore resource - ресурс хранилища данных;
- patch - исправление;
- RESTCONF capability - свойство RESTCONF;
- target resource - целевой ресурс;
- YANG data template - шаблон данных YANG.

1.1.5. YANG Patch

Ниже определены используемые в этом документе термины.

RESTCONF client - клиент RESTCONF

Клиент, реализующий протокол RESTCONF.

RESTCONF server - сервер RESTCONF

Сервер, реализующий протокол RESTCONF.

YANG Patch - исправление YANG

Концептуальный запрос редактирования с использованием шаблона yang-patch, определённого в разделе 3. В HTTP указывает метод PATCH, где представление использует тип носителя application/yang-patch+xml или application/yang-patch+json.

YANG Patch Status - статус исправления YANG

Концептуальный запрос статуса редактирования с использованием шаблона yang-patch-status, определённого в разделе 3. В HTTP указывает метод PATCH, где представление использует тип носителя application/yang-patch+xml или application/yang-patch+json.

YANG Patch template - шаблон исправления YANG

Похож на шаблон данных YANG, но представляется типом носителя application/yang-patch+xml или application/yang-patch+json.

1.1.6. Примеры

Некоторые строки протокольных сообщений в примерах этого документа разбиты на несколько строк для удобочитаемости. Строка, заканчивающаяся символом \ фактически продолжается в следующей строке текста и считается, что строки будут объединены с удалением символов \ в конце.

1.1.7. Обозначения в диаграмме дерева

В этом документе применяется упрощённое графическое представление модели данных с использованием указанных ниже символов.

- Квадратные скобки [] служат для указания ключей списков.
- В сокращениях перед узлами данных gw означает данные конфигурации (чтение и запись), ro - данные состояния (только чтение), x - операции (исполнение).
- После имени узла данных символ ? означает необязательный узел, * - список (list) и лист-список (leaf-list).
- В круглых скобках указываются варианты выбора и узлы case (последние указываются двоеточием :).
- Многоточие «...» указывает содержимое поддерева, которое не показано.

2. YANG Patch

YANG Patch представляет собой упорядоченный список правок, которые нужно применить к целевому хранилищу данных на сервере RESTCONF. Конкретные поля определены в модуле YANG, приведённом в разделе 3.

Операция YANG Patch вызывается клиентом RESTCONF путём отправки запроса методом PATCH с представлением, использующим тип носителя (media-type) application/yang-patch+xml или application/yang-patch+json. Это тело сообщения, представляющее входные параметры YANG Patch, **должно** присутствовать.

YANG Patch имеет некоторые свойства, которые невозможно передать с помощью механизма plain-patch, определённого в RESTCONF [RFC8040]:

- YANG Patch позволяет редактировать несколько субресурсов внутри одного метода PATCH;
- YANG Patch позволяет более точно по сравнению с plain patch [RFC8040] задать операцию редактирования (поддерживается 7 операций: create, delete, insert, merge, move, replace, remove);
- YANG Patch использует список edit с явным порядком выполнения, редактирование выполняется в заданном клиентом порядке и обработка ошибок может быть точной даже при возникновении нескольких ошибок в одном запросе.

YANG Patch patch-id может служить для отладки и этот идентификатор **следует** включать в журнальные записи аудита, создаваемые сервером RESTCONF для исправления.

Сервер RESTCONF **должен** возвращать поле заголовка Accept-Patch в отклике OPTIONS, как указано в [RFC5789], с типом носителя для YANG Patch. Это нужно клиенту для определения формата кодирования сообщений, поддерживаемого сервером (например, XML, JSON или оба). Заголовок Accept-Patch имеет вид

Accept-Patch: application/yang-patch+xml,application/yang-patch+json

Отметим, что YANG Patch может редактировать лишь ресурсы данных и метод PATCH не может служить для замены ресурса хранилища данных. Хотя модуль YANG ietf-yang-patch написан с использованием YANG 1.1 [RFC7950], реализации YANG Patch можно применять для содержимого, заданного с использованием YANG версии 1 [RFC6020].

YANG Patch можно представить в формате XML в соответствии с [W3C.REC-xml-20081126], а также в формате JSON в соответствии с «JSON Encoding of Data Modeled with YANG» [RFC7951]. Если нужно передать метаданные в сообщении JSON, они кодируются в соответствии с документом «Defining and Using Metadata with YANG» [RFC7952].

2.1. Целевой ресурс

Операция YANG использует Patch URI целевого ресурса RESTCONF для указания изменяемого ресурса. Это может быть ресурс хранилища данных, т. е. "{+restconf}/data", для редактирования данных конфигурации верхнего уровня или ресурс данных конфигурации, например, "{+restconf}/data/ietf-interfaces:interfaces", для редактирования субресурсов внутри ресурса данных конфигурации верхнего уровня.

Цель **должна** указывать в точности 1 ресурс. Если идентифицировано несколько ресурсов, выполнять запрос **недопустимо** и сервер **должен** возвращать сообщение "400 Bad Request" (недопустимый запрос). Если целевой ресурс не соответствует ни одному имеющемуся экземпляру, выполнять запрос **недопустимо** и сервер **должен** передать сообщение об ошибке "404 Not Found" (не найдено).

Каждое редактирование (edit) в YANG Patch указывает целевой узел данных для редактирования (см. параграф 2.4).

2.2. Запрос yang-patch

YANG Patch указывается уникальным идентификатором patch-id и может иметь комментарий. Исправление (patch) представляет собой упорядоченный набор правок (edit), каждая из которых указывается идентификатором edit-id и включает операцию редактирования (create, delete, insert, merge, move, replace, remove), применяемую к целевому ресурсу. Каждая операция редактирования может применяться к субресурсу target внутри целевого ресурса. Для операций insert и move параметр where указывает узел, который вставляется или перемещается. Для значений before и after параметр point указывает место вставки данных.

Операции merge, replace, create, delete, remove имеют такие же значения, которые определены для атрибута operation в параграфе 7.2 [RFC6241].

Каждое редактирование (edit) внутри YANG Patch **должно** указывать в точности один экземпляр ресурса данных. При указании нескольких ресурсов выполнять запрос **недопустимо** и сервер **должен** возвращать сообщение "400 Bad Request" (недопустимый запрос). Если редактирование указано для отсутствующего ресурса и задана операция delete или move¹, выполнять запрос **недопустимо** и сервер **должен** передать сообщение об ошибке "404 Not Found" (не найдено). Сервер **должен** передавать отклик yang-patch-status, указывающий недействительные операции редактирования.

YANG Patch не предоставляет доступа к каким-либо хранилищам. Способ редактирования на сервере, совмещённом с сервером NETCONF, который предоставляет доступ к отдельным хранилищам данных, остаётся за реализацией. Хранилище не может быть заменено полностью, как это предусмотрено для операции <copy-config>, заданной в параграфе 7.3 [RFC6241]. YANG Patch воздействует лишь на указанные узлы.

Тело сообщения, представляющее YANG Patch, передаётся клиентом RESTCONF для указания запроса операции редактирования. При использовании с методом HTTP PATCH эти данные указываются типом носителя YANG Patch.

Ниже приведена диаграмма дерева YANG для контейнера yang-patch.

```
+---- yang-patch
+---- patch-id      string
+---- comment?     string
+---- edit* [edit-id]
+---- edit-id      string
+---- operation     enumeration
+---- target        target-resource-offset
+---- point?       target-resource-offset
+---- where?       enumeration
+---- value?
```

2.3. Отклик yang-patch-status

Тело сообщения с YANG Patch Status возвращается клиенту RESTCONF для детального информирования о результатах операции редактирования. При использовании с методом HTTP PATCH эти данные указываются типом носителя YANG Patch Status, синтаксис которого описан в разделе 3.

Ниже приведена диаграмма дерева YANG для контейнера yang-patch-status.

```
+---- yang-patch-status
+---- patch-id      string
+---- (global-status)?
| +---: (global-errors)
| | +---- errors
| | | +---- error*
| | | +---- error-type      enumeration
| | | +---- error-tag       string
| | | +---- error-app-tag?  string
| | | +---- error-path?    instance-identifier
| | | +---- error-message?  string
| | | +---- error-info?
| +---: (ok)
| +---- ok?              empty
+---- edit-status
+---- edit* [edit-id]
+---- edit-id          string
+---- (edit-status-choice)?
+---: (ok)
```

¹В оригинале ошибочно сказано «отличается от create», см. <https://www.rfc-editor.org/errata/eid5131>. Прим. перев.

```

| +---- ok?          empty
+--:(errors)
  +---- errors
    +---- error*
      +---- error-type      enumeration
      +---- error-tag       string
      +---- error-app-tag?  string
      +---- error-path?     instance-identifier
      +---- error-message?  string
      +---- error-info?

```

2.4. Целевой узел данных

Целевой узел данных для каждой операции редактирования определяется значением целевого ресурса в запросе и листом `target` в каждой записи `edit`.

Если целевой ресурс в URI запроса указывает ресурс хранилища данных, строка пути в листе `target` считается выражением абсолютного пути, указывающим целевой узел данных для соответствующего редактирования. Первый узел листа `target` является узлом данных верхнего уровня в модуле YANG. Листу `target` **недопустимо** содержать лишь символ `/`, поскольку он будет указывать ресурс хранилища, а не ресурс данных.

Если целевой ресурс в URI запроса указывает ресурс данных конфигурации, строка пути в листе `target` считается выражением относительного пути. Первым узлом, указанным в листе `target`, является дочерний узел данных конфигурации узла, связанного с целевым ресурсом. Если лист `target` содержит лишь символ `/`, целевым узлом данных является целевой узел данных ресурса.

2.5. Операции редактирования

В YANG Patch каждое редактирование (`edit`) указывает одну операцию редактирования целевого узла данных. Набор операций содержит операции редактирования NETCONF, а также несколько новых операций.

		Операции редактирования YANG Patch.
Операция	Описание	
<code>create</code>	Создает новый ресурс данных, если такого ещё нет. При наличии ресурса возвращается ошибка.	
<code>delete</code>	Удаляет имеющийся ресурс данных, а при отсутствии такового возвращает ошибку.	
<code>insert</code>	Вставляет новый упорядоченный пользователем ресурс данных.	
<code>merge</code>	Сливает значение <code>edit</code> с целевым ресурсом данных, создавая тот при отсутствии.	
<code>move</code>	Перемещает (меняет порядок) целевой ресурс данных.	
<code>replace</code>	Заменяет целевой ресурс данных значением <code>edit</code> .	
<code>remove</code>	Удаляет ресурс данных, если он имеется.	

2.6. Обработка отклика об успешном редактировании

Если применение YANG завершилось без ошибок, сервер RESTCONF **должен** вернуть клиенту сообщение `yang-patch-status` с установкой для `global-status` значения `ok`.

В Приложении A.1.2 приведён пример отклика об успехе YANG Patch.

2.7. Обработка ошибок

При получении корректно сформированного в соответствии со схемой сообщения YANG Patch сервер RESTCONF будет выполнять предложенное редактирование в порядке указания (по возрастанию). При обработке списка `edit` может возникать несколько ошибок.

Если при выполнении YANG Patch случилась ошибка, серверу RESTCONF **следует** возвращать сообщение `yang-patch-status`. Возможно отклонение некорректного запроса до начала обработки YANG Patch (например, в распределенной реализации). В таких случаях сервер **должен** передавать соответствующее сообщение HTTP об ошибке.

Пример отклика об ошибке YANG Patch приведён в Приложении A.1.1.

2.8. Свойство RESTCONF ":yang-patch"

Предлен идентификатор URI для указания расширения YANG Patch базовому протоколу RESTCONF. Если сервер RESTCONF поддерживает тип носителя YANG Patch, свойство RESTCONF ":yang-patch", определённое в параграфе 4.3, должно быть указано в листе-списке `capability` модуля `ietf-restconf-monitoring`, определённого в [RFC8040].

3. Модуль YANG

Модуль `ietf-yang-patch` задаёт с помощью операторов расширения `yang-data` концептуальные определения, которые не предназначены для реализации в качестве содержимого хранилищ данных на серверах RESTCONF. Определение оператора расширения `yang-data` содержится в модуле `ietf-restconf` [RFC8040], используемом данным модулем.

<CODE BEGINS>

```

file "ietf-yang-patch@2017-02-22.yang"

module ietf-yang-patch {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-yang-patch";
  prefix "ypatch";

  import ietf-restconf { prefix rc; }

  organization
    "IETF NETCONF (Network Configuration) Working Group";
  contact

```

"WG Web: <<https://datatracker.ietf.org/wg/netconf/>>

WG List: <<mailto:netconf@ietf.org>>

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

Author: Martin Bjorklund
<<mailto:mbj@tail-f.com>>

Author: Kent Watsen
<<mailto:kwatsen@juniper.net>>;

description

"Этот модуль содержит концептуальные спецификации YANG для структур данных YANG Patch и YANG Patch Status.

Отметим, что определения YANG в этом модуле не представляют каких-либо данных конфигурации. Оператор YANG grouping обеспечивает нормативный синтаксис для кодирования сообщений с использованием XML и JSON.

Авторские права (Copyright (c) 2017) принадлежат IETF Trust и лицам, указанным как авторы кода. Все права защищены.

Распространение и использование в исходной и двоичной форме с изменениями или без них разрешается в соответствии с условиями, указанными в упрощенной лицензии BSD, изложенной в разделе 4. с Правового положения IETF Trust применительно к документам IETF (<http://trustee.ietf.org/license-info>).

Эта версия модуля YANG является частью RFC 8072, где правовые аспекты выражены более полно."

revision 2017-02-22 {

description

"Исходный выпуск."

reference

"RFC 8072: YANG Patch Media Type."

}

typedef target-resource-offset {

type string;

description

"Содержит строку идентификатора ресурса данных, представляющего субресурс целевого ресурса. Корнем документа для этого выражения служит целевой ресурс, указанный в операции протокола (например, URI для запроса PATCH).

Эта строка кодируется по тем же правилам, которые применяются для идентификаторов ресурсов данных в URI запроса RESTCONF."

reference

"RFC 8040, Section 3.5.3."

}

rc:yang-data "yang-patch" {

uses yang-patch;

}

rc:yang-data "yang-patch-status" {

uses yang-patch-status;

}

grouping yang-patch {

description

"Группировка, содержащая контейнер YANG, который представляет синтаксис и семантику сообщения с запросом YANG Patch edit."

container yang-patch {

description

"Представляет концептуальную последовательность операций edit, вызываемых правкой (patch). Каждая правка получает заданный клиентом идентификатор. Операции редактирования ДОЛЖНЫ применяться в порядке роста идентификатора и все редактирования ДОЛЖНЫ быть применены. При возникновении ошибки НЕДОПУСТИМО изменение целевого хранилища операцией YANG Patch.

Возможно нарушение ограничений хранилища данных, связанное с любым узлом в хранилище, включая узлы, не затрагиваемые списком edit. Все ошибки валидации ДОЛЖНЫ указываться в возвращаемом отклике."

reference

"RFC 7950, Section 8.3."


```

leaf patch-id {
  type string;
  mandatory true;
  description
    "Произвольная строка, представляемая клиентом для указания
    правок (patch) в целом. Сообщения об ошибках, возвращаемые
    сервером при выполнении правок, будут содержать значение
    patch-id. Клиенту СЛЕДУЕТ пытаться создавать уникальные
    значения patch-id, чтобы различать транзакции разных
    клиентов в записях журнала аудита на сервере.";
}

leaf comment {
  type string;
  description
    "Произвольная строка, представляемая клиентом для описания
    правок (patch) в целом. Это значение СЛЕДУЕТ указывать во
    всех записях аудита, создаваемых сервером при правках.";
}

list edit {
  key edit-id;
  ordered-by user;

  description
    "Представляет правку (edit) в запросе YANG Patch. Список
    edit применяется, как указано ниже.
    - Первое редактирование концептуально применяется к
      копии имеющегося целевого хранилища, например,
      хранилища рабочей конфигурации.
    - Каждое следующее редактирование применяется к
      результату предыдущих редактирований (edit).
    - После успешного применения всех edit результат
      проверяется в соответствии с ограничениями YANG.
    - В случае успеха сервер пытается применить результат
      к целевому хранилищу данных.";

  leaf edit-id {
    type string;
    description
      "Произвольная строка, указывающая редактирование (edit).
      Возвращаемые сервером сообщения об ошибках, относящиеся
      к конкретному редактированию, указываются edit-id.";
  }

  leaf operation {
    type enumeration {
      enum create {
        description
          "Целевой узел данных, создаваемый с использованием
          представленного значения, если его ещё нет. Лист
          target указывает создаваемый узел данных, а не
          родительский узел.";
      }
      enum delete {
        description
          "Удаляет целевой узел, если ресурс данных уже
          имеется. В ином случае возвращается ошибка.";
      }
      enum insert {
        description
          "Вставляет указанное значение в упорядоченную
          пользователем запись list или leaf-list. Целевой
          узел представляет новый ресурс данных. Если параметр
          where имеет значение before или after, параметр
          point указывает точку вставки для целевого узла.";
      }
      enum merge {
        description
          "Представленное значение объединяется (merge) с
          целевым узлом данных.";
      }
      enum move {
        description
          "Перемещает целевой узел, изменяя упорядоченный
          пользователем list или leaf-list. Целевой узел должен
          представлять имеющийся ресурс данных. Если параметр
          where имеет значение before или after, параметр
          point указывает точку вставки для целевого узла.";
      }
      enum replace {
        description
          "Представленное значение служит для замены целевого
          узла данных.";
      }
    }
  }
}

```

```

enum remove {
  description
    "Удаляет целевой узел при его наличии.";
}
mandatory true;
description
  "Операция, запрошенная для записи edit.";
}

leaf target {
  type target-resource-offset;
  mandatory true;
  description
    "Указывает целевой узел данных для операции edit. Если
    target имеет значение /, целевым узлом данных будет
    целевой ресурс. Целевым узлом ДОЛЖЕН быть ресурс данных,
    а не ресурс хранилища.";
}

leaf point {
  when "../operation = 'insert' or ../operation = 'move'"
    + "and (../where = 'before' or ../where = 'after')" {
    description
      "Этот лист применяется лишь для операций insert и move,
      применяя её перед или после имеющейся записи.";
  }
  type target-resource-offset;
  description
    "URL абсолютного пути к узлу данных, используемый как
    точка вставки или переноса для этой записи edit.";
}

leaf where {
  when "../operation = 'insert' or ../operation = 'move'" {
    description
      "Применяется лишь для операций insert и move.";
  }
  type enumeration {
    enum before {
      description
        "Указывает вставку или перенос узла данных в точку
        перед ресурсом, указанным параметром point.";
    }
    enum after {
      description
        "Указывает вставку или перенос узла данных в точку
        после ресурса, указанного параметром point.";
    }
    enum first {
      description
        "Указывает вставку или перенос узла данных в точку
        в первую позицию.";
    }
    enum last {
      description
        "Указывает вставку или перенос узла данных в точку
        в последнюю позицию.";
    }
  }
  default last;
  description
    "Указывают место размещения при добавлении или переносе
    ресурса. YANG разрешает эти операции лишь для узлов
    list и leaf-list, упорядоченных пользователем.";
}

anydata value {
  when "../operation = 'create' "
    + "or ../operation = 'merge' "
    + "or ../operation = 'replace' "
    + "or ../operation = 'insert'" {
    description
      "Применяется лишь с операциями create, merge, replace
      и insert.";
  }
  description
    "Значение, применяемое для операции. Параметр value
    содержит целевой ресурс, связанный с листом target.

    Предположим, например, что узлом target служит контейнер
    YANG с именем foo
    container foo {
      leaf a { type string; }
      leaf b { type int32; }
    }

```



```

        Узел value содержит экземпляр foo
        <value>
          <foo xmlns='example-foo-namespace'>
            <a>Значение</a>
            <b>42</b>
          </foo>
        </value>
      ";
    }
  }
} // Группировка yang-patch

grouping yang-patch-status {
  description
    "Группировка, содержащая контейнер YANG с описанием синтаксиса
    и семантики отклика YANG Patch Status.";

  container yang-patch-status {
    description
      "Контейнер, представляющий отклик, передаваемый сервером
      после обработки запроса на редактирование YANG Patch.";

    leaf patch-id {
      type string;
      mandatory true;
      description
        "Значение patch-id, использованное в запросе.";
    }

    choice global-status {
      description
        "Сообщает об ошибке или успехе операции. Если вариант не
        выбран, сообщается об ошибке из контейнера edit-status.";

      case global-errors {
        uses rc:errors;
        description
          "Этот контейнер представляется при глобальной ошибке, не
          связанной с определенным редактированием.";
      }

      leaf ok {
        type empty;
        description
          "Этот лист представляется при успешном выполнении и
          отсутствие ошибок из контейнера edit-status.";
      }
    }

    container edit-status {
      description
        "Этот контейнер представляется при наличии откликов о
        статусе конкретного редактирования. Если все операции
        edit успешны и global-status возвращает ok, сервер
        МОЖЕТ опускать этот контейнер.";

      list edit {
        key edit-id;

        description
          "Представляет список откликов со статусом операций edit
          из запроса YANG Patch. Если запись edit была пропущена,
          или сервер не дошёл до неё, список не будет содержать
          записи для этой операции edit.";

        leaf edit-id {
          type string;
          description
            "Отклик со статусом записи списка edit с данным
            edit-id.";
        }

        choice edit-status-choice {
          description
            "Выбор типа отклика о статусе для каждой записи edit.";
          leaf ok {
            type empty;
            description
              "Запись edit была выполнена без возникновения на
              сервере связанных с ней ошибок.";
          }
          case errors {
            uses rc:errors;
            description
              "Сервер обнаружил ошибку при редактировании для
              этого значения edit-id.";
          }
        }
      }
    }
  }
}

```

```

    }
  }
}
} // Группировка yang-patch-status
}

<CODE ENDS>

```

4. Взаимодействие с IANA

4.1. Регистрация нового URI и модуля YANG

Этот документ регистрирует URI как пространство имён в IETF XML Registry [RFC3688] (в формате RFC 3688).

```

URI: urn:ietf:params:xml:ns:yang:ietf-yang-patch
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.

```

Документ регистрирует модуль YANG в реестре YANG Module Names [RFC6020].

```

name: ietf-yang-patch
namespace: urn:ietf:params:xml:ns:yang:ietf-yang-patch
prefix: ypatch
reference: RFC 8072

```

4.2. Типы носителя

4.2.1. application/yang-patch+xml

Имя типа

application

Имя субтипа

yang-patch+xml

Требуемые параметры

Нет

Необязательные параметры

Нет

Кодировка

8 битов

Для этого типа всегда применяется кодировка utf-8. Каждый концептуальный узел данных YANG представляется в соответствии с XML Encoding Rules and Canonical Format для конкретного типа данных YANG, заданного в [RFC7950]. Дополнительно шаблон yang-patch из RFC 8072 задаёт структуру запроса YANG Patch.

Вопросы безопасности

Вопросы безопасности, связанные с генерацией и применением сообщений RESTCONF, рассмотрены в разделе 5 RFC 8072. Дополнительные соображения безопасности связаны с семантикой конкретных моделей данных YANG. Предполагается, что для каждого модуля YANG будут рассматриваться вопросы безопасности, связанные с определёнными в модуле узлами данных YANG.

Вопросы взаимодействия (совместимости)

В RFC 8072 задан формат сообщений и их интерпретация.

Опубликованная спецификация

RFC 8072

Использующие тип приложения

Синтаксические анализаторы экземпляра данных документа используемые протоколом или средством автоматизации, применяющим структуру данных YANG Patch.

Идентификаторы фрагментов

Синтаксис и семантика идентификаторов фрагментов совпадают с заданными для типа носителя application/xml.

Дополнительные сведения

Отменённые псевдонимы имени для этого типа - не применимо.

Magic number(s) - не применимо.

Расширения имён файлов - нет.

Коды типа файлов Macintosh - TEXT.

Контактные данные

См. раздел «Адреса авторов» в RFC 8072.

Предполагаемое использование

Общего назначения.

Ограничения на использование

Не применимы

Автор

См. раздел «Адреса авторов» в RFC 8072.

Контролёр изменений

Internet Engineering Task Force (iesg@ietf.org).

Временная регистрация? (только для дерева стандартов)

Нет

4.2.2. application/yang-patch+json

Имя типа

application

Имя субтипа

yang-patch+json

Требуемые параметры

Нет

Необязательные параметры

Нет

Кодировка

8 битов

Для этого типа всегда применяется кодировка utf-8. Каждый концептуальный узел данных YANG представляется в соответствии с RFC 7951. Аннотации метаданных кодируются в соответствии с RFC 7952. В дополнение к этому шаблон yang-patch в RFC 8072 задаёт структуру запроса YANG Patch.

Вопросы безопасности

Вопросы безопасности, связанные с генерацией и применением сообщений RESTCONF, рассмотрены в разделе 5 RFC 8072. Дополнительные соображения безопасности связаны с семантикой конкретных моделей данных YANG. Предполагается, что для каждого модуля YANG будут рассматриваться вопросы безопасности, связанные с определёнными в модуле узлами данных YANG.

Вопросы взаимодействия (совместимости)

В RFC 8072 задан формат сообщений и их интерпретация.

Опубликованная спецификация

RFC 8072

Использующие тип приложения

Синтаксические анализаторы экземпляра данных документа используемые протоколом или средством автоматизации, применяющим структуру данных YANG Patch.

Идентификаторы фрагментов

Синтаксис и семантика идентификаторов фрагментов совпадают с заданными для типа носителя application/json.

Дополнительные сведения

Отменённые псевдонимы имени для этого типа - не применимо.

Magic number(s) - не применимо.

Расширения имён файлов - нет.

Коды типа файлов Macintosh - TEXT.

Контактные данные

См. раздел «Адреса авторов» в RFC 8072.

Предполагаемое использование

Общего назначения.

Ограничения на использование

Не применимы

Автор

См. раздел «Адреса авторов» в RFC 8072.

Контролёр изменений

Internet Engineering Task Force (iesg@ietf.org).

Временная регистрация? (только для дерева стандартов)

Нет

4.3. RESTCONF Capability URN

Этот документ задаёт один идентификатор возможности в реестре RESTCONF Capability URNs [RFC8040]. Для этого реестра применяется процедура IETF Review [RFC5226].

Индекс

:yang-patch

Идентификатор возможности

urn:ietf:params:restconf:capability:yang-patch:1.0

5. Вопросы безопасности

Тип носителя YANG Patch не вносит каких-либо значимых угроз безопасности в дополнение к описанным в [RFC8040]. Этот документ задаёт инструкции по обработке редактирования для варианта метода PATCH, используемого в протоколе RESTCONF. Целостность сообщений обеспечивает протокол RESTCONF и дополнительных возможностей проверки неизменности правок (patch) не предоставляется.

Возможно применение YANG Patch с протоколами, отличными от RESTCONF, но это выходит за рамки документа.

Для RESTCONF клиент и сервер **должны** быть аутентифицированы в соответствии с разделом 2 [RFC8040]. Для реализаций серверов RESTCONF важно тщательно проверять пригодность всех параметров запроса на редактирование тем или иным способом. Если запрос YANG Patch не может быть выполнен целиком, в конфигурацию системы изменения не вносятся. Запрос PATCH **должен** применяться автоматически в соответствии с разделом 2 в [RFC5789].

Реализациям сервера RESTCONF **следует** пытаться предотвратить нарушения работы системы в результате инкрементной обработки списков YANG Patch edit. Возможна организация атак на серверы RESTCONF, основанных на порядке обработки, заданном в YANG Patch. Серверам **следует** применять в системах только полностью проверенную конфигурацию. Например, список edit, который удаляет интерфейс, затем создаёт его заново, может вызвать нарушение работы системы при инкрементном (позапном) применении правок.

Реализациям серверов RESTCONF **следует** пытаться предотвратить нарушения работы системы в результате чрезмерного расхода ресурсов при выполнении запросов редактирования YANG Patch. Иначе станут возможными атаки с попытками занять всю память или исчерпать иные ресурсы.

6. Литература

6.1. Нормативные документы

[RFC2119]

Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, [RFC 2119](https://www.rfc-editor.org/info/rfc2119), DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, [RFC 3688](https://www.rfc-editor.org/info/rfc3688), DOI 10.17487/RFC3688, January 2004, <<http://www.rfc-editor.org/info/rfc3688>>.
- [RFC5789] Dusseault, L. and J. Snell, "PATCH Method for HTTP", RFC 5789, DOI 10.17487/RFC5789, March 2010, <<http://www.rfc-editor.org/info/rfc5789>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](https://www.rfc-editor.org/info/rfc6020), DOI 10.17487/RFC6020, October 2010, <<http://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", [RFC 6241](https://www.rfc-editor.org/info/rfc6241), DOI 10.17487/RFC6241, June 2011, <<http://www.rfc-editor.org/info/rfc6241>>.
- [RFC7159] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", [RFC 7159](https://www.rfc-editor.org/info/rfc7159), DOI 10.17487/RFC7159, March 2014, <<http://www.rfc-editor.org/info/rfc7159>>.
- [RFC7230] Fielding, R., Ed., and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", [RFC 7230](https://www.rfc-editor.org/info/rfc7230), DOI 10.17487/RFC7230, June 2014, <<http://www.rfc-editor.org/info/rfc7230>>.
- [RFC7231] Fielding, R., Ed., and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <<http://www.rfc-editor.org/info/rfc7231>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", [RFC 7950](https://www.rfc-editor.org/info/rfc7950), DOI 10.17487/RFC7950, August 2016, <<http://www.rfc-editor.org/info/rfc7950>>.
- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", [RFC 7951](https://www.rfc-editor.org/info/rfc7951), DOI 10.17487/RFC7951, August 2016, <<http://www.rfc-editor.org/info/rfc7951>>.
- [RFC7952] Lhotka, L., "Defining and Using Metadata with YANG", [RFC 7952](https://www.rfc-editor.org/info/rfc7952), DOI 10.17487/RFC7952, August 2016, <<http://www.rfc-editor.org/info/rfc7952>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", [RFC 8040](https://www.rfc-editor.org/info/rfc8040), DOI 10.17487/RFC8040, January 2017, <<http://www.rfc-editor.org/info/rfc8040>>.
- [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>>.

6.2. Дополнительная литература

- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, [RFC 5226](https://www.rfc-editor.org/info/rfc5226), DOI 10.17487/RFC5226, May 2008, <<http://www.rfc-editor.org/info/rfc5226>>.

Приложение А. Пример модуля YANG

Ниже рассматривается пример модуля YANG для простого интерфейса аудиопроигрывателя (jukebox). Модуль example-jukebox описан в [RFC8040].

Диаграмма дерева YANG для модуля example-jukebox показана ниже.

```

+--rw jukebox!
  +--rw library
    | +--rw artist* [name]
    | | +--rw name string
    | | +--rw album* [name]
    | | | +--rw name string
    | | | +--rw genre? identityref
    | | | +--rw year? uint16
    | | | +--rw admin
    | | | | +--rw label? string
    | | | | +--rw catalogue-number? string
    | | | +--rw song* [name]
    | | | | +--rw name string
    | | | | +--rw location string
    | | | | +--rw format? string
    | | | | +--rw length? uint32
    | | +--ro artist-count? uint32
    | | +--ro album-count? uint32
    | | +--ro song-count? uint32
    +--rw playlist* [name]
    | +--rw name string
    | +--rw description? string
    | +--rw song* [index]
    | | +--rw index uint32
    | | +--rw id instance-identifier
  +--rw player
    +--rw gap? decimal64

```

Процедуры RPC

```

+--x play
  +--ro input
    +--ro playlist string
    +--ro song-number uint32

```

A.1. Примеры YANG Patch

В этом параграфе приведены примеры RESTCONF. В большинстве примеров используется представление JSON [RFC7159], а для некоторых приведено также представление XML [W3C.REC-xml-20081126].

A.1.1. Ошибка при добавлении ресурса

Ниже приведён пример добавления нескольких песен в имеющийся альбом. Каждая операция edit указывает 1 песню. Первая песня уже имеется в альбоме, поэтому для редактирования будет возвращаться ошибка. Остальные попытки редактирования не предпринимаются по причине отказа в первой. В примере используется формат XML.

Запрос от клиента RESTCONF приведён ниже.

```
PATCH /restconf/data/example-jukebox:jukebox/\
  library/artist=Foo%20Fighters/album=Wasting%20Light HTTP/1.1
Host: example.com
Accept: application/yang-data+xml
Content-Type: application/yang-patch+xml

<yang-patch xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-patch">
  <patch-id>add-songs-patch</patch-id>
  <edit>
    <edit-id>edit1</edit-id>
    <operation>create</operation>
    <target>/song=Bridge%20Burning</target>
    <value>
      <song xmlns="http://example.com/ns/example-jukebox">
        <name>Bridge Burning</name>
        <location>/media/bridge_burning.mp3</location>
        <format>MP3</format>
        <length>288</length>
      </song>
    </value>
  </edit>

  <edit>
    <edit-id>edit2</edit-id>
    <operation>create</operation>
    <target>/song=Rope</target>
    <value>
      <song xmlns="http://example.com/ns/example-jukebox">
        <name>Rope</name>
        <location>/media/rope.mp3</location>
        <format>MP3</format>
        <length>259</length>
      </song>
    </value>
  </edit>

  <edit>
    <edit-id>edit3</edit-id>
    <operation>create</operation>
    <target>/song=Dear%20Rosemary</target>
    <value>
      <song xmlns="http://example.com/ns/example-jukebox">
        <name>Dear Rosemary</name>
        <location>/media/dear_rosemary.mp3</location>
        <format>MP3</format>
        <length>269</length>
      </song>
    </value>
  </edit>
</yang-patch>
```

XML-отклик от сервера RESTCONF имеет показанный ниже формат.

```
HTTP/1.1 409 Conflict
Date: Thu, 26 Jan 2017 20:56:30 GMT
Server: example-server
Last-Modified: Thu, 26 Jan 2017 20:56:30 GMT
Content-Type: application/yang-data+xml

<yang-patch-status
  xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-patch">
  <patch-id>add-songs-patch</patch-id>
  <edit-status>
    <edit>
      <edit-id>edit1</edit-id>
      <errors>
        <error>
          <error-type>application</error-type>
          <error-tag>data-exists</error-tag>
          <error-path
            xmlns:jb="http://example.com/ns/example-jukebox">
            /jb:jukebox/jb:library
            /jb:artist[jb:name='Foo Fighters']
            /jb:album[jb:name='Wasting Light']
            /jb:song[jb:name='Bridge Burning']
          </error-path>
```

```

    <error-message>
      Data already exists; cannot be created
    </error-message>
  </error>
</errors>
</edit>
</edit-status>
</yang-patch-status>

```

JSON-отклик от сервера RESTCONF показан ниже для иллюстрации различий в кодировании объекта error-path. Для JSON используется кодирование instance-identifier, описанное в [RFC7951].

```

HTTP/1.1 409 Conflict
Date: Thu, 26 Jan 2017 20:56:30 GMT
Server: example-server
Last-Modified: Thu, 26 Jan 2017 20:56:30 GMT
Content-Type: application/yang-data+json

{
  "ietf-yang-patch:yang-patch-status" : {
    "patch-id" : "add-songs-patch",
    "edit-status" : {
      "edit" : [
        {
          "edit-id" : "edit1",
          "errors" : {
            "error" : [
              {
                "error-type": "application",
                "error-tag": "data-exists",
                "error-path": "/example-jukebox:jukebox/library\
                              /artist[name='Foo Fighters']\
                              /album[name='Wasting Light']\
                              /song[name='Bridge Burning']",
                "error-message":
                  "Data already exists; cannot be created"
              }
            ]
          }
        }
      ]
    }
  }
}

```

A.1.2. Успешное добавление ресурсов

В приведённом ниже примере добавляется несколько песен в имеющийся альбом.

- Каждое редактирование (edit) включает 1 строку.
- Оба редактирования выполняются успешно и создаются новые субресурсы.

Ниже показан запрос клиента RESTCONF.

```

PATCH /restconf/data/example-jukebox:jukebox/\
      library/artist=Foo%20Fighters/album=Wasting%20Light \
      HTTP/1.1
Host: example.com
Accept: application/yang-data+json
Content-Type: application/yang-patch+json

{
  "ietf-yang-patch:yang-patch" : {
    "patch-id" : "add-songs-patch-2",
    "edit" : [
      {
        "edit-id" : "edit1",
        "operation" : "create",
        "target" : "/song=Rope",
        "value" : {
          "song" : [
            {
              "name" : "Rope",
              "location" : "/media/rope.mp3",
              "format" : "MP3",
              "length" : 259
            }
          ]
        }
      },
      {
        "edit-id" : "edit2",
        "operation" : "create",
        "target" : "/song=Dear%20Rosemary",
        "value" : {
          "song" : [
            {

```



```

        "name" : "Dear Rosemary",
        "location" : "/media/dear_rosemary.mp3",
        "format" : "MP3",
        "length" : 269
      }
    ]
  }
}

```

Отклик от сервера RESTCONF показан ниже.

```

HTTP/1.1 200 OK
Date: Thu, 26 Jan 2017 20:56:30 GMT
Server: example-server
Last-Modified: Thu, 26 Jan 2017 20:56:30 GMT
Content-Type: application/yang-data+json

{
  "ietf-yang-patch:yang-patch-status" : {
    "patch-id" : "add-songs-patch-2",
    "ok" : [null]
  }
}

```

А.1.3. Вставка списка

Приведённый ниже пример демонстрирует вставку песни в имеющийся список воспроизведения (playlist). Песня 6 помещается в список Foo-One после имеющейся в нем песни 5. Операция выполняется без ошибок и отклика об ошибке не возвращается.

Запрос от клиента RESTCONF имеет вид

```

PATCH /restconf/data/example-jukebox:jukebox/\
  playlist=Foo-One HTTP/1.1
Host: example.com
Accept: application/yang-data+json
Content-Type: application/yang-patch+json

{
  "ietf-yang-patch:yang-patch" : {
    "patch-id" : "insert-song-patch",
    "comment" : "Insert song 6 after song 5",
    "edit" : [
      {
        "edit-id" : "edit1",
        "operation" : "insert",
        "target" : "/song=6",
        "point" : "/song=5",
        "where" : "after",
        "value" : {
          "example-jukebox:song" : [
            {
              "index" : 6,
              "id" : "/example-jukebox:jukebox/library\
                /artist[name='Foo Fighters']\
                /album[name='Wasting Light']\
                /song[name='Bridge Burning']"
            }
          ]
        }
      }
    ]
  }
}

```

Сервер RESTCONF передаёт показанный ниже отклик.

```

HTTP/1.1 200 OK
Date: Thu, 26 Jan 2017 20:56:30 GMT
Server: example-server
Last-Modified: Thu, 26 Jan 2017 20:56:30 GMT
Content-Type: application/yang-data+json

{
  "ietf-yang-patch:yang-patch-status" : {
    "patch-id" : "insert-song-patch",
    "ok" : [null]
  }
}

```

А.1.4. Перемещение списка

В приведённом ниже примере показано перемещение песни в имеющемся списке воспроизведения. Песня 1 в списке Foo-One перемещается в позицию после песни 3. Отметим, что параметр value не нужен для операции move. Перемещение происходит успешно и сервер не возвращает сообщения об ошибке.

Ниже представлен запрос клиента RESTCONF.

```

PATCH /restconf/data/example-jukebox:jukebox/\

```

```

playlist=Foo-One HTTP/1.1
Host: example.com
Accept: application/yang-data+json
Content-Type: application/yang-patch+json

{
  "ietf-yang-patch:yang-patch" : {
    "patch-id" : "move-song-patch",
    "comment" : "Move song 1 after song 3",
    "edit" : [
      {
        "edit-id" : "edit1",
        "operation" : "move",
        "target" : "/song=1",
        "point" : "/song=3",
        "where" : "after"
      }
    ]
  }
}

```

Отклик сервера RESTCONF имеет вид

```

HTTP/1.1 200 OK
Date: Thu, 26 Jan 2017 20:56:30 GMT
Server: example-server
Last-Modified: Thu, 26 Jan 2017 20:56:30 GMT
Content-Type: application/yang-data+json

{
  "ietf-restconf:yang-patch-status" : {
    "patch-id" : "move-song-patch",
    "ok" : [null]
  }
}

```

А.1.5. Редактирование ресурса хранилища данных

Ниже представлен пример одновременного редактирования 3 узлов данных верхнего уровня из разных модулей. Модуль foo определяет лист X, bar определяет контейнер Y с листьями A и B, baz определяет список Z с ключом C и дочерними листьями D и E.

Запрос клиента RESTCONF показан ниже.

```

PATCH /restconf/data HTTP/1.1
Host: example.com
Accept: application/yang-data+json
Content-Type: application/yang-patch+json

{
  "ietf-yang-patch:yang-patch" : {
    "patch-id" : "datastore-patch-1",
    "comment" : "Edit 3 top-level data nodes at once",
    "edit" : [
      {
        "edit-id" : "edit1",
        "operation" : "create",
        "target" : "/foo:X",
        "value" : {
          "foo:X" : 42
        }
      },
      {
        "edit-id" : "edit2",
        "operation" : "merge",
        "target" : "/bar:Y",
        "value" : {
          "bar:Y" : {
            "A" : "test1",
            "B" : 99
          }
        }
      },
      {
        "edit-id" : "edit3",
        "operation" : "replace",
        "target" : "/baz:Z=2",
        "value" : {
          "baz:Z" : [
            {
              "C" : 2,
              "D" : 100,
              "E" : false
            }
          ]
        }
      }
    ]
  }
}

```

```
}  
}
```

Отклик сервера RESTCONF имеет вид

```
HTTP/1.1 200 OK  
Date: Thu, 26 Jan 2017 20:56:30 GMT  
Server: example-server  
Last-Modified: Thu, 26 Jan 2017 20:55:30 GMT  
Content-Type: application/yang-data+json  
  
{  
  "ietf-yang-patch:yang-patch-status" : {  
    "patch-id" : "datastore-patch-1",  
    "ok" : [null]  
  }  
}
```

Благодарности

Авторы признательны Rex Fernando за вклад в этот документ.

Вклад Andy Bierman основан на работе, поддерживаемой United States Army, Space & Terrestrial Communications Directorate (S&TCD) по контракту W15P7T-13-C-A616. Выраженные в этом документе мнения, выводы, заключения и рекомендации принадлежат авторам и могут не отражать точку зрения S&TCD.

Адреса авторов

Andy Bierman
YumaWorks
Email: andy@yumaworks.com

Martin Bjorklund
Tail-f Systems
Email: mbj@tail-f.com

Kent Watsen
Juniper Networks
Email: kwatsen@juniper.net

Перевод на русский язык

Николай Малых
nmalykh@protokols.ru