

# Акронис Инфраструктура

4.7



## Заявление об авторских правах

Все права защищены.

Все остальные упоминаемые товарные знаки могут быть зарегистрированными товарными знаками соответствующих владельцев.

Распространение существенно измененных версий данного руководства запрещено без явного разрешения владельца авторских прав.

Распространение настоящих или переработанных материалов, входящих в данное руководство, в виде печатного издания (книги) запрещено без письменного разрешения их владельца.

ДОКУМЕНТАЦИЯ ПОСТАВЛЯЕТСЯ «КАК ЕСТЬ». НЕ СУЩЕСТВУЕТ НИКАКИХ ЯВНО ВЫРАЖЕННЫХ ИЛИ ПОДРАЗУМЕВАЕМЫХ ОБЯЗАТЕЛЬСТВ, ПОДТВЕРЖДЕНИЙ ИЛИ ГАРАНТИЙ, В ТОМ ЧИСЛЕ И СВЯЗАННЫХ С ТОВАРНОСТЬЮ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИЛИ ПРИГОДНОСТЬЮ ЕГО ДЛЯ ИСПОЛЬЗОВАНИЯ В ОПРЕДЕЛЕННЫХ ЦЕЛЯХ, НАСКОЛЬКО ТАКАЯ ОГРАНИЧЕННОСТЬ ОТВЕТСТВЕННОСТИ ДОПУСКАЕТСЯ ЗАКОНОМ.

С ПО или Услугой может быть предоставлен исходный код сторонних производителей. Лицензии этих сторонних производителей подробно описаны в файле `license.txt`, находящемся в корневом каталоге установки.

# Оглавление

<b>1 Об этом руководстве</b> .....	<b>5</b>
<b>2 Проверка подлинности</b> .....	<b>6</b>
<b>3 Управление пользователями</b> .....	<b>7</b>
3.1 GET service ostor-users .....	7
3.1.1 Описание .....	7
3.1.2 Запросы .....	7
3.1.3 Ответы .....	8
3.2 PUT service ostor-users .....	11
3.2.1 Описание .....	11
3.2.2 Запросы .....	11
3.2.3 Ответы .....	12
3.3 POST service ostor-users .....	13
3.3.1 Описание .....	13
3.3.2 Запросы .....	13
3.3.3 Ответы .....	14
3.4 DELETE service ostor-users .....	17
3.4.1 Описание .....	17
3.4.2 Запросы .....	17
3.4.3 Ответы .....	17
3.5 GET service ostor-buckets .....	18
3.5.1 Описание .....	18
3.5.2 Запросы .....	18
3.5.3 Ответы .....	19
<b>4 Управление учетными записями</b> .....	<b>22</b>
4.1 POST service ostor-accounts .....	22
4.1.1 Описание .....	22
4.1.2 Запросы .....	22
4.1.3 Ответы .....	23
4.2 DELETE service ostor-accounts .....	24
4.2.1 Описание .....	24
4.2.2 Запросы .....	24
4.2.3 Ответы .....	25
<b>5 Управление лимитами</b> .....	<b>26</b>
5.1 GET service ostor-limits .....	26
5.1.1 Описание .....	26

5.1.2 Запросы .....	26
5.1.3 Ответы .....	27
5.2 PUT service ostor-limits .....	28
5.2.1 Описание .....	28
5.2.2 Запросы .....	29
5.2.3 Ответы .....	31
5.3 DELETE service ostor-limits .....	33
5.3.1 Описание .....	33
5.3.2 Запросы .....	33
5.3.3 Ответы .....	34
<b>6 Управление квотами .....</b>	<b>36</b>
6.1 GET service ostor-quotas .....	36
6.1.1 Описание .....	36
6.1.2 Запросы .....	36
6.1.3 Ответы .....	37
6.2 PUT service ostor-quotas .....	40
6.2.1 Описание .....	40
6.2.2 Запросы .....	40
6.2.3 Ответы .....	41
6.3 DELETE service ostor-quotas .....	43
6.3.1 Описание .....	43
6.3.2 Запросы .....	43
6.3.3 Ответы .....	44
<b>7 Статистика использования .....</b>	<b>47</b>
7.1 GET service ostor-usage .....	47
7.1.1 Описание .....	47
7.1.2 Запросы .....	47
7.1.3 Ответы .....	48
7.2 DELETE service ostor-usage .....	50
7.2.1 Описание .....	50
7.2.2 Запросы .....	50
7.2.3 Ответы .....	51

# 1 Об этом руководстве

В этом руководстве объясняется, как использовать REST API для управления кластерами S3 на базе продукта Акронис Инфраструктура. Системный API-интерфейс позволяет администраторам хранилища управлять пользователями, учетными записями, лимитами и квотами хранилища, а также отображать статистику биллинга. Системный REST API позволяет удаленно выполнять операции аналогично функциям `ostor-s3-admin`.

## 2 Проверка подлинности

Запрос управления должен пройти аутентификацию с идентификатором ключа доступа AWS, который соответствует системному пользователю S3. Системных пользователей можно создать с помощью команды `ostor-s3-admin create-user -S`.

## 3 Управление пользователями

В этом разделе описывается, как управлять пользователями S3 и выводить список корзин S3.

### 3.1 GET service ostor-users

#### 3.1.1 Описание

Выводит информацию обо всех пользователях или пользователе с указанным адресом электронной почты или идентификатором.

#### 3.1.2 Запросы

##### Синтаксис

```
GET /?ostor-users HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>
```

```
GET /?ostor-users&emailAddress=<value> HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>
```

```
GET /?ostor-users&id=<value> HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>
```

##### Параметры

GET service ostor-users parameters

Параметр	Описание	Обязательный
emailAddress	Адрес электронной почты пользователя. Тип: строковый. Значение по умолчанию: нет.	Нет*
id	Идентификатор пользователя. Тип: строковый. Значение по умолчанию: нет.	Нет*

\*В одном запросе может быть задан только один из обязательных параметров.

Если не задан ни параметр emailAddress, ни параметр id, то в ответе выводится информация обо всех пользователях. В противном случае выводится информация о пользователе с указанным адресом электронной почты или идентификатором.

## Заголовки

В этой реализации используются только общие заголовки запросов.

### 3.1.3 Ответы

## Заголовки

В этой реализации используются только общие заголовки ответов.

## Тело

Словарь JSON с информацией о пользователях в следующем формате:

```
{
  "UserEmail" : "<email>"
  "UserId" : "<id>",
  "AWSAccessKeys" : [
    {
      "AWSAccessKeyId" : "<access_key>",
      "AWSSecretAccessKey" : "<secret_key>"
    }
  ]
}
{
  "UserEmail": "<email>",
  "UserId": "<id>",
  "State": "<state>",
  "OwnerId": "<id>",
  "Flags": ["<flag>"],
  "AWSAccessKeys": [
    {
      "AWSAccessKeyId": "<access_key>",
      "AWSSecretAccessKey": "<secret_key>"
    }
  ],
  "AccountCount": "<count>",
  "Accounts": [
    {
      "Name": "<name>",
      "AWSAccessKeys": [
        {
          "AWSAccessKeyId": "<access_key>",
          "AWSSecretAccessKey": "<secret_key>"}
      ]
    }
  ]
}
```



## Ошибки

Возвращает код ошибки 400, если задано больше одного параметра.

## Примеры

### Пример запроса № 1

Возвращает информацию обо всех пользователях.

```
GET /?ostor-users HTTP/1.1
Host: s3.example.com
Date: Wed, 24 Mar 2021 17:01:11 +0200
Authorization: <authorization_string>
```

### Пример ответа № 1

```
HTTP/1.1 200 OK
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection : keep-alive
x-amz-req-time-micros: 921
x-amz-request-id: 8000000000000016000060d778c73410
Date: Wed, 24 Mar 2021 15:01:11 GMT
Connection:keep-alive
Content-type : application/json{
  "Users": [
    {
      "UserEmail": "user1@email.com",
      "UserId": "b09693b73b3c7686",
      "State": "disabled",
      "OwnerId": "0000000000000000",
      "Flags": [
        "disabled"
      ]
    },
    {
      "UserEmail": "user2@email.com",
      "UserId": "bc6265392b818465",
      "State": "enabled",
      "OwnerId": "0000000000000000",
      "Flags": []
    },
    {
      "UserEmail": "user@example.com",
      "UserId": "f373d5175d1f3b63",
      "State": "enabled",
      "OwnerId": "0000000000000000",
      "Flags": [
        "system"
      ]
    }
  ]
}
```

```
}  
]  
}
```

### Пример запроса № 2

Возвращает информацию о пользователе с идентификатором b09693b73b3c7686.

```
GET /?ostor-users&id=b09693b73b3c7686 HTTP/1.1  
Host: s3.example.com  
Date: Wed, 24 Mar 2021 17:02:25 +0200  
Authorization: <authorization_string>
```

### Пример ответа № 2

```
HTTP/1.1 200 OK  
Server: nginx  
Content-Type: application/json  
Transfer-Encoding: chunked  
Connection: keep-alive  
Date: Wed, 24 Mar 2021 15:01:11 GMT  
x-amz-req-time-micros: 983  
x-amz-request-id: 8000000000000016000060d77d2db664  
{  
  "UserEmail": "user@email.com",  
  "UserId": "b09693b73b3c7686",  
  "State": "disabled",  
  "OwnerId": "0000000000000000",  
  "Flags": [  
    "disabled"  
  ],  
  "AWSAccessKeys": [  
    {  
      "AWSAccessKeyId": "b09693b73b3c7686FIGH",  
      "AWSSecretAccessKey": "jO2p4JBN1tWc4FEGxwZ8qW2jPCJBYp8RJ4KgBcZP"  
    }  
  ],  
  "AccountCount": "3",  
  "Accounts": [  
    {  
      "Name": "account1",  
      "AWSAccessKeys": [  
        {  
          "AWSAccessKeyId": "b09693b73b3c768613NV",  
          "AWSSecretAccessKey": "CBUpFmnpUGIXskTivgDQu4qjYksWpceGZeH6Qyct"  
        }  
      ]  
    },  
    {  
      "Name": "account2",  
      "AWSAccessKeys": [  

```

```

{
  "AWSAccessKeyId": "b09693b73b3c7686LCZ5",
  "AWSSecretAccessKey": "xLpUDFJMFM05rR9acAbUDplrPqIO6fneKNFjEB5c"
},
{
  "AWSAccessKeyId": "b09693b73b3c76866NI2",
  "AWSSecretAccessKey": "ajowU8pWSGW5ZJhA7AR9OjTrt11HmHPCJsMd247W"
}
]
},
{
  "Name": "account3",
  "AWSAccessKeys": [
    {
      "AWSAccessKeyId": "b09693b73b3c7686OVV1",
      "AWSSecretAccessKey": "EOT652BDvByLwy2qPt0VsQ6s3I0pTrfPXKdW9i75"
    },
    {
      "AWSAccessKeyId": "b09693b73b3c7686Z8BU",
      "AWSSecretAccessKey": "m8PgWFLXPeJVSWojCE3DxWDoRk80g7CMyB7xK3Hd"
    }
  ]
}
]
}
}

```

## 3.2 PUT service ostor-users

### 3.2.1 Описание

Создает нового пользователя.

### 3.2.2 Запросы

#### Синтаксис

```

PUT /?ostor-users&emailAddress=<value> HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>

```

#### Параметры

PUT service ostor-users parameters

Параметр	Описание	Обязательный
emailAddress	Адрес электронной почты пользователя.	Да

Параметр	Описание	Обязательный
	Тип: строковый. Значение по умолчанию: нет.	

## Заголовки

В этой реализации используются только общие заголовки запросов.

### 3.2.3 Ответы

## Заголовки

В этой реализации используются только общие заголовки ответов.

## Тело

Словарь JSON с информацией о пользователях в следующем формате:

```
{
  "UserEmail" : "<email>"
  "UserId" : "<id>",
  "AWSAccessKeys" : [
    {
      "AWSAccessKeyId" : "<access_key>",
      "AWSSecretAccessKey" : "<secret_key>"
    }
  ]
}
```

## Ошибки

Возвращает код ошибки 400, если одновременно задано несколько параметров.

## Примеры

### *Пример запроса*

Создает пользователя с адресом электронной почты test@test.test.

```
PUT /?ostor-users&emailAddress=test@test.test HTTP/1.1
Host: s3.example.com
Date: Thu, 07 Apr 2016 16:01:03 GMT +3:00
Authorization: <authorization_string>
```

### *Пример ответа*

```
HTTP/1.1 200 OK
x-amz-req-time-micros : 186132
```

```
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection : keep-alive
X-amz-request-id : 800000000000000030003746059efad68
Date : Thu, 07 Apr 2016 13:01:08 GMT
Content-type : application/json
{
  "UserEmail": "test@test.test",
  "UserId": "a721fc1a64f13a05",
  "AWSAccessKeys": [
    {
      "AWSAccessKeyId": "a721fc1a64f13a05OQF4",
      "AWSSecretAccessKey": "VtzYY4ZHWYzbWLUrRMSzVhB07UvD6Z5nGsAPtESV"
    }
  ]
}
```

## 3.3 POST service ostor-users

### 3.3.1 Описание

Создает или отзывает пары ключей доступа для существующих пользователей или учетных записей.

### 3.3.2 Запросы

#### Синтаксис

```
POST /?ostor-users&emailAddress=<value>&genKey HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>
```

```
POST /?ostor-users&emailAddress=<value>&revokeKey=<value> HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>
```

```
POST /?ostor-users&emailAddress=<value>&accountName=<value>&genKey HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>
```

```
POST /?ostor-users&emailAddress=<value>&accountName=<value>&revokeKey=<value>
HTTP/1.1
Host: <host>
```

Date: <date>  
Authorization: <authorization\_string>

## Параметры

POST service ostor-users parameters

Параметр	Описание	Обязательный
emailAddress	Адрес электронной почты пользователя. Тип: строковый. Значение по умолчанию: нет.	Да
accountName	Имя учетной записи. Тип: строковый. Значение по умолчанию: нет.	Нет
genKey	Генерирует новую пару ключей доступа для пользователя или учетной записи. У пользователя или учетной записи может быть только две пары ключей. Тип: флаговый. Значение по умолчанию: нет.	Нет*
revokeKey	Удаляет пару ключей, которая соответствует указанному ключу доступа. Тип: строковый. Значение по умолчанию: нет.	Нет*

\*В одном запросе может быть задан только один из обязательных параметров.

## Заголовки

В этой реализации используются только общие заголовки запросов.

### 3.3.3 Ответы

## Заголовки

В этой реализации используются только общие заголовки ответов.

## Тело

Если создается ключ, тело ответа представляет собой словарь JSON с информацией о пользователе.

```
{
  "UserEmail" : "<email>"
  "UserId" : "<id>",
  "AWSAccessKeys" : [
    {
      "AWSAccessKeyId" : "<access_key>",
      "AWSSecretAccessKey" : "<secret_key>"
    }
  ]
}
```

Если отзывается ключ, тело ответа пустое.

## Примеры

### **Пример запроса № 1**

Генерирует новую пару ключей для пользователя с адресом электронной почты user1@email.com.

```
POST /?ostor-users&emailAddress=user1@email.com&genKey HTTP/1.1
Host: s3.example.com
Date: Thu, 07 Apr 2016 15:51:13 GMT +3:00
Authorization: <authorization_string>
```

### **Пример ответа № 1**

```
HTTP/1.1 200 OK
x-amz-req-time-micros : 384103
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection : closed
x-amz-request-id : 8000000000000003000374603639905b
Date : Thu, 07 Apr 2016 12:51:09 GMT
Content-type : application/json
{
  "UserEmail": "user1@email.com",
  "UserId": "8eaa6ab4749a29b4",
  "AWSAccessKeys": [
    {
      "AWSAccessKeyId": "8eaa6ab4749a29b4034G",
      "AWSSecretAccessKey": "7spuMfShCII2tX6dFtSI7TEP7ZQbIGl1GgE0Emdy"
    },
    {
      "AWSAccessKeyId": "8eaa6ab4749a29b4EJUJ",
      "AWSSecretAccessKey": "ELzQ8CTMFCyQCGSP5InGvmJxFC9xXrEJ4CjBAA2k"
    }
  ]
}
```

### **Пример запроса № 2**

Генерирует новую пару ключей для учетной записи account1 пользователя с адресом электронной почты user1@email.com.

```
POST /?ostor-users&emailAddress=user1@email.com&accountName=account1&genKey
HTTP/1.1
Host: s3.example.com
Date: Wed, 24 Mar 2021 17:32:41 +0200
Authorization: <authorization_string>
```

### **Пример ответа № 2**

```
HTTP/1.1 200 OK
Server: nginx
Content-Type: application/json
Transfer-Encoding: chunked
Connection: keep-alive
Date: Wed, 24 Mar 2021 15:32:42 GMT
x-amz-req-time-micros: 51835
x-amz-request-id: 8000000000000016000060d7e970100a
{
  "UserEmail": "user2@email.com",
  "UserId": "bc6265392b818465",
  "AWSAccessKeys": [
    {
      "AWSAccessKeyId": "bc6265392b818465YQ0R",
      "AWSSecretAccessKey": "D6dSND8MZFSsKxp4bJFRXsCFEz3bC32nhpEzFpvP"
    }
  ]
}
```

### **Пример запроса № 3**

Отзывает пару ключей с идентификатором 8eaa6ab4749a29b4034G для пользователя с адресом электронной почты user1@email.com.

```
POST /?ostor-users&emailAddress=user1@email.com&revokeKey=8eaa6ab4749a29b4034G
HTTP/1.1
Host: s3.example.com
Date: Wed, 24 Mar 2021 17:36:57 +0200
Authorization: <authorization_string>
```

### **Пример ответа № 3**

```
HTTP/1.1 200 OK
Server: nginx
Content-Type: application/json
Transfer-Encoding: chunked
Connection: keep-alive
Date: Wed, 24 Mar 2021 15:36:58 GMT
```



```
x-amz-req-time-micros: 43652
x-amz-request-id: 8000000000000016000060d7f8b178be
```

## 3.4 DELETE service ostor-users

### 3.4.1 Описание

Удаляет пользователя с указанным адресом электронной почты или идентификатором.

### 3.4.2 Запросы

#### Синтаксис

```
DELETE /?ostor-users&emailAddress=<value> HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>
```

#### Параметры

DELETE service ostor-users parameters

Параметр	Описание	Обязательный
emailAddress	Адрес электронной почты пользователя. Тип: строковый. Значение по умолчанию: нет.	Да*
id	Идентификатор пользователя. Тип: строковый. Значение по умолчанию: нет.	Да*

\*В одном запросе может быть задан только один из обязательных параметров.

#### Заголовки

В этой реализации используются только общие заголовки запросов.

### 3.4.3 Ответы

#### Заголовки

В этой реализации используются только общие заголовки ответов.

## Тело

Пустое.

## Ошибки

Возвращает код ошибки 400, если задано больше одного обязательного параметра.

---

### Замечание

Если пользователь успешно удален, возвращается статус Status204NoContent.

---

## Примеры

### *Пример запроса*

Удаляет пользователя с адресом электронной почты test@test.test.

```
DELETE /?ostor-users&emailAddress=test@test.test HTTP/1.1
Host: s3.example.com
Date: Wed, 30 Apr 2016 22:32:00 GMT
Authorization: <authorization_string>
```

### *Пример ответа*

```
HTTP/1.1 203 No Content
x-amz-req-time-micros : 172807
Server : nginx/1.8.1
Connection : closed
x-amz-request-id : 80000000000000030005c8ca5862476a
Date : Wed, 30 Apr 2016 22:32:03 GMT
Content-type : application/xml
```

## 3.5 GET service ostor-buckets

### 3.5.1 Описание

Выводит информацию по всем корзинам или корзинам пользователя с указанным адресом электронной почты или идентификатором.

### 3.5.2 Запросы

#### Синтаксис

```
GET /?ostor-buckets HTTP/1.1
Host: <host>
```

```
Date: <date>
Authorization: <authorization_string>
```

```
GET /?ostor-buckets&emailAddress=<value> HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>
```

```
GET /?ostor-buckets&id=<value> HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>
```

## Параметры

GET service ostor-buckets parameters

Параметр	Описание	Обязательный
emailAddress	Адрес электронной почты пользователя. Тип: строковый. Значение по умолчанию: нет.	Нет*
id	Идентификатор пользователя. Тип: строковый. Значение по умолчанию: нет.	Нет*

\*В одном запросе может быть задан только один из обязательных параметров.

Если не заданы ни параметр emailAddress, ни параметр id, то в ответе перечисляются все корзины. В противном случае выводится список корзин пользователя с указанным адресом электронной почты или идентификатором.

## Заголовки

В этой реализации используются только общие заголовки запросов.

### 3.5.3 Ответы

## Заголовки

В этой реализации используются только общие заголовки ответов.

## Тело

Словарь JSON со списком корзин в следующем формате:

```

{
  "Buckets": [
    {
      "name": <name>,
      "epoch": <epoch>,
      "creation_date": <date>,
      "owner_id": <id>,
      "size":
        {
          "current": <cur>,
          "hmax": <hmax>,
          "h_integral": <hint>,
          "last_ts": <last_ts>
        }
    },
    {
      ...
    }
  ]
}

```

## Ошибки

Возвращает код ошибки 400, если задано больше одного параметра.

## Примеры

### *Пример запроса*

Возвращает информацию по всем корзинам в S3.

```

GET /?ostor-buckets HTTP/1.1
Host: s3.example.com
Date: Wed, 30 Apr 2016 22:32:00 GMT
Authorization: <authorization_string>

```

### *Пример ответа*

```

{
  "Buckets": [
    {
      "size": {
        "current": 12288,
        "h_integral": 7360512,
        "hmax": 12288,
        "last_ts": 424241
      },
      "epoch": 0,
      "owner_id": "ba7eba06129464c5",
      "name": "bucket1",
      "creation_date": "2018-05-25T17:12:00.000Z"
    }
  ]
}

```

```
},
{
  "size": {
    "current": 46700160,
    "h_integral": 28160196480,
    "hmax": 46700160,
    "last_ts": 424237
  },
  "epoch": 0,
  "owner_id": "ccbec013d9fd3918",
  "name": "bucket2",
  "creation_date": "2018-05-25T13:51:55.000Z"
},
{
  "size": {
    "current": 12288,
    "h_integral": 8036352,
    "hmax": 12288,
    "last_ts": 424186
  },
  "epoch": 0,
  "owner_id": "9d80d59edbe2862a",
  "name": "bucket3",
  "creation_date": "2018-05-23T10:30:49.000Z"
}
]}
```

## 4 Управление учетными записями

В этом разделе описывается, как управлять учетными записями S3, которые представляют собой изолированные контейнеры пользовательских корзин S3 с заданными лимитами использования.

### 4.1 POST service ostor-accounts

#### 4.1.1 Описание

Создает новую учетную запись.

#### 4.1.2 Запросы

##### Синтаксис

```
POST /?ostor-accounts&emailAddress=<value>&accountName=<value> HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>
```

```
POST /?ostor-accounts&id=<value>&accountName=<value> HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>
```

##### Параметры

POST service ostor-accounts parameters

Параметр	Описание	Обязательный
emailAddress	Адрес электронной почты пользователя. Тип: строковый. Значение по умолчанию: нет.	Нет*
id	Идентификатор пользователя. Тип: строковый. Значение по умолчанию: нет.	Нет*
accountName	Имя учетной записи. Тип: строковый. Значение по умолчанию: нет.	Да

\*В одном запросе может быть задан только один из обязательных параметров.

## Заголовки

В этой реализации используются только общие заголовки запросов.

### 4.1.3 Ответы

## Заголовки

В этой реализации используются только общие заголовки ответов.

## Тело

Словарь JSON с информацией об учетных записях в следующем формате:

```
{
  "Name" : "<name>",
  "AWSAccessKeys" : [
    {
      "AWSAccessKeyId" : "<access_key>",
      "AWSSecretAccessKey" : "<secret_key>"
    }
  ]
}
```

## Примеры

### *Пример запроса*

Создает учетную запись с именем account1 для пользователя с адресом электронной почты user1@email.com.

```
POST /?ostor-accounts&emailAddress=user1@email.com&accountName=account1 HTTP/1.1
Host: s3.example.com
Date: Wed, 24 Mar 2021 14:37:10 GMT
Authorization: <authorization_string>
```

### *Пример ответа*

```
HTTP/1.1 200 OK
Server: nginx
Content-Type: application/json
Transfer-Encoding: chunked
Connection: keep-alive
Date: Wed, 24 Mar 2021 14:37:11 GMT
x-amz-req-time-micros: 32753
x-amz-request-id: 8000000000000016000060d722e695e2
{
  "Name": "account1",
  "AWSAccessKeys": [
```

```

{
  "AWSAccessKeyId": "bc6265392b818465FQYC",
  "AWSSecretAccessKey": "iWs4rkwHMUYn8K0fPhjjAENC4QYUBIglyJhNEx4I"
}
]
}

```

## 4.2 DELETE service ostor-accounts

### 4.2.1 Описание

Удаляет учетную запись пользователя с указанным адресом электронной почты или идентификатором.

### 4.2.2 Запросы

#### Синтаксис

```

DELETE /?ostor-accounts&emailAddress=<value>&accountName=<value> HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>

```

```

DELETE /?ostor-accounts&id=<value>&accountName=<value> HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>

```

#### Параметры

DELETE service ostor-accounts parameters

Параметр	Описание	Обязательный
emailAddress	Адрес электронной почты пользователя. Тип: строковый. Значение по умолчанию: нет.	Нет*
id	Идентификатор пользователя. Тип: строковый. Значение по умолчанию: нет.	Нет*
accountName	Имя учетной записи. Тип: строковый.	Да



Параметр	Описание	Обязательный
	Значение по умолчанию: нет.	

\*В одном запросе может быть задан только один из обязательных параметров.

## Заголовки

В этой реализации используются только общие заголовки запросов.

### 4.2.3 Ответы

#### Заголовки

В этой реализации используются только общие заголовки ответов.

#### Тело

Пустое.

#### Ошибки

Возвращает код ошибки 400, если задано больше одного обязательного параметра.

---

#### Замечание

Если учетная запись успешно удалена, возвращается статус Status204NoContent.

---

## Примеры

### *Пример запроса*

Удаляет учетную запись с именем account1 для пользователя с адресом электронной почты user1@email.com.

```
DELETE /?ostor-accounts&emailAddress=user1@email.com&accountName=account1 HTTP/1.1
Host: s3.example.com
Date: Wed, 24 Mar 2021 14:53:53 GMT
Authorization: <authorization_string>
```

### *Пример ответа*

```
HTTP/1.1 204 No Content
Server: nginx
Content-Type: application/xml
Connection: keep-alive
Date: Wed, 24 Mar 2021 14:53:55 GMT
x-amz-req-time-micros: 47411
x-amz-request-id: 8000000000000016000060d75ec8e4dd
```

## 5 Управление лимитами

В этом разделе описываются лимиты скорости операций и лимиты исходящей пропускной способности, которые можно задать для пользователей и корзин S3.

### 5.1 GET service ostor-limits

#### 5.1.1 Описание

Выводит информацию о лимитах на операции и пропускной способности для указанного пользователя или корзины.

#### 5.1.2 Запросы

##### Синтаксис

```
GET /?ostor-limits&emailAddress=<value> HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>
```

```
GET /?ostor-limits&bucket=<value> HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>
```

##### Параметры

GET service ostor-limits parameters

Параметр	Описание	Обязательный
emailAddress	Адрес электронной почты пользователя. Тип: строковый. Значение по умолчанию: нет.	Да*
id	Идентификатор пользователя. Тип: строковый. Значение по умолчанию: нет.	Да*
bucket	Имя корзины. Тип: строковый. Значение по умолчанию: нет.	Да*

\*В одном запросе может быть задан только один из обязательных параметров.

## Заголовки

В этой реализации используются только общие заголовки запросов.

### 5.1.3 Ответы

## Заголовки

В этой реализации используются только общие заголовки ответов.

## Тело

Словарь JSON с информацией о лимитах для пользователя или корзины в следующем формате:

```
{
  "ops:default" : "<default_limit_value_in_ops/sec>",
  "ops:get" : "<get_ops_limit_value_in_ops/sec>",
  "ops:put" : "<put_ops_limit_value_in_ops/sec>",
  "ops:list" : "<list_ops_limit_value_in_ops/sec>",
  "ops:delete" : "<delete_ops_limit_value_in_ops/sec>",
  "bandwidth:out" : "<bandwidth_limit_value_in_kb/sec>",
}
```

Нулевое значение обозначает «без ограничений».

## Ошибки

Возвращает код ошибки 400, если одновременно задано несколько параметров.

---

### Замечание

По умолчанию лимиты отключены. Если лимиты отключены для запрашиваемого пользователя или корзины, будет возвращена ошибка. Используйте PUT ostor-limits, чтобы включить лимиты.

---

## Примеры

### *Пример запроса № 1*

Возвращает информацию о лимитах для пользователя с адресом электронной почты user1@email.com.

```
GET /?ostor-limits&emailAddress=user1@email.com HTTP/1.1
Host: s3.example.com
Date: Thu, 07 Apr 2016 14:08:55 GMT
Authorization: <authorization_string>
```

### *Пример ответа № 1*

```
HTTP/1.1 200 OK
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection: closed
x-amz-request-id : 80000000000000030005c8caec96d65b
Date : Thu, 07 Apr 2016 14:08:56 GMT
Content-type : application/json
{
  "ops:default" : "0.50",
  "ops:get" : "0.50",
  "ops:put" : "0.50",
  "ops:list" : "0.50",
  "ops:delete" : "0.50",
  "bandwidth:out" : "0"
}
```

### **Пример запроса № 2**

Возвращает информацию о лимитах для корзины bucket-1.

```
GET /?ostor-limits&bucket=bucket-1 HTTP/1.1
Host: s3.example.com
Date: Wed, 30 Apr 2016 22:32:00 GMT
Authorization: <authorization_string>
```

### **Пример ответа № 2**

```
HTTP/1.1 200 OK
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection : closed
x-amz-request-id : 80000000000000030003c6b538eedd95
Date: Wed, 30 Apr 2016 22:32:00 GMT
Content-type : application/json
{
  "ops:default" : "0",
  "ops:get" : "0",
  "ops:put" : "0",
  "ops:list" : "0",
  "ops:delete" : "0",
  "bandwidth:out" : "3.33"
}
```

## **5.2 PUT service ostor-limits**

### **5.2.1 Описание**

Задаёт значения лимитов для указанного пользователя или корзины. В одном запросе можно указать либо число операций, либо ограничение пропускной способности.

## 5.2.2 Запросы

### Синтаксис

```
PUT /?ostor-limits&emailAddress=<value> HTTP/1.1  
Host: <host>  
Date: <date>  
Authorization: <authorization_string>
```

```
PUT /?ostor-limits&bucket=<value> HTTP/1.1  
Host: <host>  
Date: <date>  
Authorization: <authorization_string>
```

### Параметры

PUT Service ostor-limits parameters

Параметр	Описание	Обязательный
emailAddress	Адрес электронной почты пользователя. Тип: строковый. Значение по умолчанию: нет.	Да*
id	Идентификатор пользователя. Тип: строковый. Значение по умолчанию: нет.	Да*
bucket	Имя корзины. Тип: строковый. Значение по умолчанию: нет.	Да
bandwidth	Включает лимиты пропускной способности. Типы лимитов пропускной способности: { out   kb/s } Тип: флаговый.	Да**
ops	Включает лимиты операций. Если параметр задан, то для всех неуказанных лимитов пропускной способности устанавливается значение 0. Типы лимитов операций: { default   ops/min, put   ops/min, get   ops/min, list   ops/min, delete   ops/min } Тип: флаговый.	Да**

Параметр	Описание	Обязательный
default	<p>Задаёт значение по умолчанию для лимитов операций. Если параметр задан, то для всех неуказанных лимитов операций устанавливается значение default, иначе устанавливается значение 0.</p> <p>Требует задания подресурса ops.</p> <p>Тип: целое число.</p> <p>По умолчанию: 0.</p>	Нет
put	<p>Задаёт значение лимита операций PUT.</p> <p>Требует задания подресурса ops.</p> <p>Тип: целое число.</p> <p>По умолчанию: default.</p>	Нет
get	<p>Задаёт значение лимита операций GET.</p> <p>Требует задания подресурса ops.</p> <p>Тип: целое число.</p> <p>По умолчанию: default.</p>	Нет
delete	<p>Задаёт значение лимита операций DELETE.</p> <p>Требует задания подресурса ops.</p> <p>Тип: целое число.</p> <p>По умолчанию: default.</p>	Нет
list	<p>Задаёт значение лимита операций LIST.</p> <p>Требует задания подресурса ops.</p> <p>Тип: целое число.</p> <p>По умолчанию: default.</p>	Нет
out	<p>Задаёт лимит исходящей пропускной способности.</p> <p>Требует задания подресурса ops.</p> <p>Тип: целое число.</p> <p>По умолчанию: 0.</p>	Нет

\*В одном запросе может быть задан только один из обязательных параметров.

\*\*В одном запросе можно задать либо параметр ops, либо параметр bandwidth.

Нулевое значение обозначает «без ограничений».

## Заголовки

В этой реализации используются только общие заголовки запросов.

### 5.2.3 Ответы

## Заголовки

В этой реализации используются только общие заголовки ответов.

## Тело

Пустое.

## Ошибки

Возвращает код ошибки 400, если указан неправильный набор параметров.

## Примеры

### **Пример запроса № 1**

Устанавливает нулевое значение для всех лимитов операций пользователя с адресом электронной почты user1@email.com.

```
PUT /?ostor-limits&emailAddress=user1@email.com&ops&default=0 HTTP/1.1
Host: s3.example.com
Date: Thu, 07 Apr 2016 14:08:55 GMT
Authorization: <authorization_string>
```

### **Пример ответа № 1**

```
HTTP/1.1 200 OK
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection: closed
x-amz-request-id : 800000000000000030005c8caec96d65b
Date : Thu, 07 Apr 2016 14:08:56 GMT
Content-type : application/json
```

### **Пример запроса № 2**

Устанавливает значение 1 оп/с для всех лимитов операций пользователя с адресом электронной почты user1@email.com.

```
PUT /?ostor-limits&emailAddress=user1@email.com&ops&default=60 HTTP/1.1
Host: s3.example.com
```

```
Date: Thu, 07 Apr 2016 14:08:55 GMT
Authorization: <authorization_string>
```

### **Пример ответа № 2**

```
HTTP/1.1 200 OK
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection: closed
x-amz-request-id : 8000000000000030005c8caec96d65b
Date : Thu, 07 Apr 2016 14:08:56 GMT
Content-type : application/json
```

### **Пример запроса № 3**

Устанавливает значение 50 Кб/с для всех лимитов `badwidth.out` корзины `testbucket`.

```
PUT /?ostor-limits&bucket=testbucket&bandwidth&out=50 HTTP/1.1
Host: s3.example.com
Date: Thu, 07 Apr 2016 14:08:55 GMT
Authorization: <authorization_string>
```

### **Пример ответа № 3**

```
HTTP/1.1 200 OK
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection: closed
x-amz-request-id : 8000000000000030005c8caec96d65b
Date : Thu, 07 Apr 2016 14:08:56 GMT
Content-type : application/json
```

### **Пример запроса № 4**

Задаёт лимиты операций для корзины `testbucket`. Новый лимит операций `PUT` – 60 оп/с, лимит операций `LIST` – 0,5 оп/с, лимиты операций `GET` и `DELETE` – 1 оп/с.

```
PUT /?ostor-limits&bucket=testbucket&ops&default=60&put=3600&list=30 HTTP/1.1
Host: s3.example.com
Date: Thu, 07 Apr 2016 14:08:55 GMT
Authorization: <authorization_string>
```

### **Пример ответа № 4**

```
HTTP/1.1 200 OK
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection: closed
x-amz-request-id : 8000000000000030005c8caec96d65b
```



Date : Thu, 07 Apr 2016 14:08:56 GMT  
Content-type : application/json

## 5.3 DELETE service ostor-limits

### 5.3.1 Описание

Задаёт значение лимита выбранного типа 0.0 (без ограничений) для указанного пользователя или корзины.

### 5.3.2 Запросы

#### Синтаксис

```
DELETE /?ostor-limits&emailAddress=<value>&ops HTTP/1.1  
Host: <host>  
Date: <date>  
Authorization: <authorization_string>
```

```
DELETE /?ostor-limits&id=<value>&ops HTTP/1.1  
Host: <host>  
Date: <date>  
Authorization: <authorization_string>
```

```
DELETE /?ostor-limits&bucket=<value>&bandwidth HTTP/1.1  
Host: <host>  
Date: <date>  
Authorization: <authorization_string>
```

#### Параметры

DELETE service ostor-limits parameters

Параметр	Описание	Обязательный
emailAddress	Адрес электронной почты пользователя. Тип: строковый. Значение по умолчанию: нет.	Да*
id	Идентификатор пользователя. Тип: строковый. Значение по умолчанию: нет.	Да*
bucket	Имя корзины.	Да*

Параметр	Описание	Обязательный
	Тип: строковый. Значение по умолчанию: нет.	
ops	Удаляет лимиты операций.	Нет
bandwidth	Удаляет лимиты пропускной способности.	Нет

\*В одном запросе может быть задан только один из обязательных параметров.

## Заголовки

В этой реализации используются только общие заголовки запросов.

### 5.3.3 Ответы

#### Заголовки

В этой реализации используются только общие заголовки ответов.

#### Тело

Пустое.

---

#### Замечание

Если лимиты успешно удалены, возвращается статус Status204NoContent.

---

## Примеры

### *Пример запроса № 1*

Удаляет все лимиты операций для пользователя с адресом электронной почты user1@email.com.

```
DELETE /?ostor-limits&emailAddress=user1@email.com&ops HTTP/1.1
Host: s3.example.com
Date: Thu, 07 Apr 2016 14:08:55 GMT
Authorization: <authorization_string>
```

### *Пример ответа № 1*

```
HTTP/1.1 204 No Content
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection: closed
x-amz-request-id : 8000000000000030005c8caec96d65b
Date : Thu, 07 Apr 2016 14:08:56 GMT
Content-type : application/json
```

### *Пример запроса № 2*

Удаляет лимиты пропускной способности для корзины testbucket.

```
DELETE /?ostor-limits&bucket=testbucket&bandwidth HTTP/1.1  
Host: s3.example.com  
Date: Thu, 07 Apr 2016 14:08:55 GMT  
Authorization: <authorization_string>
```

**Пример ответа № 2**

```
HTTP/1.1 204 No Content  
Transfer-encoding : chunked  
Server : nginx/1.8.1  
Connection: closed  
x-amz-request-id : 80000000000000030005c8caec96d65b  
Date : Thu, 07 Apr 2016 14:08:56 GMT  
Content-type : application/json
```

## 6 Управление квотами

В этом разделе описываются квоты на использование хранилища, которые можно задать для пользователей и корзин S3.

### 6.1 GET service ostor-quotas

#### 6.1.1 Описание

Выводит информацию о квотах на использование хранилища для указанного пользователя/корзины или для всех пользователей/корзин.

#### 6.1.2 Запросы

##### Синтаксис

```
GET /?ostor-quotas&emailAddress=<value> HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>
```

```
GET /?ostor-quotas&bucket=<value> HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>
```

```
GET /?ostor-quotas&default=<value> HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>
```

##### Параметры

GET service ostor-limits parameters

Параметр	Описание	Обязательный
emailAddress	Адрес электронной почты пользователя. Тип: строковый. Значение по умолчанию: нет.	Да*
id	Идентификатор пользователя. Тип: строковый. Значение по умолчанию: нет.	Да*

Параметр	Описание	Обязательный
bucket	Имя корзины. Тип: строковый. Значение по умолчанию: нет.	Да*
default	Показывает значение по умолчанию для квот. Если установлено значение user, показывает квоты по умолчанию для всех пользователей. Если установлено значение bucket, показывает квоты по умолчанию для всех корзин. Тип: строковый. Значение по умолчанию: нет.	Нет

\*В одном запросе может быть задан только один из обязательных параметров.

## Заголовки

В этой реализации используются только общие заголовки запросов.

### 6.1.3 Ответы

## Заголовки

В этой реализации используются только общие заголовки ответов.

## Тело

Словарь JSON с информацией о лимитах для пользователя или корзины в следующем формате:

```
{
  "version" : "<quota_version>",
  "type" : "{0|1}",
  "size" : "<usage_limit_value_in_bytes>"
}
```

Для параметра type 0 означает «пользователь», а 1 означает «корзина».

Для параметра size нулевое значение означает «без ограничений».

## Ошибки

Возвращает код ошибки 400, если одновременно задано несколько параметров.

## Примеры

### *Пример запроса № 1*

Возвращает информацию о квотах для пользователя с адресом электронной почты user1@email.com.

```
GET /?ostor-quotas&emailAddress=user1@email.com HTTP/1.1
Host: s3.example.com
Date: Thu, 09 Sep 2021 20:53:41 GMT
Authorization: <authorization_string>
```

#### **Пример ответа № 1**

```
HTTP/1.1 200 OK
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection: closed
x-amz-request-id : 80000000000000030005c8caec96d65b
Date : Thu, 09 Sep 2021 20:53:46 GMT
Content-type : application/json
{
  "version" : "1",
  "type" : "0",
  "size" : "1024"
}
```

#### **Пример запроса № 2**

Возвращает информацию о квотах для корзины bucket1.

```
GET /?ostor-quotas&bucket=bucket1 HTTP/1.1
Host: s3.example.com
Date: Thu, 09 Sep 2021 20:54:34 GMT
Authorization: <authorization_string>
```

#### **Пример ответа № 2**

```
HTTP/1.1 200 OK
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection : closed
x-amz-request-id : 80000000000000030003c6b538eedd95
Date: Thu, 09 Sep 2021 20:54:37 GMT
Content-type : application/json
{
  "version" : "1",
  "type" : "1",
  "size" : "256"
}
```

#### **Пример запроса № 3**

Возвращает информацию о квотах по умолчанию для пользователей

```
GET /?ostor-quotas&default=user HTTP/1.1
Host: s3.example.com
Date: Thu, 09 Sep 2021 20:57:48 GMT
Authorization: <authorization_string>
```

### **Пример ответа № 3**

```
HTTP/1.1 200 OK
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection: closed
x-amz-request-id : 80000000000000030005c8caec96d65b
Date : Thu, 09 Sep 2021 20:57:51 GMT
Content-type : application/json
{
  "version" : "1",
  "type" : "0",
  "size" : "1024"
}
```

### **Пример запроса № 4**

Возвращает информацию о квотах по умолчанию для корзины

```
GET /?ostor-quotas&default=bucket HTTP/1.1
Host: s3.example.com
Date: Thu, 09 Sep 2021 20:58:05 GMT
Authorization: <authorization_string>
```

### **Пример ответа № 4**

```
HTTP/1.1 200 OK
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection : closed
x-amz-request-id : 80000000000000030003c6b538eedd95
Date: Thu, 09 Sep 2021 20:58:09 GMT
Content-type : application/json
{
  "version" : "1",
  "type" : "1",
  "size" : "256"
}
```

## 6.2 PUT service ostor-quotas

### 6.2.1 Описание

Устанавливает значение квоты для указанного пользователя/корзины или для всех пользователей/корзин.

### 6.2.2 Запросы

#### Синтаксис

```
PUT /?ostor-quotas&emailAddress=<value>&quota-size=<value> HTTP/1.1  
Host: <host>  
Date: <date>  
Authorization: <authorization_string>
```

```
PUT /?ostor-quotas&bucket=<value>&quota-size=<value> HTTP/1.1  
Host: <host>  
Date: <date>  
Authorization: <authorization_string>
```

```
PUT /?ostor-quotas&default=<value>&quota-size=<value> HTTP/1.1  
Host: <host>  
Date: <date>  
Authorization: <authorization_string>
```

#### Параметры

PUT Service ostor-limits parameters

Параметр	Описание	Обязательный
emailAddress	Адрес электронной почты пользователя. Тип: строковый. Значение по умолчанию: нет.	Да*
id	Идентификатор пользователя. Тип: строковый. Значение по умолчанию: нет.	Да*
bucket	Имя корзины. Тип: строковый. Значение по умолчанию: нет.	Да*



Параметр	Описание	Обязательный
quota-size	Устанавливает лимит использования хранилища для отдельных пользователей/корзин, а также для всех пользователей/корзин в гигабайтах.  Тип: целое число.  По умолчанию: 0.	Да
default	Устанавливает значение по умолчанию для квот. Если установлено значение user, задает квоты по умолчанию для всех пользователей. Если установлено значение bucket, задает квоты по умолчанию для всех корзин.  Тип: строковый.  По умолчанию: нет.	Нет

\*В одном запросе может быть задан только один из обязательных параметров.

Нулевое значение обозначает «без ограничений».

## Заголовки

В этой реализации используются только общие заголовки запросов.

### 6.2.3 Ответы

#### Заголовки

В этой реализации используются только общие заголовки ответов.

#### Тело

Пустое.

#### Ошибки

Возвращает код ошибки 400, если указан неправильный набор параметров.

#### Примеры

##### **Пример запроса № 1**

Устанавливает квоту для пользователя с адресом электронной почты user1@email.com на 1024 ГБ.

```
PUT /?ostor-quotas&emailAddress=user1@email.com&ops&quota-size=1024 HTTP/1.1
Host: s3.example.com
```

```
Date: Thu, 09 Sep 2021 20:21:35 GMT
Authorization: <authorization_string>
```

### **Пример ответа № 1**

```
HTTP/1.1 200 OK
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection: closed
x-amz-request-id : 80000000000000030005c8caec96d65b
Date: Thu, 09 Sep 2021 20:21:40 GMT
Content-type : application/json
```

### **Пример запроса № 2**

Устанавливает квоту для корзины bucket1 на 256 ГБ.

```
PUT /?ostor-quotas&bucket=bucket1&quota-size=256 HTTP/1.1
Host: s3.example.com
Date: Thu, 09 Sep 2021 20:22:57 GMT
Authorization: <authorization_string>
```

### **Пример ответа № 2**

```
HTTP/1.1 200 OK
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection: closed
x-amz-request-id : 80000000000000030005c8caec96d65b
Date: Thu, 09 Sep 2021 20:23:02 GMT
Content-type : application/json
```

### **Пример запроса № 3**

Устанавливает квоту по умолчанию для пользователей на 1024 ГБ.

```
PUT /?ostor-quotas&default=user&quota-size=1024 HTTP/1.1
Host: s3.example.com
Date: Thu, 09 Sep 2021 20:24:15 GMT
Authorization: <authorization_string>
```

### **Пример ответа № 3**

```
HTTP/1.1 200 OK
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection: closed
x-amz-request-id : 80000000000000030005c8caec96d65b
Date : Thu, 09 Sep 2021 20:24:19 GMT
Content-type : application/json
```

#### Пример запроса № 4

Устанавливает квоту по умолчанию для корзины на 256 ГБ.

```
PUT /?ostor-quotas&default=bucket&quota-size=256 HTTP/1.1
Host: s3.example.com
Date: Thu, 09 Sep 2021 20:25:31
Authorization: <authorization_string>
```

#### Пример ответа № 4

```
HTTP/1.1 200 OK
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection: closed
x-amz-request-id : 80000000000000030005c8caec96d65b
Date : Thu, 09 Sep 2021 20:25:36 GMT
Content-type : application/json
```

## 6.3 DELETE service ostor-quotas

### 6.3.1 Описание

Устанавливает значение квоты на 0 (без ограничений) для указанного пользователя/корзины или для всех пользователей/корзин.

### 6.3.2 Запросы

#### Синтаксис

```
DELETE /?ostor-quotas&emailAddress=<value> HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>
```

```
DELETE /?ostor-quotas&bucket=<value> HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>
```

```
DELETE /?ostor-quotas&default=<value> HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>
```

## Параметры

DELETE service ostor-limits parameters

Параметр	Описание	Обязательный
emailAddress	Адрес электронной почты пользователя. Тип: строковый. Значение по умолчанию: нет.	Да*
id	Идентификатор пользователя. Тип: строковый. Значение по умолчанию: нет.	Да*
bucket	Имя корзины. Тип: строковый. Значение по умолчанию: нет.	Да*
default	Удаляет значение по умолчанию для квот. Если установлено значение user, удаляет квоты по умолчанию для всех пользователей. Если установлено значение bucket, удаляет квоты по умолчанию для всех корзин. Тип: строковый. По умолчанию: нет.	Нет

\*В одном запросе может быть задан только один из обязательных параметров.

## Заголовки

В этой реализации используются только общие заголовки запросов.

### 6.3.3 Ответы

## Заголовки

В этой реализации используются только общие заголовки ответов.

## Тело

Пустое.

---

### Замечание

Если квоты успешно удалены, возвращается статус Status204NoContent.

---

## Примеры

### Пример запроса № 1

Удаляет квоту для пользователя с адресом электронной почты user1@email.com.

```
DELETE /?ostor-quotas&emailAddress=user1@email.com HTTP/1.1
Host: s3.example.com
Date: Thu, 09 Sep 2021 21:13:49 GMT
Authorization: <authorization_string>
```

### Пример ответа № 1

```
HTTP/1.1 204 No Content
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection: closed
x-amz-request-id : 80000000000000030005c8caec96d65b
Date : Thu, 09 Sep 2021 21:14:03 GMT
Content-type : application/json
```

### Пример запроса № 2

Удаляет квоту для корзины bucket1.

```
DELETE /?ostor-quotas&bucket=bucket1 HTTP/1.1
Host: s3.example.com
Date: Thu, 09 Sep 2021 21:14:35 GMT
Authorization: <authorization_string>
```

### Пример ответа № 2

```
HTTP/1.1 204 No Content
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection: closed
x-amz-request-id : 80000000000000030005c8caec96d65b
Date : Thu, 09 Sep 2021 21:14:39 GMT
Content-type : application/json
```

### Пример запроса № 3

Удаляет квоты по умолчанию для пользователей.

```
DELETE /?ostor-quotas&default=user HTTP/1.1
Host: s3.example.com
Date: Thu, 09 Sep 2021 21:16:18 GMT
Authorization: <authorization_string>
```

### Пример ответа № 3

```
HTTP/1.1 204 No Content
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection: closed
x-amz-request-id : 8000000000000030005c8caec96d65b
Date : Thu, 09 Sep 2021 21:16:22 GMT
Content-type : application/json
```

#### ***Пример запроса № 4***

Удаляет квоты по умолчанию для корзин.

```
DELETE /?ostor-quotas&default=bucket HTTP/1.1
Host: s3.example.com
Date: Thu, 09 Sep 2021 21:17:01 GMT
Authorization: <authorization_string>
```

#### ***Пример ответа № 4***

```
HTTP/1.1 204 No Content
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection: closed
x-amz-request-id : 8000000000000030005c8caec96d65b
Date : Thu, 09 Sep 2021 21:17:05 GMT
Content-type : application/json
```

## 7 Статистика использования

Шлюз S3 может собирать статистику использования для корзин и пользователей S3. Собранные данные сохраняются как обычные объекты. Каждый такой объект содержит статистику за определенный период использования.

Чтобы включить сбор статистики, установите для параметра `S3_GW_USAGE_BUCKET` значение `True` в файле конфигурации шлюза (по умолчанию `/var/lib/ostor/local/gw.conf`).

Другие параметры, которые, возможно, придется настроить: `S3_GW_USAGE_PERIOD` (период использования в одном объекте статистики в секундах) и `S3_GW_USAGE_CACHE_TIMEOUT` (частота сброса статистики из памяти в хранилище в секундах).

### 7.1 GET service ostor-usage

#### 7.1.1 Описание

Выводит список существующих объектов статистики или информацию о запросах, содержащуюся в указанном объекте.

#### 7.1.2 Запросы

##### Синтаксис

```
GET /?ostor-users HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>
```

```
GET /?ostor-users&obj=object name HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>
```

##### Параметры

Этот параметр определяется подресурсом `obj`. Если подресурс `obj` не определен, ответ содержит информацию обо всех существующих объектах статистики. Иначе возвращается информация из указанного объекта `obj`.

GET service ostor-usage parameters

Параметр	Описание	Обязательный
<code>obj</code>	Имя объекта статистики. Тип: строковый.	Нет

Параметр	Описание	Обязательный
	Значение по умолчанию: нет.	

## Заголовки

В этой реализации используются только общие заголовки запросов.

### 7.1.3 Ответы

## Заголовки

В этой реализации используются только общие заголовки ответов.

## Тело

Если obj не указан:

```
{ "nr_items": number of statistics objects,
  "truncated": true if a list is truncated,
  "items": [ //list of statistics objects
    "first object's name",
    "s3-usage-obj1",
    "s3-usage-obj2",
    "s3-usage-obj3",
    ...
  ]
}
```

Если obj указан:

```
{ "fmt_version": version of response format,
  "service_id": idof a service that collected statistics,
  "start_ts": timestamp of statistics upload,
  "period": statistics upload period in seconds,
  "nr_items": number of counters,
  "items": [//list of usage counters
    {
      "key": { "bucket": "bucket-name", "epoch": bucket's epoch, "user_id": "user id", "tag": "statistics
object tag" },
      "counters": {
        "ops": { "put": count of put ops, "get": count of get ops, "list": count of list ops, "other": count
of other ops },
        "net_io": { "uploaded": number of uploaded bytes during the period,
          "downloaded": number of downloaded bytes during the period }
      }
    },
    ...
  ]
}
```



## Примеры

### Пример запроса № 1

Следующий запрос возвращает информацию обо всех объектах статистики.

```
GET /?ostor-usage /HTTP1.1
Date : Mon, 11 Apr 2016 16:43:16 GMT+3:00
Host : s3.example.com
Authorization : <authorization_string>
```

### Пример ответа № 1

```
HTTP/1.1 200 OK
x-amz-req-time-micros : 404
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection : keep-alive
x-amz-request-id : 80000000000000030006b6be3b0ae378
Date : Mon, 11 Apr 2016 13:43:16 GMT
Content-type : application/json

{"nr_items": 9,
 "truncated": false,
 "items": [
  "s3-usage-8000000000000003-2016-04-11T13:10:29.000Z-1800",
  "s3-usage-8000000000000003-2016-04-11T13:12:53.000Z-30",
  "s3-usage-8000000000000003-2016-04-11T13:13:23.000Z-30",
  "s3-usage-8000000000000003-2016-04-11T13:15:53.000Z-30",
  "s3-usage-8000000000000003-2016-04-11T13:16:23.000Z-30",
  "s3-usage-8000000000000003-2016-04-11T13:31:54.000Z-30",
  "s3-usage-8000000000000003-2016-04-11T13:33:25.000Z-30",
  "s3-usage-8000000000000003-2016-04-11T13:33:55.000Z-30",
  "s3-usage-8000000000000003-2016-04-11T13:34:25.000Z-30"
 ]
}
```

### Пример запроса № 2

Следующий запрос возвращает информацию из объекта s3-usage-8000000000000003-2016-04-11T13:33:55.000Z-30.

```
GET /?ostor-usage&obj=s3-usage-8000000000000003-2016-04-11T13:12:53.000Z-30 /HTTP1.1
Date: Mon, 11 Apr 2016 17:48:21 GMT+3:00
Host: s3.example.com
Authorization: <authorization_string>
```

### Пример ответа № 2

```
HTTP/1.1 200 OK
X-amz-req-time-micros : 576
Transfer-encoding : chunked
Server : nginx/1.8.1
Connection : keep-alive
X-amz-request-id : 8000000000000030006b6bf23c77f09
Date : Mon, 11 Apr 2016 14:48:21 GMT
Content-type : application/json

{"fmt_version": 1, "service_id": 8000000000000003,
 "start_ts": 1460380373, "period": 30, "nr_items": 2,
 "items": [
  {
    "key": {"bucket": "bucket", "epoch": 16394, "user_id": "f82c23f7823589eb", "tag": ""},
    "counters": {
      "ops": {"put": 15, "get": 0, "list": 1, "other": 0},
      "net_io": {"uploaded": 99785, "downloaded": 0}
    }
  },
  {
    "key": {"bucket": "", "epoch": 0, "user_id": "f82c23f7823589eb", "tag": ""},
    "counters": {
      "ops": {"put": 0, "get": 2, "list": 0, "other": 0},
      "net_io": {"uploaded": 0, "downloaded": 0}
    }
  }
 ]
}
```

## 7.2 DELETE service ostor-usage

### 7.2.1 Описание

Удаляет объект статистики с указанным именем.

### 7.2.2 Запросы

#### Синтаксис

```
DELETE /?ostor-users&obj=<object_name> HTTP/1.1
Host: <host>
Date: <date>
Authorization: <authorization_string>
```

## Параметры

DELETE service ostor-usage parameters

Параметр	Описание	Обязательный
obj	Имя объекта статистики. Тип: строковый. Значение по умолчанию: нет.	Нет

## Заголовки

В этой реализации используются только общие заголовки запросов.

### 7.2.3 Ответы

## Заголовки

В этой реализации используются только общие заголовки ответов.

## Тело

Пустое.

---

### Замечание

Если запрос успешен, возвращается статус Status204NoContent.

---

## Примеры

### Пример запроса

Следующий запрос удаляет объект статистики с именем s3-usage-8000000000000003-2016-04-11T13:33:55.000Z-30.

```
DELETE /?ostor-usage&obj=s3-usage-8000000000000003-2016-04-11T13:12:53.000Z-30
/HTTP1.1
Date : Mon, 11 Apr 2016 17:52:05 GMT+3:00
Host : s3.example.com
Authorization : <authorization_string>
```

### Пример ответа

```
HTTP/1.1 204 No Content
Date : Mon, 11 Apr 2016 14:52:05 GMT
x-amz-req-time-micros : 4717
Connection : keep-alive
```

x-amz-request-id : 8000000000000030006b6bf31262d2c  
Server : nginx/1.8.1