
StrongDoc Documentation

Release 0.0.10

Maxim Millen

Jun 01, 2020

Contents:

1	strongdoc package	1
1.1	Subpackages	1
1.2	strongdoc.client module	13
2	Indices and tables	15
	Python Module Index	17
	Index	19

1.1 Subpackages

1.1.1 strongdoc.api package

account module

```
class strongdoc.api.account.AccountInfo (proto_account_info)
    Bases: object

    orgid
        str

    org_email
        str

    subscription
        Subscription

    payments
        list(Payment)

    org_address
        str

    multilevel_sharing
        bool

    sharable_orgs
        list(str)

class strongdoc.api.account.Payment (proto_payment)
    Bases: object

    billed_at
        datetime.datetime
```

```
    period_start
        datetime.datetime

    period_end
        datetime.datetime

    amount
        float

    status
        str

class strongdoc.api.account.Subscription(proto_subscription)
    Bases: object

    type
        str

    status
        str

class strongdoc.api.account.UserInfo(proto_user)
    Bases: object

    userid
        str

    name
        str

    email
        optional str, otherwise None

    orgid
        optional str, otherwise None

    is_admin
        bool
```

`strongdoc.api.account.add_sharable_org(token, orgid)`
Adds another org as a sharable org. This requires an administrator privilege.

Parameters

- **token** (*str*) – The user JWT token.
- **orgid** (*str*) – The orgid of the org to add as a sharable org.

Raises `grpc.RpcError` – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns Whether the addition was a success.

Return type *bool*

`strongdoc.api.account.change_user_password(token, old_password, new_password)`
Change user password.

Parameters

- **token** (*str*) – The user JWT token.
- **old_password** (*str*) – The user's current password.
- **new_password** (*str*) – The user's new password.

Raises `grpc.RpcError` – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns Whether the change was a success.

Return type `bool`

`strongdoc.api.account.demote_user(token, userid)`

Removes admin privileges from a user. This requires an administrator privilege. A gRPC connection timeout will be implemented.

Parameters

- **token** (`str`) – The user JWT token.
- **userid** (`str`) – The userid of the user to demote.

Raises `grpc.RpcError` – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns Whether the demotion was a success.

Return type `bool`

`strongdoc.api.account.get_account_info(token)`

Gets account information such as subscription details, payments, sharable orgs, etc. This requires an administrator privilege.

Parameters **token** (`str`) – The user JWT token.

Raises `grpc.RpcError` – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns The account information

Return type `AccountInfo`

`strongdoc.api.account.get_user_info(token)`

Gets information about the active user.

Parameters **token** (`str`) – The user JWT token.

Raises `grpc.RpcError` – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns The user information

Return type `UserInfo`

`strongdoc.api.account.list_users(token)`

Lists the users of the active organization.

Parameters **token** (`str`) – The user JWT token.

Raises `grpc.RpcError` – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns A list of all the users in the organization.

Return type `list(UserInfo)`

`strongdoc.api.account.promote_user(token, userid)`

Promotes a user to admin status. This requires an administrator privilege. A gRPC connection timeout will be implemented.

Parameters

- **token** (*str*) – The user JWT token.
- **userid** (*str*) – The userid of the user to promote.

Raises `grpc.RpcError` – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns Whether the promotion was a success.

Return type `bool`

`strongdoc.api.account.register_user(token, name, password, email, make_admin)`

Registers a user for the active organization. This requires an administrator privilege. A gRPC connection timeout will be implemented.

Parameters

- **token** (*str*) – The user JWT token.
- **name** (*str*) – The new user's name.
- **password** (*str*) – The new user's password.
- **email** (*str*) – The new user's email address.
- **make_admin** (*bool*) – Whether or not the new user will be an admin.

Raises `grpc.RpcError` – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns The userID of the new user.

Return type `str`

`strongdoc.api.account.remove_organization(token, force)`

Removes an organization, deleting all data stored with the organization. This requires an administrator privilege. A gRPC connection timeout will be implemented.

Parameters

- **token** (*str*) – The user JWT token.
- **force** (*bool*) – If this is false, removal will fail if there are still data stored with the organization. This prevents accidental deletion.

Raises `grpc.RpcError` – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns Whether the removal was a success.

Return type `bool`

`strongdoc.api.account.remove_sharable_org(token, orgid)`

Removes a sharable org. This requires an administrator privilege.

Parameters

- **token** (*str*) – The user JWT token.
- **orgid** (*str*) – The orgid of the org to remove as a sharable org.

Raises `grpc.RpcError` – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns Whether the removal was a success.

Return type `bool`

`strongdoc.api.account.remove_user(token, userid)`

Removes a user from the active organization. Deletes the user if this is the user's only organization. This requires an administrator privilege. A gRPC connection timeout will be implemented.

Parameters

- **token** (*str*) – The user JWT token.
- **userid** (*str*) – The userid of the user to remove.

Raises `grpc.RpcError` – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns The number of users removed.

Return type `int`

`strongdoc.api.account.set_account_info(token, org_email, org_address)`

Set organization account info. This requires an administrator privilege.

Parameters

- **token** (*str*) – The user JWT token.
- **email** (*str*) – The organization email address. Must be a valid email address.
- **address** (*str*) – The organization address.

Raises `grpc.RpcError` – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns Whether the change was a success.

Return type `bool`

`strongdoc.api.account.set_multilevel_sharing(token, enable)`

Enables or disables multi-level sharing. This requires an administrator privilege.

Parameters

- **token** (*str*) – The user JWT token.
- **enable** (*bool*) – Whether or not multi-level sharing should be enabled.

Raises `grpc.RpcError` – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns Whether the change was a success.

Return type `bool`

`strongdoc.api.account.set_user_info(token, name, email)`

Set user info.

Parameters

- **token** (*str*) – The user JWT token.
- **name** (*str*) – The user's name. Cannot be an empty string.
- **email** (*str*) – The user's email address. Must be a valid email address.

Raises `grpc.RpcError` – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns Whether the change was a success.

Return type `bool`

document module

```
class strongdoc.api.document.DocumentMetadata(_doc)
```

```
    Bases: object
```

```
    docid
```

```
        str
```

```
    doc_name
```

```
        str
```

```
    size
```

```
        int
```

```
strongdoc.api.document.decrypt_document(token, docid, ciphertext)
```

Decrypt a document using the service. The user must provide the ciphertext returned during the encryptDocument API call.

Parameters

- **token** (*str*) – The user JWT token.
- **docid** (*str*) – The ID of the document.
- **ciphertext** (*bytes or io.BufferedReaderIOBase (must be readable)*) – The document ciphertext to be decrypted.

Returns The decrypted plaintext content of the document.

Return type *bytes*

Raises

- **TypeError** – If *ciphertext* is not of type *bytes* or type *io.BufferedReaderIOBase* and readable.
- **grpc.RpcError** – Raised by the gRPC library to indicate non-OK-status RPC termination.

```
strongdoc.api.document.decrypt_document_stream(token, docid, ciphertext, output_stream)
```

Decrypt a document using the service. The user must provide the ciphertext returned during the encryptDocument API call.

Parameters

- **token** (*str*) – The user JWT token.
- **docid** (*str*) – The ID of the document.
- **ciphertext** (*bytes or io.BufferedReaderIOBase (must be readable)*) – The document ciphertext to be decrypted.
- **output_stream** (*io.BufferedReaderIOBase (must be writable)*) – The output stream the plaintext will be written to.

Return type *None*

Raises

- **TypeError** – If *ciphertext* is not of type *bytes* or type *io.BufferedReaderIOBase* and readable, or *output_stream* is not type *io.BufferedReaderIOBase* and writable.
- **grpc.RpcError** – Raised by the gRPC library to indicate non-OK-status RPC termination.

`strongdoc.api.document.download_document(token, docid)`

Download a document from the service.

Parameters

- **token** (*str*) – The user JWT token.
- **docid** (*str*) – The ID of the document.

Raises `grpc.RpcError` – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns The downloaded document.

Return type `bytes`

`strongdoc.api.document.download_document_stream(token, docid, output_stream)`

Download a document from the service.

Parameters

- **token** (*str*) – The user JWT token.
- **docid** (*str*) – The ID of the document.
- **output_stream** (*io.BufferedIOBase (must be writable)*) – The output stream the document will be written to.

Return type `None`

Raises

- **TypeError** – If `output_stream` is not type `io.BufferedIOBase` and writable.
- `grpc.RpcError` – Raised by the gRPC library to indicate non-OK-status RPC termination.

`strongdoc.api.document.encrypt_document(token, doc_name, plaintext)`

Encrypts a document using the service, but do not store it. Instead return the encrypted ciphertext.

Parameters

- **token** (*str*) – The user JWT token.
- **doc_name** (*str*) – The name of the document.
- **plaintext** (*bytes or io.BufferedIOBase (must be readable)*) – The text of the document.

Raises

- **TypeError** – If `plaintext` is not of type `bytes` or type `io.BufferedIOBase` and readable.
- `grpc.RpcError` – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns

- **docID** (*str*) – The document ID for the uploaded document. This ID is needed to decrypt the document.
- **ciphertext** (*bytes*) – The encrypted ciphertext of the document.

`strongdoc.api.document.encrypt_document_stream(token, doc_name, plaintext, output_stream)`

Encrypts a document using the service, but do not store it. Instead return the encrypted ciphertext.

Parameters

- **token** (*str*) – The user JWT token.
- **doc_name** (*str*) – The name of the document.
- **plaintext** (*bytes or io.BufferedReaderIOBase (must be readable)*) – The text of the document.
- **output_stream** (*io.BufferedReaderIOBase (must be writable)*) – The output stream the ciphertext will be written to.

Raises

- **TypeError** – If *plaintext* is not of type *bytes* or type *io.BufferedReaderIOBase* and readable, or *output_stream* is not type *io.BufferedReaderIOBase* and writable.
- **grpc.RpcError** – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns The document ID for the uploaded document. This ID is needed to decrypt the document.

Return type *str*

`strongdoc.api.document.list_documents(token)`

Lists the documents that are accessible to the active user.

Parameters **token** (*str*) – The user JWT token.

Raises **grpc.RpcError** – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns A list of documents

Return type *list(DocumentMetadata)*

`strongdoc.api.document.remove_document(token, docid)`

Remove a document from the service.

Parameters

- **token** (*str*) – The user JWT token.
- **docid** (*str*) – The ID of the document.

Raises **grpc.RpcError** – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns Whether the removal was a success.

Return type *bool*

`strongdoc.api.document.share_document(token, docid, userid)`

Shares a document with another user.

Parameters

- **token** (*str*) – The user JWT token.
- **docid** (*str*) – The DocID of the document.
- **userid** (*str*) – The UserID of the user to be shared with.

Raises **grpc.RpcError** – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns Whether the share was a success.

Return type `bool`

`strongdoc.api.document.unshare_document(token, docid, userid)`

Unshares a document from another user.

Parameters

- **token** (`str`) – The user JWT token.
- **docid** (`str`) – The DocID of the document.
- **userid** (`str`) – The UserID of the user to be unshared from.

Raises `grpc.RpcError` – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns The number of users unshared with.

Return type `int`

`strongdoc.api.document.upload_document(token, doc_name, plaintext)`

Uploads a document to the service for storage.

Parameters

- **token** (`str`) – The user JWT token.
- **doc_name** (`str`) – The name of the document.
- **plaintext** (`bytes` or `io.BufferedReaderIOBase` (must be readable)) – The text of the document.

Returns The uploaded document ID.

Return type `str`

Raises

- **TypeError** – If *plaintext* is not of type `bytes` or type `io.BufferedReaderIOBase` and readable.
- **grpc.RpcError** – Raised by the gRPC library to indicate non-OK-status RPC termination.

login module

`strongdoc.api.login.login(userid, password, orgid)`

Verify the user and organization identity, and returns a JWT token for future API use. There must be a one second difference between logout and login. A gRPC connection timeout will be implemented.

Parameters

- **userid** (`str`) – The login user ID
- **password** (`str`) – The login user password
- **orgid** (`str`) – The login organization ID

Raises `grpc.RpcError` – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns The JWT token used to authenticate user/org when using StrongDoc APIs.

Return type `str`

`strongdoc.api.login.logout(token)`

Logs out the user associated with the specified token.

Parameters `token` (*str*) – The user JWT token.

Returns The logout status of the user.

Return type *str*

search module

class `strongdoc.api.search.DocumentResult(docid, score)`

Bases: *object*

A class that will hold a single document that matches the search result from the Search query.

docid

str - The matching document ID.

score

float - The score of the matching document.

`strongdoc.api.search.search(token, query)`

Search for document that contains a specific word. A gRPC connection timeout will be implemented.

Parameters

- **token** (*str*) – The user JWT token.
- **query** (*str*) – The query string.

Raises `grpc.RpcError` – Raised by the gRPC library to indicate non-OK-status RPC termination.

Returns The hit list of the search.

Return type *list(DocumentResult)*

billing module

`strongdoc.api.billing.FREQ_MONTHLY = 'MONTHLY'`

`strongdoc.api.billing.FREQ_YEARLY = 'YEARLY'`

class `strongdoc.api.billing.BillingFrequency(_freq)`

Bases: *object*

frequency

str (such as *FREQ_MONTHLY* or *FREQ_YEARLY*)

valid_from

datetime.datetime

valid_to

datetime.datetime

class `strongdoc.api.billing.DocumentCosts(_costs)`

Bases: *object*

cost

float

```
    size
        float

    tier
        str

class strongdoc.api.billing.SearchCosts (_costs)
    Bases: object

    cost
        float

    size
        float

    tier
        str

class strongdoc.api.billing.TrafficCosts (_costs)
    Bases: object

    cost
        float

    incoming
        float - Traffic sent to us, in MB

    outgoing
        float - Traffic sent to you, in MB

    tier
        str

class strongdoc.api.billing.BillingDetails (_details)
    Bases: object

    period_start
        datetime.datetime

    period_end
        datetime.datetime

    total_cost
        float

    documents
        DocumentCosts

    search
        SearchCosts

    traffic
        TrafficCosts

    billing_frequency
        BillingFrequency

class strongdoc.api.billing.LargeTrafficDetails (proto_large_traffic)
    Bases: object

    large_traffic_list
        list(TrafficDetails)
```

period_start
`datetime.datetime` - Start of the billing period

period_end
`datetime.datetime` - End of the billing period

class `strongdoc.api.billing.TrafficDetails` (*proto_traffic*)

Bases: `object`

time
`datetime.datetime`

userid
`str`

method
`str` - “STREAM” or “UNARY”

uri
`str` - The operation this traffic corresponds to

incoming
`float` - Traffic sent to us, in MB

outgoing
`float` - Traffic sent to you, in MB

`strongdoc.api.billing.get_billing_details` (*token*, *at_time=None*)

Gets the billing details at a specified time, or the current time if no time is specified. This requires an administrator privilege.

Parameters

- **token** (`str`) – The user JWT token.
- **at_time** (`datetime.datetime`) – Optional UTC timestamp to check for billing details at. Defaults to current time.

Raises

- **grpc.RpcError** – Raised by the gRPC library to indicate non-OK-status RPC termination.
- **ValueError** – If *at_time* is not a valid datetime.

Returns The requested billing details, or *None* if none are found for the specified time.

Return type *BillingDetails* or *None*

`strongdoc.api.billing.get_large_traffic` (*token*, *at_time=None*)

Gets a list of large traffic events (at least 512 MB in one direction) within a specified billing period. This requires an administrator privilege.

Parameters

- **token** (`str`) – The user JWT token.
- **at_time** (`datetime.datetime`) – Optional UTC timestamp which falls within the desired billing period. Defaults to current time.

Returns The requested large traffic details, or *None* if none are found for the specified time.

Return type *LargeTrafficDetails* or *None*

Raises

- **ValueError** – If *at_time* is not a valid datetime.
- **grpc.RpcError** – Raised by the gRPC library to indicate non-OK-status RPC termination.

1.2 strongdoc.client module

`strongdoc.client.connect_to_server_with_auth(token)`

Creates a secure channel with an authentication token to the StrongDoc Server

Parameters `token` (*str*) – The user JWT token

Returns `channel` – A secure gRPC SSL channel.

`strongdoc.client.connect_to_server_with_no_auth()`

Creates a secure channel to the StrongDoc Server

Returns `channel` – A secure gRPC SSL channel.

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

S

`strongdoc.api.account`, [1](#)
`strongdoc.api.billing`, [10](#)
`strongdoc.api.document`, [6](#)
`strongdoc.api.login`, [9](#)
`strongdoc.api.search`, [10](#)
`strongdoc.client`, [13](#)

A

AccountInfo (class in *strongdoc.api.account*), 1
 add_sharable_org() (in module *strongdoc.api.account*), 2
 amount (*strongdoc.api.account.Payment* attribute), 2

B

billed_at (*strongdoc.api.account.Payment* attribute), 1
 billing_frequency (*strongdoc.api.billing.BillingDetails* attribute), 11
 BillingDetails (class in *strongdoc.api.billing*), 11
 BillingFrequency (class in *strongdoc.api.billing*), 10

C

change_user_password() (in module *strongdoc.api.account*), 2
 connect_to_server_with_auth() (in module *strongdoc.client*), 13
 connect_to_server_with_no_auth() (in module *strongdoc.client*), 13
 cost (*strongdoc.api.billing.DocumentCosts* attribute), 10
 cost (*strongdoc.api.billing.SearchCosts* attribute), 11
 cost (*strongdoc.api.billing.TrafficCosts* attribute), 11

D

decrypt_document() (in module *strongdoc.api.document*), 6
 decrypt_document_stream() (in module *strongdoc.api.document*), 6
 demote_user() (in module *strongdoc.api.account*), 3
 doc_name (*strongdoc.api.document.DocumentMetadata* attribute), 6
 docid (*strongdoc.api.document.DocumentMetadata* attribute), 6

docid (*strongdoc.api.search.DocumentResult* attribute), 10
 DocumentCosts (class in *strongdoc.api.billing*), 10
 DocumentMetadata (class in *strongdoc.api.document*), 6
 DocumentResult (class in *strongdoc.api.search*), 10
 documents (*strongdoc.api.billing.BillingDetails* attribute), 11
 download_document() (in module *strongdoc.api.document*), 6
 download_document_stream() (in module *strongdoc.api.document*), 7

E

email (*strongdoc.api.account.UserInfo* attribute), 2
 encrypt_document() (in module *strongdoc.api.document*), 7
 encrypt_document_stream() (in module *strongdoc.api.document*), 7

F

FREQ_MONTHLY (in module *strongdoc.api.billing*), 10
 FREQ_YEARLY (in module *strongdoc.api.billing*), 10
 frequency (*strongdoc.api.billing.BillingFrequency* attribute), 10

G

get_account_info() (in module *strongdoc.api.account*), 3
 get_billing_details() (in module *strongdoc.api.billing*), 12
 get_large_traffic() (in module *strongdoc.api.billing*), 12
 get_user_info() (in module *strongdoc.api.account*), 3

I

incoming (*strongdoc.api.billing.TrafficCosts* attribute), 11

incoming (*strongdoc.api.billing.TrafficDetails* attribute), 12
is_admin (*strongdoc.api.account.UserInfo* attribute), 2

L

large_traffic_list (*strongdoc.api.billing.LargeTrafficDetails* attribute), 11
LargeTrafficDetails (class in *strongdoc.api.billing*), 11
list_documents() (in module *strongdoc.api.document*), 8
list_users() (in module *strongdoc.api.account*), 3
login() (in module *strongdoc.api.login*), 9
logout() (in module *strongdoc.api.login*), 9

M

method (*strongdoc.api.billing.TrafficDetails* attribute), 12
multilevel_sharing (*strongdoc.api.account.AccountInfo* attribute), 1

N

name (*strongdoc.api.account.UserInfo* attribute), 2

O

org_address (*strongdoc.api.account.AccountInfo* attribute), 1
org_email (*strongdoc.api.account.AccountInfo* attribute), 1
orgid (*strongdoc.api.account.AccountInfo* attribute), 1
orgid (*strongdoc.api.account.UserInfo* attribute), 2
outgoing (*strongdoc.api.billing.TrafficCosts* attribute), 11
outgoing (*strongdoc.api.billing.TrafficDetails* attribute), 12

P

Payment (class in *strongdoc.api.account*), 1
payments (*strongdoc.api.account.AccountInfo* attribute), 1
period_end (*strongdoc.api.account.Payment* attribute), 2
period_end (*strongdoc.api.billing.BillingDetails* attribute), 11
period_end (*strongdoc.api.billing.LargeTrafficDetails* attribute), 12
period_start (*strongdoc.api.account.Payment* attribute), 1
period_start (*strongdoc.api.billing.BillingDetails* attribute), 11
period_start (*strongdoc.api.billing.LargeTrafficDetails* attribute), 11

promote_user() (in module *strongdoc.api.account*), 3

R

register_user() (in module *strongdoc.api.account*), 4
remove_document() (in module *strongdoc.api.document*), 8
remove_organization() (in module *strongdoc.api.account*), 4
remove_sharable_org() (in module *strongdoc.api.account*), 4
remove_user() (in module *strongdoc.api.account*), 4

S

score (*strongdoc.api.search.DocumentResult* attribute), 10
search (*strongdoc.api.billing.BillingDetails* attribute), 11
search() (in module *strongdoc.api.search*), 10
SearchCosts (class in *strongdoc.api.billing*), 11
set_account_info() (in module *strongdoc.api.account*), 5
set_multilevel_sharing() (in module *strongdoc.api.account*), 5
set_user_info() (in module *strongdoc.api.account*), 5
sharable_orgs (*strongdoc.api.account.AccountInfo* attribute), 1
share_document() (in module *strongdoc.api.document*), 8
size (*strongdoc.api.billing.DocumentCosts* attribute), 10
size (*strongdoc.api.billing.SearchCosts* attribute), 11
size (*strongdoc.api.document.DocumentMetadata* attribute), 6
status (*strongdoc.api.account.Payment* attribute), 2
status (*strongdoc.api.account.Subscription* attribute), 2
strongdoc.api.account (module), 1
strongdoc.api.billing (module), 10
strongdoc.api.document (module), 6
strongdoc.api.login (module), 9
strongdoc.api.search (module), 10
strongdoc.client (module), 13
Subscription (class in *strongdoc.api.account*), 2
subscription (*strongdoc.api.account.AccountInfo* attribute), 1

T

tier (*strongdoc.api.billing.DocumentCosts* attribute), 11
tier (*strongdoc.api.billing.SearchCosts* attribute), 11
tier (*strongdoc.api.billing.TrafficCosts* attribute), 11

`time` (*strongdoc.api.billing.TrafficDetails* attribute), [12](#)
`total_cost` (*strongdoc.api.billing.BillingDetails* attribute), [11](#)
`traffic` (*strongdoc.api.billing.BillingDetails* attribute), [11](#)
`TrafficCosts` (class in *strongdoc.api.billing*), [11](#)
`TrafficDetails` (class in *strongdoc.api.billing*), [12](#)
`type` (*strongdoc.api.account.Subscription* attribute), [2](#)

U

`unshare_document()` (in module *strongdoc.api.document*), [9](#)
`upload_document()` (in module *strongdoc.api.document*), [9](#)
`uri` (*strongdoc.api.billing.TrafficDetails* attribute), [12](#)
`userid` (*strongdoc.api.account.UserInfo* attribute), [2](#)
`userid` (*strongdoc.api.billing.TrafficDetails* attribute), [12](#)
`UserInfo` (class in *strongdoc.api.account*), [2](#)

V

`valid_from` (*strongdoc.api.billing.BillingFrequency* attribute), [10](#)
`valid_to` (*strongdoc.api.billing.BillingFrequency* attribute), [10](#)