The TDC Erhverv Sync API

API version: 2.5
Document revision: 42

Index

1 Server

Base URL: https://selvbetjening.sky.tdc.dk/v1/api-sync/<customerId>/

Use only HTTPS. Concatenate the individual service endpoints to the server base URL.

The path element <customerId> is the global ID that identifies your organization. It will be provided for you. Don't include "<" and ">". A valid id consists only of digits with no leading or trailing spaces.

1.1 Authentication

Use client side HTTP **Basic authentication**.

A **login-user** with the required access rights will be created for your customer account in TDC Erhverv's system. An email address must be registered with the login-user. This email address will be contacted automatically, by TDC Erhverv's ticket system, if any unexpected errors occur that can not be resolved internally by the out support team.

The login-user should be unique for every customerId. Even if your organization includes several customer accounts, we recommend setting up a separate login-user for each one.

1.2 Access

If the response from a service is not **HTTP-Status: 200** then the client should assume that the message was not received by the API and that the embedded action has not been carried out. It is up to the client to either try again or escalate the event to TDC Erhverv's regular support.

The server will in general try to be optimistic and accept (with status 200) requests that it knows to be bad in the hope that TDC Erhverv's internal support team can fix them. Depending on the severity of the error the requested action may be carried out later, after human inspection, or result in a Zendesk support case sent to the login-user's email address as described in the previous section.

The client should be ready to deal with, at least, the following special error cases:

- HTTP-Status: 404 (Not Found)
- HTTP-Status: 401 (Unauthorized)
- HTTP-Status: 403 (Forbidden)
- HTTP-STATUS: 408 (Request Timeout)
- host unreachable
- DNS lookup failure
- SSL handshake failure
- HTTP-Status: 5xx (Internal Server Error)

The client must take special care in case of client side request time out or server side request time out (HTTP-Status: 408) since the status of the request will be ambiguous.

2 About the Services

A successful API request relies on three layers of communication working together. Usually this will happen transparently from the clients point of view, but for completeness we describe it here.

Successful HTTPS connection → Successful server auth → Successfully executed API request.

If you see a HTTP response with a status code then your request has successfully navigated the DNS lookup, various network gateways and an SSL hand shake to reach the TDC Erhverv server (congratulations!). The server's response will be UTF-8 JSON in the following format:

```
{
  "data": ... ,
  "message": {}, If the HTTP-Status is not 200, this may contain a
                 helpful error message
  "status": ""   This is the server response, will be either “success” or “fail”.
                 Fail: Please contact TDC Erhverv if the source of the
                   problem is not obvious from the HTTP-Status
                   and the “message: above.
                 Success: no matter what the status of the “data”
                   above TDC Erhverv has received your request.
}
```

If the `status` element in the server response JSON is `success` the content of the JSON element `data` will be the response from the API in the format described below.

The server will typically reject you if your credentials are bad or if the requested API endpoint could not be found. If `status` element of the response JSON is anything other than `success`, please do not try to parse the JSON element called `data`, just log the event and contact an administrator. Your request may have reached the server but was never passed to the API and as such the API will not be able to start a support ticket on your behalf and TDC Erhverv will not be aware that your request failed.

2.1 API Output Schema:

If the server returned “`status: success`” then the imbedded `data` element will be the API response in the following form:

```
"data": {                  This is the API response, this object is only not null
                           if server status is “success”.
  "status_code": ,         1--> “OK”, “0” --> “FAIL”
  "status_message": "",    Will be either “OK” or “FAIL”
                           OK: your order has been made
                           FAIL: your order could not be made, but there is no
                             need to contact us or resend the order, we will fix
                             it and contact you if this can not be done.
  "request_id": "",        The unique ID of your request
  "response": {            If the API status is “OK” this list will contain the id
                           of the order(s) that where made on your behalf.
    "order_ids": [
      "SB42", "ZD42"
    ]
  }
```

If the value of `data` indicates some kind of problem, TDC Erhverv will handle this and contact you if the issue can not be fixed. It is possible that the server responds `success` while the API ultimately responds `FAIL`.

3 About this API

At this time we do not have an official set of guidelines for client developers, instead we would like to offer the following design considerations. Hopefully this will give you a better idea about the type of client this API was designed for.

We envision that this API shall be used in an environment where it is technically challenging to run a smart/interactive/statefull client. For this reason our goal is to provide a data-sync service where the client need not know exactly what kind of change took place on the local system (was a new user added or was it just a department change of an existing user). The client should always send all available information on the user and leave it up to the server to figure out what series of changes need to happen to make reality reflect the request from the client.

Some times the input to the server will be logically impossible - e.g. the use of a phone number that is already taken by another user or request for a short-dial number that starts with "1". In these situations the server will assume that the client is incapable of resolving the data conflict in the local system and will instead create support ticket so that humans can sort the issue out.

You can let the client signal errors from the server by inspecting the returned status code but in case this is not a practical way of alerting your staff, the server will assign any error-tickets to the email address given as credential during client authentication.

The API does provide some services to lookup the current state of your account with TDC Erhverv (e.g. available phonenumbers or products on offer etc.). We expect that most customers will draw on this information out-of-band of the main data sync action. This could be part of a parallel system that operates when a new user is entered in to the local system (maybe to give the admin entering the data a dropdown with available desktop phone types). This is optional and can happen before the data sync step, where the clients just takes what-ever data, assumes it is correct and sends it to the server.

4 Service GET: Products

Use this service to list information about the subscriptiontype products available for your customer account.

Endpoint: /products

Method: GET

Response Content type: application/json; charset=utf-8

4.1 Output Schema:

```
{
  "products" : [ <Product>, ...]  Zero or more elements.
}

<Product> : {
  "id" : ""                          Use this id as input for other services.
  "name" : ""                        Full name.
  "properties" : <Product Property>  Info about this product.
}

<Product Property> : {       Info on how to use this in later service definitions.
  "mobile" : "YES"/"NO"/"OPTIONAL",
  "external_mobile" : "YES"/"NO"/"OPTIONAL",
  "landline" : "YES"/"NO"/"OPTIONAL",
  "desktop_phone" : "YES"/"NO"/"OPTIONAL",
  "sip" : "YES"/"NO",
  "upn" : "YES"/"NO",
  "extension_regex" : ""
}
```

In a later section we describe the service Add/Update a New/Existing Subscription. Here several fields can be either required or optional depending on the Product Properties of the subscriptiontype. If you are in a situation where your client must dynamically choose between different scriptiontypes, then the Products service will show you how fill out valid requests.

The JSON field `extension_regex` is a regular expression that defines the required format of the extension (short dial) that you choose when requesting a new subscription of this type. You client does not have to support this as a dynamic check, we can promise that the regular expression will always be as follows:

- may not start with a “0” or a “1”
- if you are a Business One customer: must be exactly four digits
- if you are a Skype for Business customer: must be either three of four digits
- must be unique among all your active subscriptions

4.2 *Example Output:*

```
{
  "products": [
  {
    "id": "172",
    "name": "Business One Mobil Pro",
    "properties": {
      "mobile": "YES",
      "external_mobile": "NO",
      "landline": "YES",
      "desktop_phone": "YES",
      "sip": "NO",
      "upn": "NO",
      "extension_regex": "[2-9]{4}"
    }
  },
  {
    "id": "BASIC_E4",
    "name": "Skype for Business Cloud Basic (E4)",
    "properties": {
      "mobile": "NO",
      "external_mobile": "NO",
      "landline": "NO",
      "desktop_phone": "NO",
      "sip": "YES",
      "upn": "NO",
      "extension_regex": "[2-9]{3,4}"
    }
  },
  {
    "id": "VOICE_E4",
    "name": "Skype for Business Cloud Voice (E4)",
    "properties": {
      "mobile": "NO",
      "external_mobile": "NO",
      "landline": "YES",
      "desktop_phone": "NO",
      "sip": "YES",
      "upn": "NO",
      "extension_regex": "[2-9]{3,4}"
    }
  } ]
}
```

5 Service GET: Available Landline Numbers

This service lists unused landline numbers that are ready to be assigned to a new subscription. Use one of these numbers as input for the service Add/Update a New/Existing Subscription when adding a new subscription.

Endpoint: /available-landlines

Method: GET

Response Content type: application/json; charset=utf-8

5.1 Output Schema:

```
{
  "available_landlines": [<Available Landline>, ...]  Zero or more elements.
}

<Available Landline>: {
  "number": "",              The phonenumber.
  "prefix" : "",             Country prefix, eg 45 for danish numbers.
  "allowed_as_primary_number": true/false  See below.
}
```

The phonenumber property is the local (without country prefix) phonenumber, without any breaks or spaces.

The prefix property is the country prefix, without any leading signs like “+” or “00”.

It's worth noticing that the list of landline numbers that are allowed for a subscription, depends on whether the landline number is the only (or “primary”) number of the subscription. The reason for this distinction is due to invoicing, and is only relevant for customers with multiple customerids.

Currently subscriptions which require the the “allowed_as_primary_number” attribute to be true are:

- Skype for Business Cloud Voice (incl. The E3/E4 variants)
- Skype for Business Cloud Mobile OO (incl. The E3/E4 variants)
- Business one IP basis
- Business One IP pro

All other subscription types, provided they actually supports landlines, can ignore the “allowed_as_primary_number” property

5.2 Example Output:

```
{

   "available_landlines": [

      {
        "number": "88872839",
        "prefix": "45",
        "allowed_as_primary_number": false
      },
      {
        "number": "88872840",
        "prefix": "45",
        "allowed_as_primary_number": true
      },
      {
        "number": "88872841",
        "prefix": "45",
        "allowed_as_primary_number": true
      },


      {
        "number": "174530206",
        "prefix": "46",
        "allowed_as_primary_number": true
      }


   ]

}
```

6 Service GET: Available Mobile Numbers

This service lists unused mobile numbers that are ready to be assigned to a new subscription without the need of a number porting process.

Endpoint: /available-mobile-numbers

Method: GET

Response Content type: application/json; charset=utf-8

6.1 *Output Schema:*

```
{
  "new_mobile_numbers" : [ <New Mobile>, ...],      Zero or more elements.
  "at_rest_mobile_numbers" : [ <Rest Mobile>, ...]  Zero or more elements.
}
<New Mobile>: {
  "number": "",            The phonenumber.
  "prefix": "",            Country prefix, eg 45 for danish numbers.
  "operator" : <Operator>  Operator of this numbers current SIM
}

<Rest Mobile>: {
  "number": "",                     The phonenumber.
  "prefix": "",                     Country prefix, eg 45 for danish numbers.
  "operator" : <Operator>,          Operator of this numbers current SIM
  "is_ready_for_use" : true/false,  Ready to use for a new subscription
  "may_be_used_from" : yyyy-MM-dd   Earliest date ready for use
}

<Operator>: {
  "id" : "",      Use this id as input for other services.
  "name" : ""     Human-readable name.
}
```

Available/Unused mobile numbers may be in one of two states. “New” numbers have been reserved for your customer, but have never been used. “At rest” numbers have previously been active but are currently assigned to a special “Resting” subscription.

It is important to note the ID of the operator associated with the given mobile number since only Mobile Plans with a matching operator are available. See: Service GET: Mobile Plans.

It takes time to shutdown an active subscription and turn the mobile numer in to a Resting Mobile - typically 24 hours *after* the subscription ends. During this time the mobile is still considered “resting” but is not availabel for use. Ensure `is_ready_for_use` is true before you use a mobile from this list.

Be aware that if you send a mobile number to the service Add/Update a New/Existing Subscription and this number (a): does not appear on the list of available mobile number and (b): is not currently in use on an active subscription - then the API will assume you wish to start a number porting process. You will ofcorse need to confirm this by email and a signed consent form, but to avoid confusion your client may match against the list of Available Mobile Numbers before sending requests.

6.2 Example Output:

```
{
  "new_mobile_numbers" : [ {
    "number": "23896028",
    "prefix": "45",
    "operator" : {
      "id" : "3",
      "name" : "TDC"
    }
  }],
  "at_rest_mobile_numbers" : [ {
    "number": "23895546",
    "prefix": "45",
    "operator" : {
      "id" : "3",
      "name" : "TDC"
    }
  },
  {
    "number": "60386522",
    "prefix": "45",
    "operator" : {
      "id" : "1",
      "name" : "Telia"
    }
  }]
}
```

7 Service GET: Mobile Plans

This service lists your customers current choices for mobile plans. You may request these as part of the input for the service Add/Update a New/Existing Subscription.

Endpoint: /mobile-plans

Method: GET

Response Content type: application/json; charset=utf-8

7.1 *Output Schema:*

```
{
  "kit_plans" : [ <Mobile Plan>, ...],  Zero or more elements.
  "data_plans" : [ <Mobile Plan>, ...]  Zero or more elements.
}
<Mobile Plan>: {
  "id" : "",                Use this id as input for other services.
  "name" : "",              Human-readable name.
  "operator": <Operator>    The SIM operator that supports this plan.
}
<Operator>: {
  "id" : "",        Use this id as input for other services.
  "name" : ""       Human-readable name.
}
```

For convenience we devide mobile plans come in two flavors “kit” and “data”. Kit plans offer a combination of flat rate offers for phone, data and SMS/MMS. Data plans only offer flat rate data. Every mobile number on an active subscription may be associated with zero or one mobile plan at a time. You are free to change between different plans and flavors once a subscription has been created. Use the plan id order these add-ons when creating a new subscription or when updating an existing subscriptions.

Every plan is associated with an operator. Make special note of this since only plans that match the operator on a mobile numbers SIM card may be chosen.

7.2 Example Output:

```
{
  "kit_packages" : [ {
    "id" : "351",
    "name" : "Mobil Blå fri tale/1GB/fri SMS&MMS",
    "operator": {
      "id" : "3",
      "name" : "TDC"
    }
  },
  {
    "id" : "352",
    "name" : "Mobil Grøn fri tale/5GB/fri SMS&MMS",
    "operator": {
      "id" : "1",
      "name" : "Telia"
    }
  }],
  "data_packages" : [ {
    "id" : "336",
    "name" : "Mobil data 5GB",
    "operator": {
      "id" : "3",
      "name" : "TDC"
    }
  }]
}
```

8 Service GET: Mobile Operators

When creating a new subscription with a mobile number you will need to choose the operator to handle the mobile traffic. Choose one from this list.

Endpoint: /mobile-operators

Method: GET

Response Content type: application/json; charset=utf-8

8.1 *Output Schema:*

```
{
  "mobile_operators" : [ <Mobile Operator>, ...]  One or more elements.
}


<Mobile Operator> : {
  "id" : "",     Use this id as input for other services.
  "name" : ""  Regular expression that data in this field must match.
}
```

The id of your chosen operator will be required as input to Add/Update a New/Existing Subscription.

8.2 *Example Output:*

```
{
  "mobile_operators": [
    {
      "id": "1",
      "name": "Telia"
    },
    {
      "id": "3",
      "name": "TDC"
    }
  ]
}
```

9 Service GET: Custom Fields

Some customers need to synchronize customized data values as part of their subscriptions. When this can not be modeled via TDC Erhverv's core system we offer Custom Fields. Use this service to list your Custom Fields and a description of their data format.

Endpoint: /custom-fields

Method: GET

Response Content type: application/json; charset=utf-8

9.1 *Output Schema:*

```
{
  "custom_fields" : [ <Custom Field>, ...]  Zero or more elements.
}

<Custom Field> : {
  "id" : "",    Use this id as input for other services.
  "regex" : ""  Regular expression that data in this field must match.
}
```

With every custom field we list the regular expression that customer submitted input to this field must match before it can be added to the system. If you submit a custom field value to Add/Update a New/Existing Subscription that does not match the service will fail.

9.2 *Example Output:*

```
{
  "custom_fields" : [ {
    "id" : "external_mobile",
    "regex" : "\\+{1}\\d{10}"
  }, {
    "id" : "cost_center",
    "regex" : "\\d{4,6}"
  }]
}
```

10 Service GET: Desktop Phone Types

When ordering subscriptions from the Business One line, you will need to specify what model of landline phone you are currently using.

Endpoint: /desktop-phones-types

Method: GET

Response Content type: application/json; charset=utf-8

10.1 Output Schema:

```
{
  "desktop_phones_types" : [ <Desktop Phone Type>, . . .]  Zero or more elements.
}


<Desktop Phone Type> : {
  "id" : "",    Use this id as input for other services.
  "name" : ""  Human-readable name.
}
```

Compare the model name of the phone on your desk to the names on this list and use the corresponding id as input to Add/Update a New/Existing Subscription.

10.2 Example Output:

```
{
  "desktop_phones_types" : [ {
    "name" : "Leje Avaya IP 9611G 0124",
    "id" : "389"
  }, {
    "name" : "Leje Avaya IP 9620L 0124",
    "id" : "266"
  }]
}
```

11 Service POST: Add/Update a New/Existing Subscription

This service accepts a dump of data from your system and will update the TDC Erhverv system to reflect your choices. This service will create a new subscription or update an existing subscription as needed depending on if the input `users_unique_external_id` has been seen by the API before or not (more about this later).

Endpoint: /sync

Method: POST

Input Content type: application/json; charset=utf-8

11.1 Input Schema:

```
{
  "users_unique_external_id", Required, not empty
  "subscription_type",        Required, not empty
  "first_name": "",           Required, not empty
  "last_name": "",            Required, not empty
  "initials": "",             Required, may be empty
  "title": "",                Required, may be empty
  "department": "",           Required, may be empty
  "email": "",                Required, not empty, must be a valid email
  "sip": "",                  Requirement: see product properties
  "upn": "",                  Requirement: see product properties
  "new_s4b_password": "",     Optional
  "suppress_password_reset_mail": boolean, Optional, default false
  "landline": <Landline>,     Requirement: see product properties
  "display_number": {         Optional, only with landline or mobile.
    "phonenumber": ""           Required, must be an available display number.
  },
  "mobile": <Mobile>,         Requirement: see product properties
  "external_mobile": <ExternalMobile>,    Requirement: see product properties
  "desktop_phone": {          Requirement: see product properties
    "id" : ""                   Required, id of an available phone type
  }
  "custom_fields" : [<Custom Field Value>, . . . ] Optional, zero or more
}

<Landline> : {
  "phonenumber": "",          Required, see format below.
  "extension": ""             Optional, see format below.
}

<Mobile> : {
  "phonenumber": "",  Required, see format below.
  "mobile_plan_id": "",   Optional.
  "operator_id": "",      Optional.
  "onenumber_type": "",   Optional, default 'landline'
}

<ExternalMobile> : {
  "phonenumber": ""   Required, see format below.
}

<Custom Field Value> : {
 "id" : "",     Required
 "value" : ""   Required
}
```

11.1.1 Phonenumber

Several fields include phonenumber input, the following formats are supported:

- 42140630
- 42 14 06 30
- +4542140630
- +45 42140630
- +(45) 42140630
- +(45)42140630
- +(45) 42 14 06 30

If a country code is not included, +45 (Denmark) is assumed.

11.1.2 Users_unique_external_id

This field may be any unique ID that you use internally in your company. For users created in e.g. Active Directory we suggest providing the user's GUID. It is important that you store this ID and that it persists across any changes (name, email, department, phonenumbers) that may happen to this user. This ID is used as key when you later want to delete or update this user/subscription.

If you send an id that the server has never seen before this will be interpreted as a request to **Create** a new subscription. If the id exists then this subscription will be **Re-synced** to match the remainder of the data in your request. The API was designed this way on purpose so that the client need not be aware of the state (exists/not-exists) of a given user before pushing data to TDC Erhverv.

To help existing customers integrate with the API the following extra rule applies; If no user can be found with the given external ID *and* existing user can be found with the same email but *no external ID*, then the input external ID will be added to this existing user. The service will then assume you requested a re-sync of this user and perform this action.

11.1.3 Subscription_type

Must match the id of a product.

Depending on users_unique_external_id the API deduces if you wish to create a new subscription or re-sync an existing subscription and acts accordingly.

Create: a new subscription of the chosen type will be created.

Re-sync: you must include the subscription type that the subscription was created with. If not, then this is seen as a request to upgrade to the new subscription type. If this is your intention, then all other input must be consistent with the requirements of this new subscription type.

11.1.4 Password

You may optionally include the field **new_s4b_password**, but only if you are synchronizing a subscription from the Skype for Business product line. If included, this password will be set on the Skype for Business profile associated with the user.

The password must conform to the complexity requirements set by Microsoft (https://technet.microsoft.com/en-us/library/hh994562(v=ws.11).aspx), if this is not the case then the entire synchronization request will fail.

In the case where your synchronization request is creating a new user, then the password can not be set until the user is ready. This may take several minutes. By this time the API-response has already returned status: OK. We try our best to pre-validate the password, but ultimately the decision lies with the Microsoft server software. In the event that a password is rejected after the API has responded, a support ticket will be opened with the API-user, to inform you that a new password is required. This can be done via reset password.

If you are supplying a passwords during user creation, then you may want to include the option **`suppress_password_reset_mail`**. This skips the email that is normally sent to the end user with a link to the Selfservice password reset service.

11.1.5 Landline

The requirement of this field depends on the product properties of the input subscription_type. It must include a phonenumber and may include an extension. The extension must be formatted as described in the product properties.

Depending on users_unique_external_id the API deduces if you wish to create a new subscription or re-sync an existing subscription and acts accordingly.

Create: the phonenumber must be one of the available landlines, this landline is now assigned to your subscription and available for calls. If you do not provide a value for extension, then one will be chosen for you. If you do input your own value for an extension, then be sure to conform to the rules given in the product properties.

Re-sync: you must include the phonenumber that the subscription was created with. If not, then your request will change the subscription to this new phonenumber. If you omit the extension then nothing will be changed but if you include the extension and it does not match the current value, then an extension change will be performed.

11.1.6 Display_number

Before you can use a given display number it must be registered for your customer. This number will be displayed as the caller when making calls using this subscription.

Only available for subscription types that include a landline or a mobile. If the subscription includes both landline and mobile, then the display number will only be set for the mobile.

Depending on users_unique_external_id the API deduces if you wish to create a new subscription or re-sync an existing subscription and acts accordingly.

Create: if a display number is given it is set.

Re-sync: if a display number is given it is set. If none is given, then any existing display number is removed.

11.1.7 Mobile

Check the product properties to see if a mobile field is required.

Depending on users_unique_external_id the API deduces if you wish to create a new subscription or re-sync an existing subscription and acts accordingly.

Create:

If phonenumber is from Service GET: Available Mobile Numbers then it is activated and assigned to the new subscription ready for calls. If phonenumber is not from the list, then a number porting process is started. You will need to confirm this order via a link in an email that will be sent to the email address in your request.

You can omit the Mobile Plan if you do not want one, but if you include one then the id must match Service GET: Mobile Plans.

The operator_id must be the id of an operator that is available for your account, see Service GET: Mobile Operators. You will be sendt a SIM card that uses this operator's network.

Re-sync:

You must include the same phonenumber that is stored on the existing subscription. Mobile number change is not supported. You should include the same mobile plan the subscription already has (if any). If you include the id of a new mobile plan, this will be added to your subscription replacing any previous plan. If no `mobile_plan_id` is included then any existing mobile plan will be closed.

The operator_id must be the id of the operator that this subscription was created with.

Extra:

If the product properties indicate that that mobile is not supported, then you may still include a phonenumber (but not a mobile plan). The phone number will be written to any Custom Field that has the id “mobile”. There will be no other consequences. This ensures backwards compatibility with API version 1.

Onenumber Type:

If the product supports it, you may set **onenumber_type** to “landline” or “mobile”. If the field is omitted but the product supports a onenumber type, then the a landline onenumber will be set as default.

11.1.8 External Mobile

Check the product properties to see if a mobile field is required.

Some product types allow the user profile to be linked to a mobile number that is not covered by the subscription. This could be a mobile number that is operated by another mobile carrier. We consider these numbers “external” becaue we can only provide a limited integration. The API only supports danish national numbres as external mobiles.

11.1.9 Desktop_phone

The requirement of this field depends on the product properties of the input subscription_type. If required, this must be the id of a desktop phone type.

Depending on users_unique_external_id the API deduces if you wish to create a new subscription or re-sync an existing subscription and acts accordingly.

Create: your new subscription will be configured use this type of phone.

Re-sync: changing or omitting the desktop phone type will not have any effect. Changing your current desktop phone type must be done by closing your subscription and creating a new one.

11.1.10 Custom_fields

Service GET: Custom Fields lists the fields you have to choose from. Fill out some values (remember to respect the regular expression) and include them in your request.

Depending on users_unique_external_id the API deduces if you wish to create a new subscription or re-sync an existing subscription and acts accordingly.

Create: the values for your chosen fields will be written on the new user and synchronized to the appropriate external system.

Re-sync: the values of all custom fields for the existing user are cleared and only the values in this request are written.

11.1.11 Email

The TDC Erhverv system only supports *one subscription* per unique email. If you send an email address that is already in use with another unique_external_id, then the request will fail and you will be contacted by the support team.

Depending on users_unique_external_id the API deduces if you wish to create a new subscription or re-sync an existing subscription and acts accordingly.

Create: a user with this email is created and is associated with the new subscription

Re-sync: if the input email differs from stored email, then stored email is updated.

11.1.12 Sip

The users SIP address. Like the email, it must be unique. Unlike the email, the domain must be registered with your customer profile or the request will fail.

Depending on users_unique_external_id the API deduces if you wish to create a new subscription or re-sync an existing subscription and acts accordingly.

Create: a user with this sip is created and is associated with the new subscription

Re-sync: if the input sip differs from stored sip, then stored sip is updated.

11.1.13 Upn

The users UPN address. Like the email, it must be unique. Unlike the email, the domain must be registered with your customer profile or the request will fail.

Not all customers can update the upn of their users, this depends on the ADFS settings in the cloud environment. Inspect the upn field of the Product Properties for your account - if a upn is not required the upn fieldvalue in all input JSON must be null (or just exclude the field) or the request will fail.

Depending on users_unique_external_id the API deduces if you wish to create a new subscription or re-sync an existing subscription and acts accordingly.

Create: a user with this upn is created and is associated with the new subscription

Re-sync: if the input upn differs from stored upn, then stored upn is updated.

11.1.14 First_name, last_name, initials, title & department

Depending on users_unique_external_id the API deduces if you wish to create a new subscription or re-sync an existing subscription and acts accordingly.

Create: a user with this information is created and is associated with the new subscription

Re-sync: if the input information differs from stored information, then stored information is updated.

11.1.15 Confirmation

If the API returns a “FAIL” response code then you may be certain that the order needs to be looked at by TDC Erhverv support. We will contact you via your clients authentication email if we need your help to resolve the issue.

If the API status is “OK” it usually takes only a few minutes to fulfill the order. When everything is ready a welcome-email will be sent to the new users email address and an email receipt will sent to your client authentication email. The API will only send emails when a user is first created or a user changes subscription type.

Some times an error may still occur while setting up the phone service even after the API has responded “OK”. In cases like this TDC Erhverv support will handle the issue but you may experience a significant delay before the changes take effect and any confirmation emails are sent. If the API response included one or more order-IDs, then you may send them to TDC Erhverv's support to query the current status of your order.

11.2 Example API Requests:

Syncronize a “Cloud Voice” subscription with a landline and single custom field.

```
Authorization: Basic YXBpLXVzZXJAY2lycXVlLmRrOm15UGFzd29yZA==

POST to https://selvbetjening.sky.tdc.dk/v1/api-sync/103928/sync


JSON BODY:

{

  "users_unique_external_id": "1a551afc-8ad8-4e20-bc64-613ffc2f5955",

  "subscription_type": "VOICE_E4",

  "email": "test6@cirque.dk",

  "sip": "test6@tenant4.l4.test-cirque.dk",

  "first_name": "test6",

  "last_name": "test6",

  "initials": null,

  "title": null,

  "department": null,

  "landline": {

    "phonenumber": "+45 88872837"

  },

  "custom_fields": [

    {

      "id": "mobile",

      "value": "+4642140644"

    }

  ]

}
```

Syncronize a “Business One Mobil Pro” subscription with a landline, a mobile and several custom fields.

```
Authorization: Basic YXBpLXVzZXJAY2lycXVlLmRrOm15UGFzd29yZA==
POST to https://selvbetjening.sky.tdc.dk/v1/api-sync/104244/sync

JSON BODY:
{
  "users_unique_external_id": "366daf71-a2a4-4078-b442-16735fc019a9",
  "subscription_type": "172",
  "email": "test@cirque.dk",
  "first_name": "Test",
  "last_name": "Testersen",
  "initials": "TT",
  "title": "tester",
  "department": null,
  "landline": {
    "phonenumber": "88379514",
    "extension": "9514"
  },
  "desktop_phone": {
    "id": "9620"
  },
  "display_number": {
    "phonenumber": "88379539"
  },
  "mobile": {
    "phonenumber": "52180002",
    "mobile_plan_id": "353",
    "operator_id": "1"
  },
  "custom_fields": [
    {
      "id": "site",
      "value": "my site"
    },
    {
      "id": "department-head",
      "value": "my department head"
    },
    {
      "id": "office-number",
      "value": "my office number"
    },
    {
      "id": "cost-center",
      "value": "my cost center"
    }
  ]
}
```

11.3 Example API Successful response:

HTTP-Status: 200

```
{
  "data": {
    "status_message": "OK",
    "status_code": 1,
    "request_id": "eb481d51-5bd9-4dc4-8366-6ffe706606df",
    "response": {
      "order_ids": [
        "SB42"
      ]
    }
  },
  "message": {},
  "status": "success"
}
```

When the API status is “OK” where will always be one or more order IDs in the response. The client should log these in case you want to make a query with TDC Erhverv's support team.

11.4 Example API Failed response:

HTTP-Status: 200

```
{
  "data": {
    "status_message": "FAIL",
    "status_code": 0,
    "request_id": "6adf39b4-9e10-46d7-bad3-a6d43aa2d2be",
    "response": null
  },
  "message": {},
  "status": "success"
}
```

Even if the API fails there *may* still be a “response”, but you should not count on this since it depends on the nature of the fault.

12 Service DELETE: Cancel Subscription

This service cancels an active subscription.

Endpoint: /sync/subscription?email=<user email>&users_unique_external_id=<user id>

Method: DELETE

Query Parameters:

- **email**: the email of the user who's subscription you want to cancel
- **users_unique_external_id**: the external id of the user whos subscription you want to cancel

Only one of these parameters has to be included. We recommend you use users_unique_external_id if possible to avoid conflicts with potential email changes. If no matching user can be found the service will do nothing and return with status OK - i.e. this service is idempotet.

Remember to URL encode your query parameters.

The subscription will count as cancelled starting from the day this service is called, but it may take up to 24 hour before any associated phonenumbers become suspended. If a subscription has not been active for at least two days any Cancel request will always fail.

12.1 Example API Requests:

Syncronize a “Cloud Voice” subscription with a landline and single custom field.

```
Authorization: Basic YXBpLXVzZXJAY2lycXVlLmRrOm15UGFzd29yZA==

DELETE to

https://selvbetjening.sky.tdc.dk/v1/api-sync/103928/subscription?email=test
%40cirque.dk  Rememeber URL encoding of the email!


or


https://selvbetjening.sky.tdc.dk/v1/api-sync/103928/subscription?
users_unique_external_id=1057f75f-f7c7-4019-951c-74695fb81886
```

12.2 Example API Successful response:

HTTP-Status: 200

```
{
  "data": {
    "status_message": "OK",
    "status_code": 1,
    "request_id": "4f2245dc-2b92-4f05-9d5b-f1ab1cf99e5a",
    "response": {
      "order_ids": [
        "28789"
      ]
    }
  },
  "message": {},
  "status": "success"
}
```

This is the exact same output format as used for the Add/Update a New/Existing Subscription - service. In this case the returned order id may be given to TDC Erhverv support to track the progress of the cancel order.

13 Service POST: Reset S4B Password

This service sets/resets the S4B password of an existing user.

Endpoint: /{customerid}/s4b/password

Method: POST

Input Content type: application/json; charset=utf-8

13.1 Input Schema:

```
{
  "users_unique_external_id" : "", Required, not empty
  "user_email": "", Required, not empty, must be a valid email
  "new_password" : "" Required, not empty, must conform to complexity requirements
}
```

13.2 Complexity Requirements

The password must contain at least three of the following:

- Kapital letters
- Lowercase letters
- Special characters
- Numbers

In addition the password must be 8+ characters long and may not contain runs of 3+ characters that also are part of the users name.

13.3 Success Responce

The password was changed successfully:

HTTP-Status: 200
Content type: application/json

```
{
  "status" : "SUCCESS",
  "data" : {
    "status_message" : "OK",
    "status_code" : 1,
    "request_id" : "b13478d2-1de6-4af1-8589-2683321e3ad0",
    "response" : {
      "password_was_reset" : true,
      "message" : ""
    }
  }
}
```

13.4 Failure

The password was changed successfully:

HTTP-Status: 200
Content type: application/json

```
{
  "status" : "SUCCESS",
  "data" : {
    "status_message" : "OK",
    "status_code" : 1,
    "request_id" : "594cfd80-ff65-4bac-ac29-4a923ce992f5",
    "response" : {
      "password_was_reset" : false,
      "message" : "password does not match complexity requirements"
    }
  }
}
```

The client must parse `data.response.password_was_reset` to determine if the proposed new pasword was accepted.

14 Service GET: Order Status

The response from Add/Update a New/Existing Subscription may contain a list of `order_ids`. Use this service to query the status of your orders.

Endpoint: /order-status/<order_id>

Method: GET

Response Content type: application/json; charset=utf-8

14.1 Output Schema:

```
{
  "was_order_found" : true/false,   Does the order id exist.
  "order_status" : <Order Status>   Null if not found.
}

<Order Status> : {
  "order_id": "",     The id you searched for.
  "ordered_by": "",   The login of the person/client who made the order.
  "customer_id": "",  The customerid of the account that owns the order.
  "status": ""        The current status of the order.
}
```

You can only search for orders that belong the same customer account as your login user.

Order status

- **ok** - The order has completed successfully.
- **not processed** - The order is waiting to run.
- **cancelled** - This order will never run again.
- **paused** - The order is waiting for something.
- **error** - The order has failed, we are looking in to it.

14.2 Example API Requests:

Lookup an order that exists.

```
GET https://selvbetjening.sky.tdc.dk/v1/api-sync/103928/order-status/SB15704

JSON Response:
{
  "was_order_found": true,
  "order_status": {
    "orderid": "SB15704",
    "ordered_by": "support@sky.tdc.dk",
    "customerid": "103928",
    "status": "OK"
  }
}
```

Lookup an order that does not exist.

```
GET https://selvbetjening.sky.tdc.dk/v1/api-sync/103928/order-status/SB00000

JSON Response:
{
  "was_order_found": false,
  "order_status": null
}
```

15 Service GET: Lookup Existing Subscriptions

User data may be updates by the Selfservice web interface or by the support team. To avoid your API client conflicting or requesting unnecessary updates, you may use this service to view the current status of your subscriptions.

Before synchronization you can lookup user/subscription data, modified with your changes and POST back the result. This flow ensures your updates do not remove data/phonenumbers from the subscription that may have been added since your last API update.

A search for an email or sip will also tell you if this data is free to use on a new subscription or if it is currently in use on a users that your API client may not know about.

Endpoint: /subscription
Method: GET

15.1 About errors:

Unlike the majority of the API services, a failed Lookup (in rare cases a misconfigured user or customer account may cause a server error) will not autogenerate a support ticket. If the HTTP-Status or one of the two internal status codes (`status` or `status_message`) in the response JSON indicate a failure your client should back off for two seconds and then retry a maximum of three times. If the issue persists please contact your accountmanager.

15.2 Input:

You may search by any one of the following fields:

- email
- sip
- users_unique_external_id
- phonenumber

If you input more than one search criteria the output will be all subscrptions that match at least one.

15.3 Output Schema

```
{
  "email_status": <EmailStatus>,  Only not-null if you searched by email.
  "sip_status": <SipStatus>,      Only not-null if you searched by sip.
  "phone_status": <PhoneStatus> , Only not-null if you searched by phonenumber.
  "found_by_email": <Subscription>,       Only not-null if email-search hit.
  "found_by_external_id": <Subscription>, Only not-null if id-search hit.
  "found_by_sip": <Subscription>,         Only not-null if sip-search hit.
  "found_by_phonenumber": <Subscription>  Only not-null if number-search hit.
}


<EmailStatus> : {"code" : 0, "name" : "  }
One of: [(0,"valid unused"), (1,"invalid format"), (2,"in use")]

<SipStatus> : {"code" : 0, "name" : "  }
One of: [(0,"valid unused"), (1,"invalid format"), (2,"in use"), (3,"disallowed
domain")]

<PhoneStatus> : {"code" : 0, "name" : "  }
One of: [(1,"invalid format"), (2,"in use"), (3,"not in use")]


<Subscription> : {
  "users_unique_external_id" : "", Your local id, if we have it
  "subscription_type" : "",        Id of the product type. From /products
  "email" : "",                    User email
  "sip" : "",                      User sip (or empty if non-S4B)
  "upn" : "",                      User upn (or empty if non-S4B)
  "first_name" : "",               User first name
  "last_name" : "",                User last name
  "initials" : "",                 User initials
  "title" : "",                    User title
  "department" : "",               User department
  "landline" : <Landline>,         If the subscription includes a landline
  "mobile" : <Mobile>,             If the subscription includes a mobile
  "external_mobile" : <ExternalMobile>,
                                   If the subscription includes an ext. mobile
  "custom_fields" : [<CustomField>],
                                   If the subscription has custom fields set
  "can_mobile_plan_be_changed" : true/false,
                                   Can a mobile plan be set/changed?
  "can_be_synchronized" : true/false
                                   Can a /sync operation be run on this user?
}

<Landline> : {
  "phonenumber" : "",                Format: +88 88888888
  "extension" : "",                  Phones local extension
  "displaynumber" : <DisplayNumber>  Number displayed on call out
}

<Mobile> : {
  "phonenumber" : "",               Format: +88 88888888
  "mobile_plan_id" : "",            ID of mobile plan. From /mobile-plans
  "operator_id" : "",               ID of mobile plan. From /mobile-operators
  "onenumber_type" : "",            Is "landline" or "mobile"
  "displaynumber" : <DisplayNumber> Number displayed on call out
```

```
}

<DisplayNumber> : {
  "phonenumber" : ""  Format: +88 88888888
}

<ExternalMobile> : {
  "phonenumber" : ""  Format: +88 88888888
}
```

15.4 Example:

```
GET https://selvbetjening.sky.tdc.dk/v1/api-sync/103928/subscription?email=test
%40cirque.net

JSON Response:
{
  "email_status": {
    "code": 2,
    "name": "in use"
  },
  "sip_status": null,
  "phone_status": null,
  "found_by_email": {
  "users_unique_external_id": "fb7c122d-b174-4cb6-a582-0866f78c085e",
  "subscription_type": "MOBILE",
  "email": "test@cirque.dk",
  "sip": "test@tenant4.l4.test-cirque.dk",
  "first_name": "Test",
  "last_name": "User",
  "initials": "TU",
  "title": "Tester",
  "department": " ",
  "landline": {
    "phonenumber": "+45 88872832",
    "extension": "2832",
    "displaynumber": "+45 88872832"
  },
  "mobile": {
    "phonenumber": "+45 60386651",
    "mobile_plan_id": null,
    "operator_id": "3",
    "onenumber_type": "mobile",
    "displaynumber": {
      "phonenumber": "+45 88872836"
    }
  },
  "external_mobile": null,
  "custom_fields": [{
    "id": "costcenter",
    "value": "888888"
  }],
  "can_mobile_plan_be_changed": true,
  "can_be_synchronized": true
  },
  "found_by_external_id": null,
  "found_by_sip": null,
  "found_by_phonenumber": null
}
```

```
GET https://selvbetjening.sky.tdc.dk/v1/api-sync/103928/subscription?phonenumber=
%2B45%208__88888

JSON Response:
{
  "email_status": null,
  "sip_status": null,
  "phone_status": {
    "code": 1,
    "name": "invalid format"
  },
  "found_by_email": null,
  "found_by_external_id": null,
  "found_by_sip": null,
  "found_by_phonenumber": null
}
```

16 Service GET: List All Subscriptions

This service will output a list of all Subscriptions on your customer account. This is a quick way to prime your synchronization client when starting up a new system. The output is not as detailed af Lookup Existing Subscriptions, so an additional lookup may be needed.

Endpoint: /subscriptions
Method: GET

16.1 About errors:

Unlike the majority of the API services, a failed Lookup (in rare cases a misconfigured user or customer account may cause a server error) will not autogenerate a support ticket. If the HTTP-Status or one of the two internal status codes (status or status_message) in the response JSON indicate a failure your client should back off for two seconds and then retry a maximum of three times. If the issue persists please contact your accountmanager.

16.2 Output Schema

```
[{
  "firstname": "",  User first name
  "lastname": "",   User last name
  "initials": "",   User initials
  "email": "",      User email
  "sip": "",        User sip (or empty if non-S4B)
  "upn": "",        User upn (or empty if non-S4B)
  "title": "",      User title
  "landline": "",   Subscription Landline (if present)
  "mobile": "",     Subscription Mobile (if present)
  "international_number": "", Subscription int.nr (if present)
  "number": "",               Subscription "Primary Number"
  "prefix": "",               International dial prefix of Primary Number
  "country_code": "",         ISO country code for nationality of Primary Number
  "department": "",           User department
  "termination_date": "",     Subscription End Date: yyyy-MM-dd
  "subscription_type": "",    Name of subsciption product type (see /products)
  "external_id": "",          users_unique_external_id
--- Any Custom Fields are displayed, by name, below.
--- If there is a name clash the postfix "(<n>)" - see example below.
  "cf1": "", Optional
  "cf2": "", Optional
  "cf3": ""  Optional
}
, … ]
```

16.3 Example

```
GET https://selvbetjening.sky.tdc.dk/v1/api-sync/103928/subscriptions

JSON Response:
{
  "subscriptions": [
    {
      "firstname": "Test",
      "lastname": "User",
      "initials": "TU",
      "email": "test@cirque.dk",
      "sip": "test@tenant4.14.test-cirque.dk",
      "title": "Tester",
      "landline": "88888888",
      "mobile": "",
      "international_number": "",
      "number": "+45 88888888",
      "prefix": "45",
      "country_code": "dk",
      "department": "Doc",
      "termination_date": "",
      "subscription_type": "Skype for Business Cloud Voice",
      "external_id": "44e25f71-4f71-418a-8aab-3a57d61458fb",
      "street": "Teglholmsgade 1",  Custom Field
      "home_postal_address": "",    Custom Field
      "costcenter": "campus east",  Custom Field
      "mobile(2)": "+4512345678"    Custom Field (not a "phonenumber" - just text)
    },...
  ]
}
```

17 Errors Conditions

This section is a reference guide for how the API behaves under different error conditions.

17.1 Server Error: Service Not Found

In this example: no customer was found matching the given customerId.

HTTP-Status: 404
Content type: application/json

```
{
  "data": {},
  "message": {
    "message": "Resource not found: '/v1/api-sync/10392k8/sync'"
  },
  "status": "fail"
}
```

17.2 Server Error: Missing Credentials

In this example: the basic auth credentials where missing.

HTTP-Status: 401
Content type: application/json

```
{
  "data": {},
  "message": {
    "message": "Invalid access token"
  },
  "status": "fail"
}
```

17.3 Server Error: Server-side Timeout

In this example: internal problems caused server to abandon the connection before the response could be sent. Request may, or may not, have been executed.

HTTP-Status: 408
Content type: application/json

```
{
  "data": {},
  "message": {
    "message": "Request timeout"
  },
  "status": "error"
}
```