

**Let's seize  
opportunities.**

**Together**

## **Manual Go! Partner SSO** **Version 1.3**

A manual for external parties integrating with OTYS  
to implement an OTYS Single Sign On functionality  
for users of the OTYS Go! system

## Version management

Below you will find an overview of the versions available for this document.

Version	Date	Author	Description
1.0	05-05-2021 - 01-06-2021	Bastiaan Brans	Created initial version of document, including four chapters: 'General' (chapter 1), 'Activation & configuration' (chapter 2) & 'Process SSO requests' (chapter 3).
1.1	12-07-2022	Bastiaan Brans	Updated 'Process SSO enabled by Key-user' (paragraph 3.2) & 'Process SSO trigger by user' (paragraph 3.3) including their code examples; to make the login process more clear.
1.2	12-02-2024	Bastiaan Brans	Added the 'sessionExpireDate' to the responses of the requests mentioned in paragraph 3.4 ('Authenticate user'), which we are now returning based upon these existing requests.
1.3	02-04-2024	Bastiaan Brans	Added new response field 'defaultLanguage' in paragraph 3.4.2 ('Authenticate session token, retrieve user data') & paragraph 3.4.3 ('Authenticate session token, retrieve client & user data').

# Contents

- Version management ..... 2
- Contents ..... 3
- 1. General ..... 4
  - 1.1 OTYS Recruiting Technology..... 4
  - 1.2 OTYS Go! Partner Framework ..... 4
  - 1.3 OTYS Go! Partner SSO ..... 5
  - 1.4 About this document..... 6
- 2. Activation & configuration ..... 7
  - 2.1 Activation by OTYS for initial client(s) ..... 7
  - 2.2 Configuration in OTYS client..... 8
  - 2.3 Triggering SSO action in OTYS client..... 8
  - 2.4 Activation by OTYS for additional clients ..... 8
- 3. Process SSO requests..... 9
  - 3.1 General structure ..... 9
  - 3.2 Process SSO enabled by Key-user..... 10
  - 3.3 Process SSO trigger by User..... 11
  - 3.4 Authenticate user ..... 13
    - 3.4.1 Authenticate session token, retrieve client data ..... 13
    - 3.4.2 Authenticate session token, retrieve user data ..... 14
    - 3.4.3 Authenticate session token, retrieve client & user data..... 16
    - 3.4.4 Error handling..... 18
  - 3.5 Retrieving current users ..... 18

# 1. General

In this chapter we will provide you some generic information about OTYS, the OTYS Go! Partner framework and OTYS Go! Partner SSO.

## 1.1 OTYS Recruiting Technology

OTYS Recruiting Technology (hereafter 'OTYS') develops, maintains and optimizes software for around 1.000 clients worldwide. The core of this software is created for recruiting purposes (agencies, corporate recruitment & job boards).

The most important elements of the OTYS solution are:

- OTYS Go!  
This is a browser based software solution in which the main users (for example recruiters) of the client work on a daily basis. Each client has a client specific URL from which they access OTYS Go! (<https://clientprefix.otysapp.com>). Users then login using either their OTYS username & password or using an external SSO option (for example Azure). After they are logged in, they will have access to the modules & functionalities based upon the permissions assigned to their account.
- OTYS website  
In most cases the client also has one or multiple websites build, hosted & maintained by OTYS linked to their system. This is normally a website which is publicly available (so that the company can for example use it to show & tell more about their company and publish their jobs), with some portals behind a login (for candidates, customers, hiring managers and/or suppliers). These 'OTYS websites' interact directly with the OTYS system. Some clients choose to let a third party build, host & maintain their website; in which case database interaction is for example done through the OTYS web services.

## 1.2 OTYS Go! Partner Framework

OTYS believes it should focus on the elements they excel in and that their focus should be on just that. Other parties are able to do other things better and OTYS believes that their clients are best helped by having a 'best of breed'-solution using partners. To support this 'best of breed'-solution for both clients & partners, OTYS created the 'OTYS Go! Partner framework' which allows partners to easily connect to the OTYS solution and provide users a nice (seamless) experiences.

The OTYS Go! Partner framework consists of the following elements:

1. OTYS Web Services (OWS)  
OTYS has very extensive web services to connect to the OTYS system. Where other similar parties have web services as an 'add on' to their own application (with normally limited capabilities), OTYS chose to create a web service layer which is an integral part of their current software solution. All actions users take through the user interface (OTYS Go!) are done by OWS request, thus allowing external parties to connect with the web services and (technically) be able to do all the things end users are able to do through the GUI.
2. OTYS Go! Partner OWS Webhooks  
If an external party connects to OTYS using OWS, there is always some kind of 'trigger' to exchange data. If this trigger should be based upon a user action in OTYS Go! (for example a user changing a status) , it is possible to create OWS webhooks. If configured, the partner can receive an 'active signal' that a user is taking the action. This way as a partner you are able to 'instantly' start a process if for example a user changes a status in the OTYS Go! system.

3. OTYS Go! Partner SSO

If an OTYS Go! user interacts with a partner functionality, it is probably a wise thing to add some kind of authentication. Larger clients will probably have an Active Directory (such as Microsoft Azure) set up to make user management & authentication work smoothly. Smaller clients will probably not have set this up, so for those we created an OTYS Go! Partner SSO functionality, which allows partners to automatically sync users with OTYS Go! users and to easily authenticate users using Single Sign On (with OTYS being the 'Active Directory').

4. OTYS Go! Partner widgets

In some cases a partner wants to be able to show data from their own system to users of OTYS Go! within the context of OTYS Go! (for example show additional information you have about a candidate within an OTYS Go! candidate detail). You might even want to add some 'interaction' in this context (for example a button to click or a form to be submitted). For these kinds of scenarios we have created OTYS Go! Partner widgets, which allows partners to provide iframe-ish content which can be loaded directly into the right context in OTYS Go!

5. OTYS Go! Partner custom actions

The above mentioned OTYS Go! Partner widgets can be of great added value if you want to show some information within the context of OTYS Go!, however if you have more complex ('full blown') interfaces or simply want to 'trigger a specific action', the widgets might not be the best choice. For these kinds of interactions we have created 'OTYS Go! Partner custom actions' which allows partners to create custom buttons which can be added on various places in the OTYS Go! system (for example in a candidate detail). Clicking this button will open a URL on your side (with for example the candidate OUID as a parameter) in the current browser tab, a new browser tab or on the background.

## 1.3 OTYS Go! Partner SSO

In OTYS Go! we have many functionalities which support our clients in their recruitment process. Since "you can't be great at everything", we also have many partner integrations for specific functionalities we do not have and do not want to build because we don't want to (it is not 'core OTYS') or other parties are simply better at it. Most of these integrations mainly consist of a 'data exchange' between OTYS & the partner. Our users will have an OTYS Go! user interface with some form of partner integration and most probably also a user interface from the partner side. Traditionally users will then also have two accounts: one for OTYS and one for the partner side.

Because our clients are using more-and-more specialized partner products, they are having more-and-more accounts. This is annoying for the end users of the client (for example needing to log 5 sets of account credentials and logging into each system every day) and annoying for the administrators of the client (needing to create all the accounts in case of a new colleague and needing to close all the accounts in case of a colleague leaving the company).

Larger OTYS clients will probably have Single Sign On of Azure implemented, with which OTYS is integrated (see Azure integration). If the partners are also integrated with Azure, the single sign on process will take place through Azure. We however also have many smaller clients which do not have Single Sign On from Azure in place. To provide them with the convenience of having one central directory with Single Sign On for OTYS and integrated partners, we have create OTYS Go! Partner SSO. This way the user can login to OTYS Go! and from OTYS Go! open a partner interface (where we send authentication data on the background, so that the user can be automatically logged in).

OTYS Go! Partner SSO consists of:

- Sending client & user data from OTYS to the partner once a Key-user decides to use a partner integration (so that the partner can create and/or connect accounts);

- Retrieving client & user data from OTYS by the partner after a Key-user has decided to use a partner integration (so that the partner can keep the accounts 'synced' and can for example close accounts on the partner side, which are blocked on the OTYS side)
- A direct link from OTYS to a URL at the partners side which posts a 'partner session token' for the user, where the partner is able to validate the session token using his 'partner API key' and see which client/user it is, so that the partner can automatically log on the user.

OTYS Go! Partner SSO is also integrated into our OTYS Go! Partner widgets & custom actions; allowing you as a partner to add building blocks and reusing already implemented code & logic.

## 1.4 About this document

This document describes the concept of OTYS Go! Partner SSO and how external parties can use them in their processes. This document will not describe the generic working of OWS and is not a document where you learn how to develop these kinds of integrations in general. It is therefore intended for external parties already connecting to OTYS using OWS and want to improve this integration by implementing webhooks.

Examples are provided in PHP. If you will be using PHP to create an endpoint to process our the SSO requests, you can copy-paste our examples and work your way from there. If you are using a different scripting language, we are quite sure that the PHP examples give you good guide marks to write appropriate code in the scripting language of your choice.

Although we do our best to help your brain convert this information into the usage of the OTYS Go! Partner SSO for the topics you want to implement it (for example by providing examples), your brain will still need to do the hard work by figuring out how your process can use this OTYS Go! Partner SSO in an optimal way.

If you have feedback about this document, we would love to hear it from us. Please send us an email at [partners@otys.com](mailto:partners@otys.com) and provide us your feedback.

## 2. Activation & configuration

In this chapter we will explain how the OTYS Go! Partner SSO can be activated and configured. Please note that after you have completed this chapter you will be able to start creating the endpoints which will handle all requests. In the next chapter we will explain how you are able to process the requests send to your endpoint.

### 2.1 Activation by OTYS for initial client(s)

To activate OTYS Go! Partner SSO for you as a partner, we will need the following information from you as a partner:

- Partner name  
The name your company should show in our user interface. Probably not your 'Official name', but the name people will recognize your company with.
- Partner description  
It would be nice to have a small description of what your company / solution does. We prefer to have some specific information of what your product actually does (for example 'We automatically send LinkedIn requests to selected candidates, based upon configurable templates'), instead of having a nice (yet less clear) marketing text like 'We connect people to make the world a better place'.
- Endpoint URL  
A generic endpoint URL which will be handling our requests. It is fine to make this a temporary endpoint during the development process and that we change it to the 'official endpoint' at a later stage.
- Logo  
Your company logo, preferably both in SVG- & PNG-format. Although we prefer to receive the SVG & PNG, please don't convert your JPG to a PNG (we can do that as well) or delay development because you are not able to find your SVG logo (we can always add better versions before actual 'go live').
- OTYS client(s) for initial activation  
OTYS Partner SSO can be enabled for specific (mutual) clients or for all OTYS clients (only if we are sure this is of added value). During the development phase you will probably only enable it for very specific clients (for example your own OTYS testing client and maybe for an actual client you want to include in the development process). Please let us know for which OTYS clients the OTYS Go! Partner SSO integration should initially be activated. Of course we will be able to add clients at a later stage upon your request.

Please send the information mentioned above to [partners@otys.com](mailto:partners@otys.com). In return we will provide you with an OTYS Go! Partner API key, which can be used when validating OTYS Go! Partner session tokens (more about that in the next chapter).

We are only able to provide this OTYS Go! Partner API key once, so please make sure it is stored in a safe place. If you at some time lost it, we are able to generate a new key, however this will cause your old key not to work anymore.

## 2.2 Configuration in OTYS client

Once OTYS Go! Partner SSO has been activated by us and you have received your OTYS Go! Partner API key, you will need to enable the SSO integration in the specific OTYS client(s).

This activation on the clients side can only be done by Key-users, by taking the following steps:

1. Open OTYS Go!
2. Login to your OTYS (Key-user) account
3. Click on your name on the top right and choose 'Client settings'
4. At 'Search setting number...' in panel on right, type in 'GE285' followed by enter
5. Open setting 'Partners – Enabled partners for client' in the list on the left
6. You will see a list of partners available for you client. Click on the button 'Enable integration' below the partner you want to enable and confirm by clicking on 'Enable'

The partner integration has now been enabled and a request has been executed to your previously provided endpoint URL to notify you the integration has been enabled for this client (more about this request and the way to handle it in the next chapter).

## 2.3 Triggering SSO action in OTYS client

After OTYS Go! Partner SSO has been enabled in a client, it is possible to easily trigger a simple SSO action in the OTYS client. At this stage your endpoint has not been coded yet to be able to handle the requests (so don't expect it to work out of the box), but it allows you to see what happens.

Triggering this simple SSO action can be done by all users, by taking the following steps:

1. Open OTYS Go!
2. Login to your OTYS account
3. Click on your name on the top right and choose 'Partner'
4. You will see a list of partners enabled for you client. Click on the button 'Login' below the partner you want to access
5. A new browser tab will open with your previously provided endpoint URL

In the process above we will post the email address of the user and a Go! Partner SSO user session token to the provided endpoint. Using this token and your Go! Partner API key, you will be able to authenticate the user (see the next chapter for all the details).

## 2.4 Activation by OTYS for additional clients

After you have created a nicely working integration based upon your OTYS testing client, we can enable the integration for additional OTYS clients upon your request. It is also possible to globally enable your integration for all OTYS clients, however this will of course need to be sure your integration will work and will be of added value for all our clients (worldwide). In 'normal cases' we will activate your integration for (a group of) specific clients.

Please note that after the integration has been activated for additional clients on our side, a Key-user of the client will still need to enabled it from their end.



## 3. Process SSO requests

In chapter 2 you found out how you can activate & configure your OTYS Go! Partner SSO integration, so we expect you now have an OTYS Go! Partner API key which is linked to the endpoint URL you have provided. In this chapter we will explain how you can process the SSO requests.

### 3.1 General structure

The OTYS Go! Partner SSO requests consist of three main components:

- SSO enabled by Key-user  
If a Key-user enables the OTYS Go! Partner SSO integration for your partner integration (see paragraph 2.2), we will trigger your endpoint and post data so that you 'know' a new client has enabled your integration. This allows you to (automate) follow up actions in case of a new client.
- SSO trigger by User  
If a User triggers an SSO action for your partner integration (see paragraph 2.3), we will open your endpoint in a new browser tab and post the users email address & OTYS Go! Partner user session token. This allows you to authenticate the user and automatically login the user in your application.
- Authenticate user  
Based upon either one of the previous triggers ('SSO enabled by Key-user' & 'SSO trigger by User'), you will want to authenticate the request. We have various ways to do so, all requiring the users OTYS Go! partner session token and your OTYS Go! Partner API key.
- Retrieving current users  
OTYS users can be 'blocked' (normally if an existing employee leaves the client). We have a method on our side in place which allows you to retrieve the active users of a client which have had interaction with your integration, which you can for example run in a nightly process to automatically block users on your end.

The process above uses the following elements for authentication:

- OTYS Go! Partner API key  
This is the API key we provided to you after activating your partner integration (see paragraph 2.1) and allows you to authenticate users or retrieve current users of clients which have your integration enabled. Please note this is a generic API key linked to you as a partner with which you take authentication actions for all clients which have your integration enabled. It is not an OTYS Web Services (API) key to take additional actions in the clients database (retrieving candidates, creating jobs, etc). If your process requires that interaction with the clients database besides user authentication, you will (still) need an OTYS Web Services (API) key per client.
- OTYS Go! Partner user session token  
This is the session token which is linked to the user triggering the SSO action and you as a partner. Once this session token is posted to your endpoint, you can validate the user by sending us a request with this session token and your OTYS Go! Partner API key. If it is a valid match, we will return some basic information about the user and/or client so that you know it is a valid request.

The exact usage will be explained in depth in the following paragraphs.

## 3.2 Process SSO enabled by Key-user

As mentioned above; if a Key-user enables the OTYS Go! Partner SSO integration for your partner integration (see paragraph 2.2), we will trigger your endpoint and post data so that you ‘know’ a new client has enabled your integration. This allows you to (automate) follow up actions in case of a new client.

In such a case we will post an array to your endpoint which (currently only) includes one object ‘integrationData’. This object includes a JSON with the email address of the user initiating the enabling of the integration (named ‘userPrimaryEmail’) and a onetime session token for the user (named ‘sessionToken’). This trigger allows you to take the desired follow up action.

To retrieve & process this integration data you can create a PHP file which looks as follows:

```
<?php
// Execute actions if posted data includes integration data
if($_POST['integrationData']) {

    // Retrieve JSON with integration data from array
    $json = $_POST['integrationData'];

    // Convert JSON to associative array
    $data = json_decode($json, true);

    // Define user primary email & session token variables
    $userPrimaryEmail = $data['userPrimaryEmail'];
    $userSessionToken = $data['sessionToken'];

    // EXECUTE FOLLOW UP ACTIONS AFTER INTEGRATION WAS ENABLED
    echo "Take follow up actions for integration enabled by user with
    email address '$userSessionToken' and user session token
    '$userSessionToken'.";

// Execute actions if posted data does not include integration data
} else {

    // EXECUTE ALTERNATIVE FOLLOW UP ACTIONS
    echo "Take follow up actions for other situations.";

}

?>
```

As you can see, the script above does not do something substantial, however it ‘recognizes’ that it was triggered by a Key-user enabling the integration and now ‘knows’ the primary email address of the Key-user and the related session token. This can be used for further follow up actions (for example validating the session token, as described later in this chapter).

Please note that once a Key-user has enabled an OTYS Go! Partner SSO integration, he can disable it. Once it is disabled he can reenable it; causing again a similar request to the endpoint (so please note your follow up script recognizes has a method of recognizing existing clients and does not always create new clients; to prevent duplicates).

### 3.3 Process SSO trigger by User

As mentioned above; if a User (which can be a Key-user or a 'normal' User) triggers an SSO action for your partner integration (see paragraph 2.3), we will open your endpoint in a new browser tab and post the users email address & OTYS Go! Partner user session token. This allows you to authenticate the user and automatically login the user in your application.

In such a case we will post an array to your endpoint which (currently only) includes one object 'loginData'. This object includes a JSON with the email address of the user initiating the login (named 'userPrimaryEmail') and a onetime session token for the user (named 'sessionToken'). This trigger allows you to take the desired follow up action.

To retrieve & process this login data you can create a PHP file which looks as follows:

```
<?php
// Execute actions if posted data includes login data
if($_POST['loginData']) {

    // Retrieve JSON with login data from array
    $json = $_POST['loginData'];

    // Convert JSON to associative array
    $data = json_decode($json, true);

    // Define user primary email & session token variables
    $userPrimaryEmail = $data['userPrimaryEmail'];
    $userSessionToken = $data['sessionToken'];

    // EXECUTE FOLLOW UP ACTIONS AFTER USER LOGS IN
    echo "Take follow up actions for login by user with email address
    '$userSessionToken' and user session token '$userSessionToken.'";

// Execute actions if posted data does not include login data
} else {

    // EXECUTE ALTERNATIVE FOLLOW UP ACTIONS
    echo "Take follow up actions for other situations.";

}

?>
```

As you can see the format is quite similar to the format of the Go! Partner SSO being enabled by the Key-user (see the previous paragraph), with the simple difference of the naming of the object.

Since both actions (processing if the integration was enabled as mentioned in the previous paragraph and a login action as mentioned in this paragraph) trigger the same endpoint, if you want to support both actions it makes sense to let your script handle both requests

To retrieve & process this integration and login data you can create a PHP file which looks as follows:

```
<?php
// Execute actions if posted data includes login data
if($_POST['loginData']) {

    // Retrieve JSON with login data from array
    $json = $_POST['loginData'];

    // Convert JSON to associative array
    $data = json_decode($json, true);

    // Define user primary email & session token variables
    $userPrimaryEmail = $data['userPrimaryEmail'];
    $userSessionToken = $data['sessionToken'];

    // EXECUTE FOLLOW UP ACTIONS AFTER USER LOGS IN
    echo "Take follow up actions for login by user with email address
    '$userSessionToken' and user session token '$userSessionToken.'";

// Execute actions if posted data includes integration data
} elseif($_POST['integrationData']) {

    // Retrieve JSON with integration data from array
    $json = $_POST['loginData'];

    // Convert JSON to associative array
    $data = json_decode($json, true);

    // Define user primary email & session token variables
    $userPrimaryEmail = $data['userPrimaryEmail'];
    $userSessionToken = $data['sessionToken'];

    // EXECUTE FOLLOW UP ACTIONS AFTER INTEGRATION WAS ENABLED
    echo "Take follow up actions for integration enabled by user with
    email address '$userSessionToken' and user session token
    '$userSessionToken.'";

// Execute actions if posted data does not include login/integration data
} else {

    // EXECUTE ALTERNATIVE FOLLOW UP ACTIONS
    echo "Take follow up actions for other situations.";

}

?>
```

Again, the script above does not do something substantial, however it 'recognizes' that it was either triggered by a User triggering the SSO login or by a Key-user enabling the integration and now 'knows' the primary email address of the Key-user and the related session token. This can again be used for further follow up actions.

### 3.4 Authenticate user

Based upon the steps above you should now have an OTYS Go! Partner user session token. It is now time to authenticate the user based upon this session token in combination with your OTYS Go! Partner API key.

We have some 'flavors' in doing so (depending whether you would like to retrieve client and/or user data); all requiring you to connect to our OWS endpoint (<https://ows.otys.nl/jservice.php>) and providing the OTYS Go! Partner user session token & your OTYS Go! Partner API key.

#### 3.4.1 Authenticate session token, retrieve client data

You can authenticate the user and retrieve data of the linked client by sending the following request to OWS, where **[partnerapikey]** is the OTYS Go! Partner API key you received from us and **[partnersessiontoken]** is the OTYS Go! Partner user session token retrieved from the posted data:

```

{
  "jsonrpc": "2.0",
  "method": "Otys.Services.SsoService.getClient",
  "params": [
    "[partnerapikey]",
    "[partnersessiontoken]"
  ],
  "id": 1
}

```

If a valid & matching OTYS Go! Partner API key and OTYS Go! Partner user session token have been provided, this will result in the following response:

```

{
  "result": {
    "Client": {
      "clientName": "[clientname]",
      "clientId": [clientid],
      "clientCode": "[clientcode]",
      "clientWebsite": "[clientwebsite]",
      "clientEmail": "[clientemail]"
    },
    "Authentication": {
      "sessionToken": "[partnersessiontoken]",
      "sessionExpireDate": "[sessionexpiredate]"
    }
  },
  "id": 1
}

```

The response consists of the following fields:

- **[clientname]**: The name of the client as it is registered in the OTYS system.
- **[clientid]**: The unique client id (something like '1234') of the client in the OTYS system.
- **[clientcode]**: The unique client code (something like 'abcdef') of the client in the OTYS system.
- **[clientwebsite]**: The website of the client as it is registered in the OTYS system.
- **[clientemail]**: The main email address of the client as it is registered in the OTYS system.
- **[partnersessiontoken]**: The OTYS Go! Partner user session token you provided in your original request.
- **[sessionexpiredate]**: The above-mentioned session token will automatically expire 24 hours after the last users partner interaction. This field will show the current session expire date.

Depending upon your use case, this way you know the logged in user is a valid user linked to the OTYS client with client ID **[clientid]**.

#### 3.4.2 Authenticate session token, retrieve user data

You can also authenticate the user and retrieve data of the linked user by sending the following request to OWS, where **[partnerapikey]** is the OTYS Go! Partner API key you received from us and **[partnersessiontoken]** is the OTYS Go! Partner user session token retrieved from the posted data:

```
{
  "jsonrpc": "2.0",
  "method": "Otys.Services.SsoService.getUser",
  "params": [
    "[partnerapikey]",
    "[partnersessiontoken]"
  ],
  "id": 1
}
```

If a valid & matching OTYS Go! Partner API key and OTYS Go! Partner user session token have been provided, this will result in the following response:

```
{
  "result": {
    "User": {
      "userId": [userid],
      "firstName": "[userfirstname]",
      "infix": "[userinfix]",
      "lastName": "[userlastname]",
      "emailPrimary": "[userprimaryemail]",
      "defaultLanguage": "[defaultlanguage]"
    },
    "Authentication": {
      "sessionToken": "[partnersessiontoken]",
      "sessionExpireDate": "[sessionexpiredate]"
    }
  },
  "id": 1
}
```

The response consists of the following fields:

- **[userid]**: The unique user id (something like '12345') of the user in the OTYS system.
- **[userfirstname]**: The first name of the user as it is registered in the OTYS system.
- **[userinfix]**: The infix of the user as it is registered in the OTYS system.
- **[userlastname]**: The last name of the user as it is registered in the OTYS system.
- **[userprimaryemail]**: The primary email address of the user as it is registered in the OTYS system.
- **[defaultlanguage]**: The users default language in ISO 639-1 format (for example 'en'), as configured in the users profile.
- **[partnersessiontoken]**: The OTYS Go! Partner user session token you provided in your original request.
- **[sessionexpiredate]**: The above-mentioned session token will automatically expire 24 hours after the last users partner interaction. This field will show the current session expire date.

Depending upon your use case, this way you know the logged in user is a valid user linked to the OTYS client with user ID **[userid]**.

### 3.4.3 Authenticate session token, retrieve client & user data

If you want to authenticate the user and retrieve data of both the linked client & user, there is no need to send the two previous request. You can retrieve them both at the same time by sending the following request to OWS, where **[partnerapikey]** is the OTYS Go! Partner API key you received from us and **[partnersessiontoken]** is the OTYS Go! Partner user session token retrieved from the posted data:

```
{
  "jsonrpc": "2.0",
  "method": "Otys.Services.SsoService.getClientAndUser",
  "params": [
    "[partnerapikey]",
    "[partnersessiontoken]"
  ],
  "id": 1
}
```

If a valid & matching OTYS Go! Partner API key and OTYS Go! Partner user session token have been provided, this will result in the following response:

```
{
  "result": {
    "Client": {
      "clientName": "[clientname]",
      "clientId": [clientid],
      "clientCode": "[clientcode]",
      "clientWebsite": "[clientwebsite]",
      "clientEmail": "[clientemail]" ,
      "defaultLanguage": "[defaultlanguage]"
    },
    "User": {
      "userId": [userid],
      "firstName": "[userfirstname]",
      "infix": "[userinfix]",
      "lastName": "[userlastname]",
      "emailPrimary": "[userprimaryemail]"
    },
    "Authentication": {
      "sessionToken": "[partnersessiontoken]",
      "sessionExpireDate": "[sessionexpiredate]"
    }
  },
  "id": 1
}
```



The response consists of the following fields:

- **[clientname]**: The name of the client as it is registered in the OTYS system.
- **[clientid]**: The unique client id (something like '1234') of the client in the OTYS system.
- **[clientcode]**: The unique client code (something like 'abcdef') of the client in the OTYS system.
- **[clientwebsite]**: The website of the client as it is registered in the OTYS system.
- **[clientemail]**: The main email address of the client as it is registered in the OTYS system.
- **[userid]**: The unique user id (something like '12345') of the user in the OTYS system.
- **[userfirstname]**: The first name of the user as it is registered in the OTYS system.
- **[userinfix]**: The infix of the user as it is registered in the OTYS system.
- **[userlastname]**: The last name of the user as it is registered in the OTYS system.
- **[userprimaryemail]**: The primary email address of the user as it is registered in the OTYS system.
- **[defaultlanguage]**: The users default language in ISO 639-1 format (for example 'en'), as configured in the users profile.
- **[partnersessiontoken]**: The OTYS Go! Partner user session token you provided in your original request.
- **[sessionexpiredate]**: The above-mentioned session token will automatically expire 24 hours after the last users partner interaction. This field will show the current session expire date.

Depending upon your use case, this way you know the logged in user is a valid user linked to the OTYS client with client ID **[clientid]** and user ID **[userid]**.

### 3.4.4 Error handling

The responses mentioned above are all based upon 'happy flows' (you providing us a valid & matching OTYS Go! Partner API key and OTYS Go! Partner user session token). Of course life is not always great, which in this case can result in some errors.

If you provide us an incorrect OTYS Go! Partner API key, this will result in the following response:

```
{
  "error": {
    "message": "Invalid API key.",
    "code": 0
  },
  "id": 1
}
```

Alternatively, if you provide us an incorrect (non-matching) OTYS Go! Partner user session token, this will result in the following response:

```
{
  "error": {
    "message": "Invalid session token.",
    "code": 0
  },
  "id": 1
}
```

Of course it is up to you to further 'handle' these errors in your script.

## 3.5 Retrieving current users

As mentioned before; OTYS users can be 'blocked' (normally if an existing employee leaves the client). We have a method on our side in place which allows you to retrieve the active users of a client which have had interaction with your integration, which you can for example run in a nightly process to automatically block users on your end.

You can retrieve the active users of a client by sending the following request to OWS, where **[partnerapikey]** is the OTYS Go! Partner API key you received from us:

```
{
  "jsonrpc": "2.0",
  "method": "Otys.Services.SsoService.getUsers",
  "params": [
    "[partnerapikey]"
  ],
  "id": 1
}
```

If a valid OTYS Go! Partner API key has been provided, this will result in the following response:

```
{
  "result": [
    {
      "clientId": "[clientid]",
      "userId": "[userid]"
    }
  ],
  "id": 1
}
```

Please note the array which includes the client ID & user ID is recursive and the response will include all (active) users of all clients which have had at least interaction (SSO trigger) with you as a partner.

The response consists of the following fields:

- **[clientid]**: The unique client id (something like '1234') of the client in the OTYS system.
- **[userid]**: The unique user id (something like '12345') of the user in the OTYS system.

Depending upon your use case, this way you know the logged in user is a valid user linked to the OTYS client with user ID **[userid]**.