

**Let's seize
opportunities.**

Together

Manual Google for Jobs function Version 1.1

A manual for external parties integrating with OTYS to implement a Google for Jobs function that will generate a Google for Jobs code for a vacancy detail.

Version management

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

Version	Date	Author	Description
1.0	29-11-2021 - 03-12-2021	Bastiaan Brans	Created initial version of document, including three chapters: 'General' (chapter 1) & 'Implementation' (chapter 2).
1.1	04-02-2022	Bastiaan Brans	Added chapter 'Code explanation' (chapter 3). Updated paragraph 'About this document' (1.5), so it reflects this new chapter. Small formatting change in paragraph 'Calling the function' (2.3). Updated Google for Jobs function (in 'OTYS Google for Jobs function.zip') so that comment on line 924 is correct (changed 'Start follow up' to 'End follow up').

Contents

Version management	2
Contents	3
1. General	4
1.1 OTYS Recruiting Technology.....	4
1.2 OTYS web services	4
1.3 Google for Jobs	4
1.4 Google for Jobs function	5
1.5 About this document.....	5
2. Implementation.....	6
2.1 Download & upload function	6
2.2 Mandatory changes.....	6
2.3 Calling the function	7
2.4 Optional changes.....	8
3. Code explanation.....	9

1. General

In this chapter we will provide you some generic information about OTYS, the OTYS web services (OWS) and the Google for Jobs function.

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 web services

OTYS has very extensive web services (called 'OTYS web services' or simply 'OWS') 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.

OWS can be used for all kinds of database interactions, including creating websites for OTYS clients that 'interact' with the OTYS website. This way an external web builder is able to for example retrieve all published or specific vacancies to include on the website.

1.3 Google for Jobs

Besides its generic internet search functionality, Google offers various specific Google search services (for example Google Shopping). Besides these search services normally having specific 'search start pages', they are also incorporated in the generic Google search. This way if you are searching for 'Xbox Series X' you will find web shops offering this product in a 'structured' way incorporated in the generic Google result (normally at the top and/or right side). Where the normal Google search result take very generic element into account (like the title, headers & other body elements of a page);

these specialized services normally need 'structured data'. This way Google 'knows' what for example the price is and what the shipping costs are for the product, so that they are able to offer more structured search results to its visitors. For this Google normally uses standards from Schema.org (<https://schema.org/>).

At some stage Google also launched a specialized search service for searching for jobs, called 'Google for Jobs'. By offering job details in a structured way (so that Google 'knows' what for example the job title and the job description is), Google will include the job in their Google for Jobs search engine (which is also incorporated in their generic search results).

The way this structured data should be added to job postings for them to be included in Google for Jobs can be found on:

<https://developers.google.com/search/docs/advanced/structured-data/job-posting>

1.4 Google for Jobs function

If an OTYS client has a website build, hosted & maintained by OTYS, we have an 'out of the box' solution that will add a Google for Jobs code to the vacancy details of the website. Since clients use the OTYS system differently, there are various ways to 'configure' the system to make sure the Google for Jobs code contains the correct information.

If an OTYS client has a website build, hosted & maintained by a third party, this third party is normally creating the vacancy detail page and should thus create code that generates the Google for Jobs code. Creating this 'hardcoded' for a very specific customer is not all that complex. However, since most web builders prefer to create code in a more generic and 'stateless' way; it is better to create code that uses the same logic OTYS does itself. This way the same piece of code will generate a different Google for Jobs code depending upon the OTYS configuration.

Since we can understand this is quite a lot of work for third parties and OTYS knows this logic best, we have created a PHP function that generates this Google for Jobs code for a specific vacancy. It uses the same logic OTYS uses and retrieves all needed logic through OWS.

We are offering this Google for Jobs function to make your life (as an OTYS partner) easier. Feel free to use & implement it 'as is'; feel free to use it as a base after which you make adjustments you find fit or feel free not to use it at all. Please note the function is only available as a PHP function. If you are using a different scripting language you will probably be able to use it as an 'example' to understand the OTYS logic to create a similar function/script in the scripting language you use.

1.5 About this document

This document describes the concept of the Google for Jobs function and how external parties can use them in their website. 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 easily add a Google for Jobs code in their vacancy details. The basic implementation of the function (chapter 2) is quite simple & straight forward. We also offer you a line-by-line code explanation (which takes up much more text & time to read); however please note this is intended for reference purposes (for example if you do not understand a piece of the code) and possibly / probably even not needed.

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

2. Implementation

In this chapter we will explain how the Google for Jobs function can be implemented in your own code, so that you are able to call it & generate a Google for Jobs code in a vacancy detail.

2.1 Download & upload function

You can download the function from the following URL:

https://www.otys.com/partners/downloads/OTYS_Google_for_Jobs_function.zip

If you unpack the ZIP-file, you will see a main folder 'OTYS Google for Jobs function', a sub folder in format 'yyyymmdd' (allowing us to add multiple versions in the ZIP file if we do updates in the future) and within that sub folder the function itself (googleForJobs.inc.php) and a test file to call the function (googleForJobsTest.php):

- OTYS Google for Jobs function
 - yyyymmdd
 - googleForJobs.inc.php
 - googleForJobsTest.php

You should of course upload function and (if desired) the test file to an appropriate location on your server so that you are able to include it in your current script(s).

2.2 Mandatory changes

Although you are free to change the function as much as you like, there are just a few mandatory changes to make sure it works 'out of the box'.

Since the function relies on OWS to create the Google for Jobs code, the script does need either an active OWS session or an OWS endpoint & API key (so that the script can create the session itself).

Use active session

If the function should use an active OWS session (for example because you include it on a place where there is already a connection with OWS), it is expecting the variable **\$sessionToken** with the session token. If your current logic is using a different variable for this session token, it is probably easiest to search & replace '\$sessionToken' by the variable you are using in 'googleForJobs.inc.php'.

Create session

If the function should create an OWS session token itself, it is expecting variables **\$owsEndpoint** & **\$owsApiKey**. Normally these variables are declared in a separate file, so for this purpose we added a placeholder piece of code in googleForJobs.inc.php to include this file:

```
// Include: Include OWS credentials ($owsEndpoint, $owsApiKey)
include("OWS_CREDENTIALS_FILE_PATH");
```

By simply changing '**OWS_CREDENTIALS_FILE_PATH**' to the file path where the OWS end point & API key can be found, you are set to go. If your current logic is using a different variable for this session token, it is probably easiest to search & replace '\$owsEndpoint' & '\$owsApiKey' by the variables you are using in 'googleForJobs.inc.php'.

2.3 Calling the function

The function can be called quite easily. Most simple way to test this, is by using the test file googleForJobsTest.php which is included in the ZIP file and looks as follows:

```
<?php

// Include: Google for Jobs function
include("GOOGLE_FOR_JOBS_FUNCTION_FILE_PATH");

// General: Declare script input variables
$vacancyOuid = 'VACANCY_OUID';
$vacancyLanguage = 'VACANCY_LANGUAGE';
$gfjHiringOrganizationLogo = 'GFJ_HIRING_ORGANIZATION_LOGO_URL';
$websiteName = 'WEBSITE_NAME';
$websiteUrl = 'WEBSITE_URL';
$createOwsSession = 'CREATE_OWS_SESSION';

// General: Call Google for Jobs function and store in 'gfjSnippet'-
variable
$gfjSnippet = createGfjCode($vacancyOuid, $vacancyLanguage,
$gfjHiringOrganizationLogo, $websiteName, $websiteUrl, $createOwsSession);

// General: Echo 'gfjSnippet'-variable
echo $gfjSnippet;

?>
```

You will need to define fill the fields for the section of 'input variables' to get a correct Google for Jobs code returned:

- **GOOGLE FOR JOBS FUNCTION FILE PATH**
The file path where the function (googleForJobs.inc.php) can be found.
- **VACANCY OUID**
The vacancy OUID (OTYS Unique ID) for which you want to retrieve the Google for Jobs code.
- **VACANCY LANGUAGE**
The language version of the vacancy (in ISO 639-1 format) for which you want to retrieve the Google for Jobs code (for example 'en', 'de', 'fr' or 'nl').
- **GFJ HIRING ORGANIZATION LOGO URL**
It is possible (and wise) to include a logo URL of the hiring organization. Since the website is hosted by you as a partner and not by OTYS, we will need you to provide a URL on your website where this logo can be publicly retrieved. Of course it is possible to retrieve the logo from OWS, however Google needs your public URL. If you do not want the Google for Jobs code to include a URL you can pass parameter 'false'.
- **WEBSITE NAME**
The name of the clients website.
- **WEBSITE URL**
The URL of the clients website.
- **CREATE OWS SESSION**
As mentioned in paragraph 2.2 the function can create & close a session itself or rely on an existing active session. If you want the function to create & close the OWS session pass parameter 'true', if you want the function to use an existing session pass parameter 'false'.

2.4 Optional changes

Besides the mandatory changes as mentioned in paragraph 2.2, you can of course make various optional changes. Which changes you would like to make is of course up to you, however below some 'usual suspects':

- The script is currently created as a function which can be called from a different script. If you prefer to incorporate the script in the script that generates the vacancy detail (without it being in a function), feel free to extract the code from the function and incorporate it in your script that generates a vacancy detail.
- The script currently retrieves all vacancy data to generate the Google for Jobs code. To display your vacancy detail, you will for sure already retrieve some of those fields (and maybe even all). So to limit your OWS requests you might want to combine these pieces of code.
- The script uses a structured way for variable names based upon OWS & Google for Jobs data structure. This can result in some (at first sight) 'strange' variable names (for example `$gfjFieldBaseSalaryValueValue`). Feel free to change variable names based upon your own naming conventions (as long as you change them on all instances of course).

Of course you are also free to make other changes. However, keep in mind that the script is created as a 'generic script' which can be used for all OTYS clients. It therefore uses more code & requests to for example retrieve mappings & settings for the specific client. You can make it a lot more 'simple' by removing this logic, but the script will then also be less versatile.

If in the future we will make updates to the script, we will include older versions in the download package as well. This will help you to pinpoint changes and ease up the process of making improvements in custom scripts.

3. Code explanation

In this chapter we will offer you a line-by-line code explanation. Please note this is intended for reference purposes (for example if you do not understand a piece of the code) and not intended for you to ‘fully read’.

```
// General: Start function create Google for Jobs code
function createGfjCode($vacancyOuid, $vacancyLanguage,
$gfjHiringOrganizationLogo, $websiteName, $websiteUrl, $createOwsSession) {
    global $owsEndpoint, $owsApiKey, $sessionToken, $request, $response;
```

The code above will create the function including the arguments the function needs (‘\$vacancyOuid’, ‘\$vacancyLanguage’, ‘\$gfjHiringOrganizationLogo’, ‘\$websiteName’, ‘\$websiteUrl’ & ‘\$createOwsSession’) and declare the global variables. For more information about these arguments & variables, see paragraph 2.2 (‘Mandatory changes’) & paragraph 2.3 (‘Calling the function’).

```
// Include: Include OWS credentials ($owsEndpoint, $owsApiKey)
include("OWS_CREDENTIALS_FILE_PATH");
```

The code above will include the OWS credentials. For more information about this include, see paragraph 2.2 (‘Mandatory changes’).

```
// Check input data: Die script & show error in case of incorrect OWS API key format
if(!preg_match("/^[a-zA-Z0-9+=\|/]{56}$/", $owsApiKey)) {
    die("<!-- Error #1. Incorrect OWS API key format. Not able to generate Google for Jobs code. -->");
}

// Check input data: Die script & show error in case of incorrect vacancy OUID format
if(!preg_match("/^[a-zA-Z0-9_-]{16}$/", $vacancyOuid)) {
    die("<!-- Error #2. Incorrect vacancy OUID format. Not able to generate Google for Jobs code. -->");
}

// Check input data: Die script & show error in case of incorrect vacancy language format
if(!preg_match("/^[a-z]{2}$/", $vacancyLanguage)) {
    die("<!-- Error #3. Incorrect vacancy language format '$vacancyLanguage'. Not able to generate Google for Jobs code. -->");
}
```

The code above will check the formatting of variables \$owsApiKey, \$vacancyOuid & \$vacancyLanguage based upon regular expressions and die the script with a proper error message, if the variable does not meet the regular expression. This avoids the rest of the script being executed with variables that are ‘bound to fail’.

```
// Function: Send OWS request
function sendOwsRequest() {
    global $request, $owsEndpoint, $response;
    $requestJson = json_encode($request);
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $owsEndpoint);
    curl_setopt($curl, CURLOPT_HTTPHEADER, array('Content-Type:
application/json'));
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $requestJson);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
    $responseJson = curl_exec($curl);
    $response = json_decode($responseJson, true);
    curl_close($curl);
}
```

The code above will create a 'function in the function' to send an OWS request (since this will be done multiple times in the script).

```
// Function: End OWS session
function endOwsSession() {
    global $sessionToken, $request, $owsEndpoint, $response;
    $request = [
        'jsonrpc'=>'2.0',
        'method'=>'logout',
        'params'=>[
            $sessionToken
        ],
        'id'=>1,
    ];
    sendOwsRequest();
}
```

The code above will create a 'function in the function' to end an OWS session (since this will be done multiple times in the script).

```
// Retrieve session token: Start follow up actions if OWS session needs to
be created
if($createOwsSession == true) {

    // Retrieve session token: Create OWS request
    $request = [
        'jsonrpc'=>'2.0',
        'method'=>'loginByUid',
        'params'=>[
            $owsApiKey
        ],
        'id'=>1,
    ];

    // Retrieve session token: Send OWS request
    sendOwsRequest();
}
```

```

// Retrieve session token: Die script & show error in case of invalid
Ows API key
if($response['result'] == false) {
    die("<!-- Error #4. Invalid Ows API key. Not able to generate
Google for Jobs code. -->");
}

// Retrieve session token: Declare variables
$sessionToken = $response['result'];

// Retrieve session token: End follow up actions if Ows session needs to be
created
}

```

The code above will check if the executed function has argument \$createOwsSession set to true; in which case it will use the Ows API key to create an Ows session token. If it fails it dies the script with a proper error. If it succeeds it stores the session token in a variable.

```

// Retrieve client data: Create Ows request
$request = [
    'jsonrpc'=>'2.0',
    'method'=>'check',
    'params'=>[
        $sessionToken
    ],
    'id'=>1,
];

// Retrieve client data: Send Ows request
sendOwsRequest();

// Retrieve client data: Die script & show error in case of invalid session
if($response['result'] == false) {
    die("<!-- Error #5. Invalid session. Not able to generate Google for
Jobs code. -->");
}

// Retrieve client data: Declare variables
$clientId = $response['result']['clientId'];
$clientName = $response['result']['client'];

```

The code above will create & send an Ows request to retrieve the client ID & client name. If it fails it dies the script with a proper error. If it succeeds it stores the client ID & client name in variables.

```

// Retrieve Google for Jobs main mapping: Create Ows request
$request = [
    'jsonrpc'=>'2.0',
    'method'=>'Otys.Services.GoogleForJobsService.getDataStructure',
    'params'=>[
        $sessionToken,
        $clientId
    ],
    'id'=>1,
];

```

```

// Retrieve Google for Jobs main mapping: Send OWS request
sendOwsRequest();

// Retrieve Google for Jobs main mapping: Die script & show error in case
of no permission for main mapping
if($response['error']) {
    if($createOwsSession == true) {
        endOwsSession();
    }
    die ("<!-- Error #6. No permissions for main mapping. Not able to
generate Google for Jobs code. -->");
}

// Retrieve Google for Jobs main mapping: Declare variables
$gfjEnabled = $response['result']['enable_google_for_jobs'];
$gfjFieldTitle = $response['result']['savedMappings']['title'];
$gfjFieldDescription = $response['result']['savedMappings']['description'];
$gfjFieldIdentifierName =
$response['result']['savedMappings']['identifier__name'];
$gfjFieldIdentifierValue =
$response['result']['savedMappings']['identifier__value'];
$gfjFieldDatePosted = $response['result']['savedMappings']['datePosted'];
$gfjFieldValidThrough =
$response['result']['savedMappings']['validThrough'];
$gfjFieldEmploymentType =
$response['result']['savedMappings']['employmentType'];
$gfjFieldHiringOrganizationName =
$response['result']['savedMappings']['hiringOrganization__name'];
$gfjFieldHiringOrganizationSameAs =
$response['result']['savedMappings']['hiringOrganization__sameAs'];
$gfjFieldCustomerLogo =
$response['result']['savedMappings']['hiringOrganization__logo'];
$gfjFieldStreetAddress =
$response['result']['savedMappings']['jobLocation__address__streetAddress
'];
$gfjFieldAddressLocality =
$response['result']['savedMappings']['jobLocation__address__addressLocali
ty'];
$gfjFieldAddressRegion =
$response['result']['savedMappings']['jobLocation__address__addressRegion
'];
$gfjFieldPostalCode =
$response['result']['savedMappings']['jobLocation__address__postalCode'];
$gfjFieldAddressCountry =
$response['result']['savedMappings']['jobLocation__address__addressCount
ry'];
$gfjFieldBaseSalaryCurrency =
$response['result']['savedMappings']['baseSalary__currency'];
$gfjFieldBaseSalaryValueValue =
$response['result']['savedMappings']['baseSalary__value__value'];
$gfjFieldBaseSalaryValueUnitText =
$response['result']['savedMappings']['baseSalary__value__unitText'];

```

The code above will create & send an OWS request to retrieve the Google for Maps mapping of clients. Key-users of the OTYS client are able to configure this mapping in OTYS Go! by opening client setting 'Google for Jobs – Mapping' (GE214). This way they are able to for example configure which OTYS vacancy date field should be used for the Google for Jobs 'datePosted'-field. If it fails (normally if the user does not have the permissions to execute this operation) it dies the script with a proper error. If it succeeds it stores the mapped fields in variables.

```

// Retrieve Google for Jobs employment type mapping: Create OWS request
$request = [
    'jsonrpc'=>'2.0',
    'method'=>'Otys.Services.GoogleForJobsService.getEmploymentTypeMappings',
    'params'=>[
        $sessionToken,
        $clientId
    ],
    'id'=>1,
];

// Retrieve Google for Jobs employment type mapping: Send OWS request
sendOwsRequest();

// Retrieve Google for Jobs sub mapping: Die script & show error in case of no permission for sub mapping
if($response['error']) {
    if($createOwsSession == true) {
        endOwsSession();
    }
    die ("<!-- Error #7. No permissions for sub mapping. Not able to generate Google for Jobs code. -->");
}

// Retrieve Google for Jobs employment type mapping: Declare variables
$gfjFieldEmploymentTypes = array();
foreach($response['result']['mappedValues'] as $key => $value) {
    array_push($gfjFieldEmploymentTypes, array(
        'valueOtys' => $key,
        'ValueGfj' => $value
    ));
}

```

The code above will create & send an OWS request to retrieve the Google for Maps sub mapping for employment types of clients. Key-users of the OTYS client are able to configure this mapping in OTYS Go! by opening client setting 'Google for Jobs – Mapping' (GE214) and clicking on the button 'Map employment types'. This way they are able to for example configure which OTYS option value should be used for the Google for Jobs 'FULL_TIME'-option. If it fails (normally if the user does not have the permissions to execute this operation) it dies the script with a proper error. If it succeeds it stores the mapped fields in variables.

```

// Create Google for Jobs code: Start follow up actions if Google for Jobs is enabled
if($gfjEnabled == true) {

```

The code above will check if (based upon the retrieved mapping) Google for Jobs is enabled for the client; in which case it will start executing follow up actions.

```

// Set content language: Create OWS request
$request = [
    'jsonrpc'=>'2.0',
    'method'=>'Otys.Services.VacancyService.setLanguage',
    'params'=>[
        $sessionToken,
        $vacancyLanguage
    ],
    'id'=>1,
];

// Set content language: Send OWS request
sendOwsRequest();

```

The code above will create & send an OWS request to change the content language based upon the function argument '\$vacancyLanguage'. The OTYS system supports multilingual vacancy content (so one vacancy can be published in both English, German, French & Dutch) and this way we will make sure follow up requests will return data in the correct content language.

```

// Retrieve vacancy data: Create OWS request
$request = [
    'jsonrpc'=>'2.0',
    'method'=>'Otys.Services.VacancyService.getDetail',
    'params'=>[
        $sessionToken,
        $vacancyOuid,
        array(
            "entryDateTime"=>1,
            "externalReferenceNr"=>1,
            "location"=>1,
            "locationAddress"=>1,
            "locationCity"=>1,
            "locationCountryCode"=>1,
            "locationPostcode"=>1,
            "locationState"=>1,
            "matchCriteria_1"=>1,
            "matchCriteria_2"=>1,
            "matchCriteria_3"=>1,
            "matchCriteria_4"=>1,
            "matchCriteria_5"=>1,
            "matchCriteria_6"=>1,
            "matchCriteria_7"=>1,
            "matchCriteria_8"=>1,
            "matchCriteria_9"=>1,
            "matchCriteria_10"=>1,
            "matchCriteria_11"=>1,
            "matchCriteria_12"=>1,
            "matchCriteria_13"=>1,
            "matchCriteria_14"=>1,
            "matchCriteria_15"=>1,
            "matchCriteria_16"=>1,
            "matchCriteria_17"=>1,
            "matchCriteria_18"=>1,
            "publicationEndDate"=>1,
            "publicationFirstDate"=>1,
            "publicationStartDate"=>1,
            "referenceNr"=>1,

```

```
"relation"=>1,  
"relationId"=>1,  
"salaryCurrency"=>1,  
"salaryUnit"=>1,  
"salaryValue"=>1,  
"textField_companyCulture"=>array(  
  "title"=>1,  
  "text"=>1,  
)  
,  
"textField_companyProfile"=>array(  
  "title"=>1,  
  "text"=>1,  
)  
,  
"textField_description"=>array(  
  "title"=>1,  
  "text"=>1,  
)  
,  
"textField_extra1"=>array(  
  "title"=>1,  
  "text"=>1,  
)  
,  
"textField_extra2"=>array(  
  "title"=>1,  
  "text"=>1,  
)  
,  
"textField_extra3"=>array(  
  "title"=>1,  
  "text"=>1,  
)  
,  
"textField_extra4"=>array(  
  "title"=>1,  
  "text"=>1,  
)  
,  
"textField_extra5"=>array(  
  "title"=>1,  
  "text"=>1,  
)  
,  
"textField_extra6"=>array(  
  "title"=>1,  
  "text"=>1,  
)  
,  
"textField_extra7"=>array(  
  "title"=>1,  
  "text"=>1,  
)  
,  
"textField_extra8"=>array(  
  "title"=>1,  
  "text"=>1,  
)  
,  
"textField_extra9"=>array(  
  "title"=>1,  
  "text"=>1,  
)  
,  
"textField_extra10"=>array(  
  "title"=>1,  
  "text"=>1,  
)  
,  
"textField_requirements"=>array(  
  "title"=>1,  
  "text"=>1,  
)  
,
```

```

        "textField_salary"=>array(
            "title"=>1,
            "text"=>1,
        ),
        "textField_summary"=>array(
            "title"=>1,
            "text"=>1,
        ),
        "title"=>1,
    ),
],
'id'=>1,
];

// Retrieve vacancy data: Send OWS request
sendOwsRequest();

// Retrieve session token: Die script & show error in case of incorrect
vacancy OUID
if($response['result'] == false) {
    if($createOwsSession == true) {
        endOwsSession();
    }
    die ("<!-- Error #8. Incorrect vacancy OUID. Not able to generate
Google for Jobs code. -->");
}

// Retrieve vacancy data: Declare variables
$vacancyEntryDateTime = $response['result']['entryDateTime'];
$vacancyExternalReferenceNr = $response['result']['externalReferenceNr'];
$vacancyLocation = $response['result']['location'];
$vacancyLocationAddress = $response['result']['locationAddress'];
$vacancyLocationCity = $response['result']['locationCity'];
$vacancyLocationCountryCode = $response['result']['locationCountryCode'];
$vacancyLocationPostcode = $response['result']['locationPostcode'];
$vacancyLocationState = $response['result']['locationState'];
$vacancyMatchCriteria1 = $response['result']['matchCriteria_1'];
$vacancyMatchCriteria2 = $response['result']['matchCriteria_2'];
$vacancyMatchCriteria3 = $response['result']['matchCriteria_3'];
$vacancyMatchCriteria4 = $response['result']['matchCriteria_4'];
$vacancyMatchCriteria5 = $response['result']['matchCriteria_5'];
$vacancyMatchCriteria6 = $response['result']['matchCriteria_6'];
$vacancyMatchCriteria7 = $response['result']['matchCriteria_7'];
$vacancyMatchCriteria8 = $response['result']['matchCriteria_8'];
$vacancyMatchCriteria9 = $response['result']['matchCriteria_9'];
$vacancyMatchCriteria10 = $response['result']['matchCriteria_10'];
$vacancyMatchCriteria11 = $response['result']['matchCriteria_11'];
$vacancyMatchCriteria12 = $response['result']['matchCriteria_12'];
$vacancyMatchCriteria13 = $response['result']['matchCriteria_13'];
$vacancyMatchCriteria14 = $response['result']['matchCriteria_14'];
$vacancyMatchCriteria15 = $response['result']['matchCriteria_15'];
$vacancyMatchCriteria16 = $response['result']['matchCriteria_16'];
$vacancyMatchCriteria17 = $response['result']['matchCriteria_17'];
$vacancyMatchCriteria18 = $response['result']['matchCriteria_18'];
$vacancyPublicationEndDate = $response['result']['publicationEndDate'];
$vacancyPublicationFirstDate = $response['result']['publicationFirstDate'];
$vacancyPublicationStartDate = $response['result']['publicationStartDate'];
$vacancyRefernceNr = $response['result']['referenceNr'];
$vacancyRelation = $response['result']['relation'];
$vacancyRelationId = $response['result']['relationId'];
$vacancySalaryCurrency = $response['result']['salaryCurrency'];

```



```
$vacancySalaryUnit = $response['result']['salaryUnit'];  
$vacancySalaryValue = $response['result']['salaryValue'];  
$vacancyTextFieldCompanyCultureTitle =  
$response['result']['textField_companyCulture']['title'];  
$vacancyTextFieldCompanyCultureText =  
$response['result']['textField_companyCulture']['text'];  
$vacancyTextFieldCompanyProfileTitle =  
$response['result']['textField_companyProfile']['title'];  
$vacancyTextFieldCompanyProfileText =  
$response['result']['textField_companyProfile']['text'];  
$vacancyTextFieldDescriptionTitle =  
$response['result']['textField_description']['title'];  
$vacancyTextFieldDescriptionText =  
$response['result']['textField_description']['text'];  
$vacancyTextFieldExtra1Title =  
$response['result']['textField_extra1']['title'];  
$vacancyTextFieldExtra1Text =  
$response['result']['textField_extra1']['text'];  
$vacancyTextFieldExtra2Title =  
$response['result']['textField_extra2']['title'];  
$vacancyTextFieldExtra2Text =  
$response['result']['textField_extra2']['text'];  
$vacancyTextFieldExtra3Title =  
$response['result']['textField_extra3']['title'];  
$vacancyTextFieldExtra3Text =  
$response['result']['textField_extra3']['text'];  
$vacancyTextFieldExtra4Title =  
$response['result']['textField_extra4']['title'];  
$vacancyTextFieldExtra4Text =  
$response['result']['textField_extra4']['text'];  
$vacancyTextFieldExtra5Title =  
$response['result']['textField_extra5']['title'];  
$vacancyTextFieldExtra5Text =  
$response['result']['textField_extra5']['text'];  
$vacancyTextFieldExtra6Title =  
$response['result']['textField_extra6']['title'];  
$vacancyTextFieldExtra6Text =  
$response['result']['textField_extra6']['text'];  
$vacancyTextFieldExtra7Title =  
$response['result']['textField_extra7']['title'];  
$vacancyTextFieldExtra7Text =  
$response['result']['textField_extra7']['text'];  
$vacancyTextFieldExtra8Title =  
$response['result']['textField_extra8']['title'];  
$vacancyTextFieldExtra8Text =  
$response['result']['textField_extra8']['text'];  
$vacancyTextFieldExtra9Title =  
$response['result']['textField_extra9']['title'];  
$vacancyTextFieldExtra9Text =  
$response['result']['textField_extra9']['text'];  
$vacancyTextFieldExtra10Title =  
$response['result']['textField_extra10']['title'];  
$vacancyTextFieldExtra10Text =  
$response['result']['textField_extra10']['text'];  
$vacancyTextFieldRequirementsTitle =  
$response['result']['textField_requirements']['title'];  
$vacancyTextFieldRequirementsText =  
$response['result']['textField_requirements']['text'];  
$vacancyTextFieldSalaryTitle =  
$response['result']['textField_salary']['title'];
```

```

$vacancyTextFieldSalaryText =
$response['result']['textField_salary']['text'];
$vacancyTextFieldSummaryTitle =
$response['result']['textField_summary']['title'];
$vacancyTextFieldSummaryText =
$response['result']['textField_summary']['text'];
$vacancyTitle = $response['result']['title'];

```

The code above will create & send an OWS request to retrieve the information of the vacancy for which the Google for Jobs-function is being executed. Please note it is currently retrieving all possibly needed fields (regardless of whether it is used based upon the mapping or not) and that this piece of can be optimized (but it currently ‘works’ and is not causing any issues). If it fails (normally if the user does not have the permissions to retrieve the information from this vacancy) it dies the script with a proper error. If it succeeds it stores the vacancy fields in variables.

```

// Retrieve vacancy text field data: Create OWS request
$request = [
    'jsonrpc'=>'2.0',
    'method'=>'Otys.Services.VacancyTextFieldService.getList',
    'params'=>[
        $sessionToken,
    ],
    'id'=>1,
];

// Retrieve vacancy text field data: Send OWS request
sendOwsRequest();

// Retrieve vacancy text field data: Die script & show error in case of no
permissions for retrieving vacancy texts
if($response['error']) {
    if($createOwsSession == true) {
        endOwsSession();
    }
    die ("<!-- Error #9. No permissions for retrieving vacancy texts. Not
able to generate Google for Jobs code. -->");
}

// Retrieve vacancy text field data: Create array
$vacancyTextFields = array();
foreach($response['result'] as $vacancyTextField) {
    $vacancyTextFieldRank = $vacancyTextField['rank'];
    $vacancyTextFieldField = $vacancyTextField['field'];
    $vacancyTextFieldPublished = $vacancyTextField['published'];
    array_push($vacancyTextFields, array(
        'rank' => $vacancyTextFieldRank,
        'field' => $vacancyTextFieldField,
        'published' => $vacancyTextFieldPublished
    ));
}

```

The code above will create & send an OWS request to retrieve the configuration of the vacancy texts for the client. Each client can have 5 standard vacancy texts (‘Company profile’, ‘Company culture’, ‘Job description’, ‘Job requirements’ & ‘Benefits’) and 10 additional vacancy texts (‘Vacancy extra info #1’, ‘Vacancy extra info #2’, etc). A Key-user is however able to configure which of these fields should be published on the website (they can for example use ‘Vacancy extra info #5’, but configured

it that it should NOT be shown on the website in the vacancy detail) and they are also able to order these fields (for example place 'Company profile' above or below 'Job description'). If it fails (normally if the user does not have the permissions to retrieve this configuration) it dies the script with a proper error. If it succeeds it stores the configuration in an array.

```
// Retrieve vacancy text field data: Create all texts variable
foreach($vacancyTextFields as $vacancyTextField) {
    if($vacancyTextField['field'] == 'bedrcultuur') {
        if($vacancyTextField['published'] == true &&
$vacancyTextFieldCompanyCultureText) {
            $vacancyAllTexts .=
"<h2>$vacancyTextFieldCompanyCultureTitle</h2>$vacancyTextFieldCompanyCultu
reText";
        }
    } elseif($vacancyTextField['field'] == 'bedrprofiel') {
        if($vacancyTextField['published'] == true &&
$vacancyTextFieldCompanyProfileText) {
            $vacancyAllTexts .=
"<h2>$vacancyTextFieldCompanyProfileTitle</h2>$vacancyTextFieldCompanyProfi
leText";
        }
    } elseif($vacancyTextField['field'] == 'chapo') {
        if($vacancyTextField['published'] == true &&
$vacancyTextFieldDescriptionText) {
            $vacancyAllTexts .=
"<h2>$vacancyTextFieldDescriptionTitle</h2>$vacancyTextFieldDescriptionText
";
        }
    } elseif($vacancyTextField['field'] == 'extra_fld_01') {
        if($vacancyTextField['published'] == true &&
$vacancyTextFieldExtra1Text) {
            $vacancyAllTexts .=
"<h2>$vacancyTextFieldExtra1Title</h2>$vacancyTextFieldExtra1Text";
        }
    } elseif($vacancyTextField['field'] == 'extra_fld_02') {
        if($vacancyTextField['published'] == true &&
$vacancyTextFieldExtra2Text) {
            $vacancyAllTexts .=
"<h2>$vacancyTextFieldExtra2Title</h2>$vacancyTextFieldExtra2Text";
        }
    } elseif($vacancyTextField['field'] == 'extra_fld_03') {
        if($vacancyTextField['published'] == true &&
$vacancyTextFieldExtra3Text) {
            $vacancyAllTexts .=
"<h2>$vacancyTextFieldExtra3Title</h2>$vacancyTextFieldExtra3Text";
        }
    } elseif($vacancyTextField['field'] == 'extra_fld_04') {
        if($vacancyTextField['published'] == true &&
$vacancyTextFieldExtra4Text) {
            $vacancyAllTexts .=
"<h2>$vacancyTextFieldExtra4Title</h2>$vacancyTextFieldExtra4Text";
        }
    } elseif($vacancyTextField['field'] == 'extra_fld_05') {
        if($vacancyTextField['published'] == true &&
$vacancyTextFieldExtra5Text) {
            $vacancyAllTexts .=
"<h2>$vacancyTextFieldExtra5Title</h2>$vacancyTextFieldExtra5Text";
        }
    } elseif($vacancyTextField['field'] == 'extra_fld_06') {
```

```

        if($vacancyTextField['published'] == true &&
$vacancyTextFieldExtra6Text) {
            $vacancyAllTexts .=
"<h2>$vacancyTextFieldExtra6Title</h2>$vacancyTextFieldExtra6Text";
        }
    } elseif($vacancyTextField['field'] == 'extra_fld_07') {
        if($vacancyTextField['published'] == true &&
$vacancyTextFieldExtra7Text) {
            $vacancyAllTexts .=
"<h2>$vacancyTextFieldExtra7Title</h2>$vacancyTextFieldExtra7Text";
        }
    } elseif($vacancyTextField['field'] == 'extra_fld_08') {
        if($vacancyTextField['published'] == true &&
$vacancyTextFieldExtra8Text) {
            $vacancyAllTexts .=
"<h2>$vacancyTextFieldExtra8Title</h2>$vacancyTextFieldExtra8Text";
        }
    } elseif($vacancyTextField['field'] == 'extra_fld_09') {
        if($vacancyTextField['published'] == true &&
$vacancyTextFieldExtra9Text) {
            $vacancyAllTexts .=
"<h2>$vacancyTextFieldExtra9Title</h2>$vacancyTextFieldExtra9Text";
        }
    } elseif($vacancyTextField['field'] == 'extra_fld_10') {
        if($vacancyTextField['published'] == true &&
$vacancyTextFieldExtra10Text) {
            $vacancyAllTexts .=
"<h2>$vacancyTextFieldExtra10Title</h2>$vacancyTextFieldExtra10Text";
        }
    } elseif($vacancyTextField['field'] == 'functiee') {
        if($vacancyTextField['published'] == true &&
$vacancyTextFieldRequirementsText) {
            $vacancyAllTexts .=
"<h2>$vacancyTextFieldRequirementsTitle</h2>$vacancyTextFieldRequirementsTe
xt";
        }
    } elseif($vacancyTextField['field'] == 'functieo') {
        if($vacancyTextField['published'] == true &&
$vacancyTextFieldSalaryText) {
            $vacancyAllTexts .=
"<h2>$vacancyTextFieldSalaryTitle</h2>$vacancyTextFieldSalaryText";
        }
    } elseif($vacancyTextField['field'] == 'sal_o') {
        if($vacancyTextField['published'] == true &&
$vacancyTextFieldSummaryText) {
            $vacancyAllTexts .=
"<h2>$vacancyTextFieldSummaryTitle</h2>$vacancyTextFieldSummaryText";
        }
    }
}
$vacancyAllTexts = str_replace(array("\r", "\n"), '', $vacancyAllTexts);

```

The code above will use the retrieved text fields of the vacancy and based upon the just before created configuration array create a variable that shows all applicable vacancy texts (where the fields are filled and 'published' according to the configuration) in the correct order (according to the configuration). This piece of can also be optimized (but it again currently 'works' and is not causing any issues).

```

// Retrieve relation data: Start follow up actions if vacancy is linked to
relation
if($vacancyRelationId) {

    // Retrieve relation data: Create OWS request
    $request = [
        'jsonrpc'=>'2.0',
        'method'=>'Otys.Services.RelationService.getDetail',
        'params'=>[
            $sessionToken,
            $vacancyRelationId,
            array(
                "website"=>1,
            ),
        ],
        'id'=>1,
    ];

    // Retrieve relation data: Send OWS request
    sendOwsRequest();

    // Retrieve relation data: Declare variables
    $relationWebsite = $response['result']['website'];

// Retrieve relation data: End follow up actions if vacancy is linked to
relation
}

```

The code above will check if the vacancy is linked to a 'relation' in the OTYS system and -if so- create & send a request to retrieve the relation website and store it in a variable.

```

// Retrieve directApply property setting: Create OWS request
$request = [
    'jsonrpc'=>'2.0',
    'method'=>'Otys.Services.CsmService.getValue',
    'params'=>[
        $sessionToken,
        "SE3321",
        $clientId,
        0,
        0
    ],
    'id'=>1,
];

// Retrieve directApply property setting: Send OWS request
sendOwsRequest();

// Retrieve directApply property setting: Die script & show error in case
of no permissions for retrieving directApply property setting
if($response['error']) {
    if($createOwsSession == true) {
        endOwsSession();
    }
    die ("<!-- Error #10. No permissions for retrieving directApply
property setting. Not able to generate Google for Jobs code. -->");
}

```

```
// Retrieve relation data: Declare variables
if($response['result']['value'] == 1) {
    $gfjDirectApply = 'true';
} elseif($response['result']['value'] == 0) {
    $gfjDirectApply = 'false';
}
```

The code above will create & send an OWS request to check if the 'directApply'-setting for the client is enabled or disabled and store it in a variable. Key-users of the OTYS client are able to configure this mapping in OTYS Go! by opening client setting 'Google for Jobs – Add 'directApply' property to structured data' (SE3321). It is intended to indicate that it is 'easy' to apply on the vacancy on the website (by showing a form on the vacancy detail page or after a single click of -for example- an 'Apply'-button). This way Google for Jobs will promote this user friendly behavior and 'punish' horrible application processes where you first need to create an account and can only after that registration fill in the application form.

```
// Create Google for Jobs code: Set Google for Jobs field 'datePosted'
if($gfjFieldDatePosted == 'otys_recruit.vacaturebank.publiceer_date') {
    $gfjDatePosted = date("Y-m-d",
    strtotime($vacancyPublicationFirstDate));
} elseif($gfjFieldDatePosted == 'otys_recruit.vacaturebank.invoerdatum') {
    $gfjDatePosted = date("Y-m-d", strtotime($vacancyEntryDateTime));
} elseif($gfjFieldDatePosted ==
'otys_recruit.vacaturebank.public_publish_date') {
    $gfjDatePosted = date("Y-m-d",
    strtotime($vacancyPublicationStartDate));
} else {
    $gfjDatePosted = date("Y-m-d",
    strtotime($vacancyPublicationFirstDate));
}
```

The code above will (based upon the retrieved mapping & the retrieved fields of the vacancy) define the variable used for the 'datePosted'-field in the Google for Jobs code.

```
// Create Google for Jobs code: Set Google for Jobs field
'hiringOrganization.name'
if($gfjFieldHiringOrganizationName == 'otys_klanten.klanten.klant') {
    $gfjHiringOrganizationName = $clientName;
} elseif($gfjFieldHiringOrganizationName == 'otys_klanten.websites.Name') {
    $gfjHiringOrganizationName = $websiteName;
} elseif($gfjFieldHiringOrganizationName ==
'otys_recruit.werkgevers.werkgever') {
    $gfjHiringOrganizationName = $vacancyRelation;
} else {
    $gfjHiringOrganizationName = $clientName;
}
```

The code above will (based upon the retrieved mapping & the retrieved fields of the vacancy) define the variable used for the 'hiringOrganization.name'-field in the Google for Jobs code.

```
// Create Google for Jobs code: Set Google for Jobs field
'hiringOrganization.sameAs'
if($gfjFieldHiringOrganizationSameAs == 'otys_klanten.websites.Website') {
    $gfjHiringOrganizationSameAs = $websiteUrl;
} elseif($gfjFieldHiringOrganizationSameAs ==
'otys_recruit.werkgevers.wg_website') {
    $gfjHiringOrganizationSameAs = $relationWebsite;
} else {
    $gfjHiringOrganizationSameAs = $relationWebsite;
}
```

The code above will (based upon the retrieved mapping & the retrieved fields of the vacancy) define the variable used for the 'hiringOrganization.sameAs'-field in the Google for Jobs code.

```
// Create Google for Jobs code: Set Google for Jobs field
'jobLocation.address.streetAddress'
if($gfjFieldStreetAddress == 'otys_recruit.vacaturebank.address') {
    $gfjJobLocationAddressStreetAddress = $vacancyLocationAddress;
} else {
    $gfjJobLocationAddressStreetAddress = $vacancyLocationAddress;
}
```

The code above will (based upon the retrieved mapping & the retrieve fields of the vacancy) define the variable used for the 'jobLocation.address.streetAddress'-field in the Google for Jobs code. Please note the current mapping interface only supports one field to be mapped, making the code above a bit 'funny'. However to be able to keep things consistent & easily change the code if in the future multiple fields are supported, an if-statement was implemented for this field as well.

```
// Create Google for Jobs code: Set Google for Jobs field
'jobLocation.address.addressLocality'
if($gfjFieldAddressLocality == 'otys_recruit.vacaturebank.locatie') {
    $gfjJobLocationAddressAddressLocality = $vacancyLocation;
} elseif($gfjFieldAddressLocality == 'otys_recruit.vacaturebank.city') {
    $gfjJobLocationAddressAddressLocality = $vacancyLocationCity;
} else {
    $gfjJobLocationAddressAddressLocality = $vacancyLocation;
}
```

The code above will (based upon the retrieved mapping & the retrieved fields of the vacancy) define the variable used for the 'jobLocation.address.addressLocality'-field in the Google for Jobs code.

```
// Create Google for Jobs code: Set Google for Jobs field
'jobLocation.address.addressRegion'
if($gfjFieldAddressRegion == 'otys_recruit.vacaturebank.state') {
    $gfjJobLocationAddressRegion = $vacancyLocationState;
} else {
    $gfjJobLocationAddressRegion = $vacancyLocationState;
}
```

The code above will (based upon the retrieved mapping & the retrieve fields of the vacancy) define the variable used for the 'jobLocation.address.addressRegion'-field in the Google for Jobs code. Please note the current mapping interface only supports one field to be mapped, making the code above a bit 'funny'. However to be able to keep things consistent & easily change the code if in the future multiple fields are supported, an if-statement was implemented for this field as well.

```
// Create Google for Jobs code: Set Google for Jobs field
'jobLocation.address.postalCode'
if($gfjFieldPostalCode == 'otys_recruit.vacaturebank.zipcode') {
    $gfjJobLocationAddressPostalCode = $vacancyLocationPostcode;
} else {
    $gfjJobLocationAddressPostalCode = $vacancyLocationPostcode;
}
```

The code above will (based upon the retrieved mapping & the retrieve fields of the vacancy) define the variable used for the 'jobLocation.address.PostalCode'-field in the Google for Jobs code. Please note the current mapping interface only supports one field to be mapped, making the code above a bit 'funny'. However to be able to keep things consistent & easily change the code if in the future multiple fields are supported, an if-statement was implemented for this field as well.

```
// Create Google for Jobs code: Set Google for Jobs field
'jobLocation.address.addressCountry'
if($gfjFieldAddressCountry == 'otys_recruit.vacaturebank.country_code') {
    $gfjJobLocationAddressAddressCountry = $vacancyLocationCountryCode;
} else {
    $gfjJobLocationAddressAddressCountry = $vacancyLocationCountryCode;
}
```

The code above will (based upon the retrieved mapping & the retrieve fields of the vacancy) define the variable used for the 'jobLocation.address.addressCountry'-field in the Google for Jobs code. Please note the current mapping interface only supports one field to be mapped, making the code above a bit 'funny'. However to be able to keep things consistent & easily change the code if in the future multiple fields are supported, an if-statement was implemented for this field as well.


```
// Create Google for Jobs code: Set Google for Jobs field 'title'  
if($gfjFieldTitle == 'otys_recruit.vacaturebank.functie_o') {  
    $gfjTitle = $vacancyTitle;  
} else {  
    $gfjTitle = $vacancyTitle;  
}
```

The code above will (based upon the retrieved mapping & the retrieve fields of the vacancy) define the variable used for the 'title'-field in the Google for Jobs code. Please note the current mapping interface only supports one field to be mapped, making the code above a bit 'funny'. However to be able to keep things consistent & easily change the code if in the future multiple fields are supported, an if-statement was implemented for this field as well.

```
// Create Google for Jobs code: Set Google for Jobs field  
'baseSalary.currency'  
if($gfjFieldBaseSalaryCurrency ==  
'otys_recruit.vacaturebank.extra_field_salary_currency') {  
    $gfjBaseSalaryCurrency = $vacancySalaryCurrency;  
} else {  
    $gfjBaseSalaryCurrency = $vacancySalaryCurrency;  
}
```

The code above will (based upon the retrieved mapping & the retrieve fields of the vacancy) define the variable used for the 'baseSalary.currency'-field in the Google for Jobs code. Please note the current mapping interface only supports one field to be mapped, making the code above a bit 'funny'. However to be able to keep things consistent & easily change the code if in the future multiple fields are supported, an if-statement was implemented for this field as well.

```
// Create Google for Jobs code: Set Google for Jobs field  
'baseSalary.value.value'  
if($gfjFieldBaseSalaryValueValue ==  
'otys_recruit.vacaturebank.extra_field_salary_value') {  
    $gfjFieldBaseSalaryValueValue = $vacancySalaryValue;  
} else {  
    $gfjFieldBaseSalaryValueValue = $vacancySalaryValue;  
}
```

The code above will (based upon the retrieved mapping & the retrieve fields of the vacancy) define the variable used for the 'baseSalary.value.value'-field in the Google for Jobs code. Please note the current mapping interface only supports one field to be mapped, making the code above a bit 'funny'. However to be able to keep things consistent & easily change the code if in the future multiple fields are supported, an if-statement was implemented for this field as well.

```
// Create Google for Jobs code: Set Google for Jobs field
'baseSalary.value.unitText'
if($gfjFieldBaseSalaryValueUnitText ==
'otys_recruit.vacaturebank.extra_field_salary_unit') {
    $gfjBaseSalaryValueUnitText = strtoupper($vacancySalaryUnit);
} else {
    $gfjBaseSalaryValueUnitText = strtoupper($vacancySalaryUnit);
}
```

The code above will (based upon the retrieved mapping & the retrieve fields of the vacancy) define the variable used for the 'baseSalary.value.unitText'-field in the Google for Jobs code. Please note the current mapping interface only supports one field to be mapped, making the code above a bit 'funny'. However to be able to keep things consistent & easily change the code if in the future multiple fields are supported, an if-statement was implemented for this field as well.

```
// Create Google for Jobs code: Set Google for Jobs field 'identifier.name'
if($gfjFieldIdentifierName == 'otys_klanten.klanten.klant') {
    $gfjIdentifierName = $clientName;
} elseif($gfjFieldIdentifierName == 'otys_klanten.websites.Name') {
    $gfjIdentifierName = $websiteName;
} elseif($gfjFieldIdentifierName == 'otys_recruit.werkgevers.werkgever') {
    $gfjIdentifierName = $response['result']['relation'];
} else {
    $gfjIdentifierName = $clientName;
}
```

The code above will (based upon the retrieved mapping & the retrieved fields of the vacancy) define the variable used for the 'identifier.name'-field in the Google for Jobs code.

```
// Create Google for Jobs code: Set Google for Jobs field
'identifier.value'
if($gfjFieldIdentifierValue == 'otys_recruit.vacaturebank.external_ref_nr')
{
    $gfjIdentifierValue = $vacancyExternalReferenceNr;
} elseif($gfjFieldIdentifierValue ==
'otys_recruit.vacaturebank.ref_nr_varchar') {
    $gfjIdentifierValue = $vacancyRefernceNr;
} else {
    $gfjIdentifierValue = $vacancyRefernceNr;
}
```

The code above will (based upon the retrieved mapping & the retrieved fields of the vacancy) define the variable used for the 'identifier.value'-field in the Google for Jobs code.

```

// Create Google for Jobs code: Set Google for Jobs field 'validThrough'
if($gfjFieldValidThrough == 'otys_recruit.vacaturebank.public_end_date') {
    $gfjValidThrough = date("Y-m-d",
strtotime($vacancyPublicationEndDate));
} else {
    $gfjValidThrough = date("Y-m-d",
strtotime($vacancyPublicationEndDate));
}

```

The code above will (based upon the retrieved mapping & the retrieve fields of the vacancy) define the variable used for the 'validThrough'-field in the Google for Jobs code. Please note the current mapping interface only supports one field to be mapped, making the code above a bit 'funny'. However to be able to keep things consistent & easily change the code if in the future multiple fields are supported, an if-statement was implemented for this field as well.

```

// Create Google for Jobs code: Set Google for Jobs field 'description'
if($gfjFieldValidThrough == 'otys_recruit.vacaturebank.allTexts') {
    $gfjDescription = $vacancyAllTexts;
} else {
    $gfjDescription = $vacancyAllTexts;
}

```

The code above will (based upon the retrieved mapping, the retrieve fields of the vacancy & the retrieved vacancy texts settings) define the variable used for the 'description'-field in the Google for Jobs code. Please note the current mapping interface only supports one field to be mapped, making the code above a bit 'funny'. However to be able to keep things consistent & easily change the code if in the future multiple fields are supported, an if-statement was implemented for this field as well.

```

// Create Google for Jobs code: Set Google for Jobs field 'employmentType'
if($gfjFieldEmploymentType == 'otys_recruit.vacaturebank.branche') {
    foreach($gfjFieldEmploymentTypes as $gfjFieldEmploymentType) {
        if($gfjFieldEmploymentType['valueOtys'] == preg_replace('/1/',
'', key($vacancyMatchCriteria1))) {
            $gfjEmploymentType = $gfjFieldEmploymentType['ValueGfj'];
        }
    }
} elseif($gfjFieldEmploymentType ==
'otys_recruit.vacaturebank.bedrcategorie') {
    foreach($gfjFieldEmploymentTypes as $gfjFieldEmploymentType) {
        if($gfjFieldEmploymentType['valueOtys'] == preg_replace('/2/',
'', key($vacancyMatchCriteria2))) {
            $gfjEmploymentType = $gfjFieldEmploymentType['ValueGfj'];
        }
    }
} elseif($gfjFieldEmploymentType ==
'otys_recruit.vacaturebank.verzekering') {
    foreach($gfjFieldEmploymentTypes as $gfjFieldEmploymentType) {
        if($gfjFieldEmploymentType['valueOtys'] == preg_replace('/3/',
'', key($vacancyMatchCriteria3))) {
            $gfjEmploymentType = $gfjFieldEmploymentType['ValueGfj'];
        }
    }
}

```

```

} elseif($gfjFieldEmploymentType ==
'otys_recruit.vacaturebank.functienaam') {
    foreach($gfjFieldEmploymentTypes as $gfjFieldEmploymentType) {
        if($gfjFieldEmploymentType['valueOtys'] == preg_replace('/4_',
'', key($vacancyMatchCriteria4))) {
            $gfjEmploymentType = $gfjFieldEmploymentType['ValueGfj'];
        }
    }
} elseif($gfjFieldEmploymentType == 'otys_recruit.vacaturebank.dienstvb') {
    foreach($gfjFieldEmploymentTypes as $gfjFieldEmploymentType) {
        if($gfjFieldEmploymentType['valueOtys'] == preg_replace('/5_',
'', key($vacancyMatchCriteria5))) {
            $gfjEmploymentType = $gfjFieldEmploymentType['ValueGfj'];
        }
    }
} elseif($gfjFieldEmploymentType == 'otys_recruit.vacaturebank.functiegr')
{
    foreach($gfjFieldEmploymentTypes as $gfjFieldEmploymentType) {
        if($gfjFieldEmploymentType['valueOtys'] == preg_replace('/6_',
'', key($vacancyMatchCriteria6))) {
            $gfjEmploymentType = $gfjFieldEmploymentType['ValueGfj'];
        }
    }
} elseif($gfjFieldEmploymentType == 'otys_recruit.vacaturebank.salaris') {
    foreach($gfjFieldEmploymentTypes as $gfjFieldEmploymentType) {
        if($gfjFieldEmploymentType['valueOtys'] == preg_replace('/7_',
'', key($vacancyMatchCriteria7))) {
            $gfjEmploymentType = $gfjFieldEmploymentType['ValueGfj'];
        }
    }
} elseif($gfjFieldEmploymentType == 'otys_recruit.vacaturebank.gewuren') {
    foreach($gfjFieldEmploymentTypes as $gfjFieldEmploymentType) {
        if($gfjFieldEmploymentType['valueOtys'] == preg_replace('/8_',
'', key($vacancyMatchCriteria8))) {
            $gfjEmploymentType = $gfjFieldEmploymentType['ValueGfj'];
        }
    }
} elseif($gfjFieldEmploymentType == 'otys_recruit.vacaturebank.opleidr') {
    foreach($gfjFieldEmploymentTypes as $gfjFieldEmploymentType) {
        if($gfjFieldEmploymentType['valueOtys'] == preg_replace('/9_',
'', key($vacancyMatchCriteria9))) {
            $gfjEmploymentType = $gfjFieldEmploymentType['ValueGfj'];
        }
    }
} elseif($gfjFieldEmploymentType == 'otys_recruit.vacaturebank.opleidn') {
    foreach($gfjFieldEmploymentTypes as $gfjFieldEmploymentType) {
        if($gfjFieldEmploymentType['valueOtys'] ==
preg_replace('/10_', '', key($vacancyMatchCriteria10))) {
            $gfjEmploymentType = $gfjFieldEmploymentType['ValueGfj'];
        }
    }
} elseif($gfjFieldEmploymentType == 'otys_recruit.vacaturebank.wervaring')
{
    foreach($gfjFieldEmploymentTypes as $gfjFieldEmploymentType) {
        if($gfjFieldEmploymentType['valueOtys'] ==
preg_replace('/11_', '', key($vacancyMatchCriteria11))) {
            $gfjEmploymentType = $gfjFieldEmploymentType['ValueGfj'];
        }
    }
} elseif($gfjFieldEmploymentType == 'otys_recruit.vacaturebank.regio') {
    foreach($gfjFieldEmploymentTypes as $gfjFieldEmploymentType) {

```

```

        if($gfjFieldEmploymentType['valueOtys'] ==
preg_replace('/12_/', '', key($vacancyMatchCriteria12))) {
            $gfjEmploymentType = $gfjFieldEmploymentType['ValueGfj'];
        }
    }
} elseif($gfjFieldEmploymentType ==
'otys_recruit.vacaturebank.beoordeling') {
    foreach($gfjFieldEmploymentTypes as $gfjFieldEmploymentType) {
        if($gfjFieldEmploymentType['valueOtys'] ==
preg_replace('/13_/', '', key($vacancyMatchCriteria13))) {
            $gfjEmploymentType = $gfjFieldEmploymentType['ValueGfj'];
        }
    }
} elseif($gfjFieldEmploymentType ==
'otys_recruit.vacaturebank.custom_crit_data_1') {
    foreach($gfjFieldEmploymentTypes as $gfjFieldEmploymentType) {
        if($gfjFieldEmploymentType['valueOtys'] ==
preg_replace('/14_/', '', key($vacancyMatchCriteria14))) {
            $gfjEmploymentType = $gfjFieldEmploymentType['ValueGfj'];
        }
    }
} elseif($gfjFieldEmploymentType ==
'otys_recruit.vacaturebank.custom_crit_data_2') {
    foreach($gfjFieldEmploymentTypes as $gfjFieldEmploymentType) {
        if($gfjFieldEmploymentType['valueOtys'] ==
preg_replace('/15_/', '', key($vacancyMatchCriteria15))) {
            $gfjEmploymentType = $gfjFieldEmploymentType['ValueGfj'];
        }
    }
} elseif($gfjFieldEmploymentType ==
'otys_recruit.vacaturebank.custom_crit_data_3') {
    foreach($gfjFieldEmploymentTypes as $gfjFieldEmploymentType) {
        if($gfjFieldEmploymentType['valueOtys'] ==
preg_replace('/16_/', '', key($vacancyMatchCriteria16))) {
            $gfjEmploymentType = $gfjFieldEmploymentType['ValueGfj'];
        }
    }
} elseif($gfjFieldEmploymentType ==
'otys_recruit.vacaturebank.custom_crit_data_4') {
    foreach($gfjFieldEmploymentTypes as $gfjFieldEmploymentType) {
        if($gfjFieldEmploymentType['valueOtys'] ==
preg_replace('/17_/', '', key($vacancyMatchCriteria17))) {
            $gfjEmploymentType = $gfjFieldEmploymentType['ValueGfj'];
        }
    }
} elseif($gfjFieldEmploymentType ==
'otys_recruit.vacaturebank.custom_crit_data_5') {
    foreach($gfjFieldEmploymentTypes as $gfjFieldEmploymentType) {
        if($gfjFieldEmploymentType['valueOtys'] ==
preg_replace('/18_/', '', key($vacancyMatchCriteria18))) {
            $gfjEmploymentType = $gfjFieldEmploymentType['ValueGfj'];
        }
    }
} else {
    foreach($gfjFieldEmploymentTypes as $gfjFieldEmploymentType) {
        if($gfjFieldEmploymentType['valueOtys'] == preg_replace('/5_/',
'', key($vacancyMatchCriteria5))) {
            $gfjEmploymentType = $gfjFieldEmploymentType['ValueGfj'];
        }
    }
}

```

```
}

```

The code above will use the retrieved information of the vacancy and based upon the retrieved sub mapping for the employment type define the employment type variable for the Google for Jobs code. Please note in the OTYS system it is possible to define multiple employment types for a vacancy, while Google for Jobs only supports one. Since the OTYS system does not have a 'primary option'-functionality for this, the code above is just picking one in case of multiple selected options (from a technical perspective there is of course a 'logic' which one is picked, but this is more of a technical thing than something that can be 'explained' to end users).

```
// Create Google for Jobs code: Create array
$gfjArray = array(
  "@context"=>"https://schema.org/",
  "@type"=>"JobPosting",
  "title"=>"$gfjTitle",
  "description"=>"$gfjDescription",
  "identifier"=>array(
    "@type"=>"PropertyValue",
    "name"=>"$gfjIdentifierName",
    "value"=>"$gfjIdentifierValue",
  ),
  "datePosted"=>"$gfjDatePosted",
  "validThrough"=>"$gfjValidThrough",
  "directApply"=>$gfjDirectApply,
  "employmentType"=>"$gfjEmploymentType",
  "hiringOrganization"=>array(
    "@type"=>"Organization",
    "name"=>"$gfjHiringOrganizationName",
    "sameAs"=>"$gfjHiringOrganizationSameAs",
    "logo"=>"$gfjHiringOrganizationLogo",
  ),
  "jobLocation"=>array(
    "@type"=>"Place",
    "address"=>array(
      "@type"=>"PostalAddress",
      "streetAddress"=>"$gfjJobLocationAddressStreetAddress",
      "addressLocality"=>"$gfjJobLocationAddressAddressLocality",
      "addressRegion"=>"$gfjJobLocationAddressRegion",
      "postalCode"=>"$gfjJobLocationAddressPostalCode",
      "addressCountry"=>"$gfjJobLocationAddressAddressCountry",
    ),
  ),
  "baseSalary"=>array(
    "@type"=>"MonetaryAmount",
    "currency"=>"$gfjBaseSalaryCurrency",
    "value"=>array(
      "@type"=>"QuantitativeValue",
      "value"=>$gfjFieldBaseSalaryValueValue,
      "unitText"=>"$gfjBaseSalaryValueUnitText",
    ),
  ),
);

```

The code above will create an array of the (to be) Google for Jobs-code based upon all defined variables in previous script operations.

```

// Create Google for Jobs code: Remove empty elements from array
if(!$gfjArray['title']) {
    unset($gfjArray['title']);
}
if(!$gfjArray['description']) {
    unset($gfjArray['description']);
}
if(!$gfjArray['identifier']['name']) {
    unset($gfjArray['identifier']['name']);
}
if(!$gfjArray['identifier']['value']) {
    unset($gfjArray['identifier']['value']);
}
if(!$gfjArray['datePosted'] || $gfjArray['datePosted'] == '1970-01-01') {
    unset($gfjArray['datePosted']);
}
if(!$gfjArray['validThrough'] || $gfjArray['validThrough'] == '1970-01-01') {
    {
        unset($gfjArray['validThrough']);
    }
}
if(!$gfjArray['directApply']) {
    unset($gfjArray['directApply']);
}
if(!$gfjArray['employmentType']) {
    unset($gfjArray['employmentType']);
}
if(!$gfjArray['hiringOrganization']['name']) {
    unset($gfjArray['hiringOrganization']['name']);
}
if(!$gfjArray['hiringOrganization']['sameAs']) {
    unset($gfjArray['hiringOrganization']['sameAs']);
}
if(!$gfjArray['hiringOrganization']['logo']) {
    unset($gfjArray['hiringOrganization']['logo']);
}
if(!$gfjArray['jobLocation']['address']['streetAddress']) {
    unset($gfjArray['jobLocation']['address']['streetAddress']);
}
if(!$gfjArray['jobLocation']['address']['addressLocality']) {
    unset($gfjArray['jobLocation']['address']['addressLocality']);
}
if(!$gfjArray['jobLocation']['address']['addressRegion']) {
    unset($gfjArray['jobLocation']['address']['addressRegion']);
}
if(!$gfjArray['jobLocation']['address']['postalCode']) {
    unset($gfjArray['jobLocation']['address']['postalCode']);
}
if(!$gfjArray['jobLocation']['address']['addressCountry']) {
    unset($gfjArray['jobLocation']['address']['addressCountry']);
}
if(!$gfjArray['baseSalary']['currency']) {
    unset($gfjArray['baseSalary']['currency']);
}
if(!$gfjArray['baseSalary']['value']['value']) {
    unset($gfjArray['baseSalary']['value']['value']);
}
if(!$gfjArray['baseSalary']['value']['unitText']) {
    unset($gfjArray['baseSalary']['value']['unitText']);
}
}

```

The code above will remove all empty elements from the previously created array.

```
// Create Google for Jobs code: Create JSON
$gfjJson = json_encode($gfjArray);

// Create Google for Jobs code: Create code
$gfjCode = "<script type=\"application/ld+json\">$gfjJson</script>";
```

The code above will encode the previously created & adjusted array into a JSON and place it in script-tags; so that it meets Google for Jobs-standards. The 'end result' will be placed in variable '\$gfjCode'.

```
// Create Google for Jobs code: End follow up actions if Google for Jobs is enabled
};
```

The code above will end executing follow up actions, after the previous check if Google for Jobs is enabled for the client.

```
// End session: Start follow up actions if OWS session needs to be ended
if($createOwsSession == true) {

    // End session: Create array request
    $request = [
        'jsonrpc'=>'2.0',
        'method'=>'logout',
        'params'=>[
            $sessionToken
        ],
        'id'=>1,
    ];

    // End session: Send OWS request
    sendOwsRequest();

// End session: End follow up actions if OWS session needs to be ended
}
```

The code above will check if previously an OWS session has been created and -if so- end the session.

```
// General: Return Google for Jobs code
return $gfjCode;
```

The code above will return the Google for Jobs-code stored in the '\$gfjCode'-variable.


```
// General: End function create Google for Jobs code  
}
```

The code above will end the function that started it all. 😊