Whitepaper & Sample Project for Job Board API Integration

Last Modified on 01/31/2024 12:59 pm CST

Overview

The following document provides information and instruction to enable any third-party developers to use the TempWorks API if you are going to custom-build a job board.

This article covers the following:

1. Authenticating to the API
2. Getting and Searching Job Board Orders
3. Getting Job Board Order Details for a Specific Order
4. Job Board Configurations
5. Creating an Employee Record
6. Creating a Web Candidate Order Candidate Record
7. Constructing an HRCenter Application URL

Note All URLs in the reference documentation use the same base URL. The URL will depend on who is hosting the installation of TempWorks you are using. For the majority of clients hosted by TempWorks Software, this URL will be https://api.ontempworks.com/.

Note For a sample template project covering authenticating, making API calls for creating employees, web candidates, and searching job orders, please navigate to https://tempworks.visualstudio.com/Sample%20Projects/_git/JobBoardAPIIntegration.

Authenticating to the API

Personal access tokens are used to give access to the TempWorks Open API. This allows you to create your own applications and gives opportunity to your vendors to connect directly to our system. When you use the personal access token, you are authenticating to our system with the service representative account selected when generating the token. Also, what endpoints you have access depends on the scopes selected when generating the token.

Obtaining a Personal Access Token

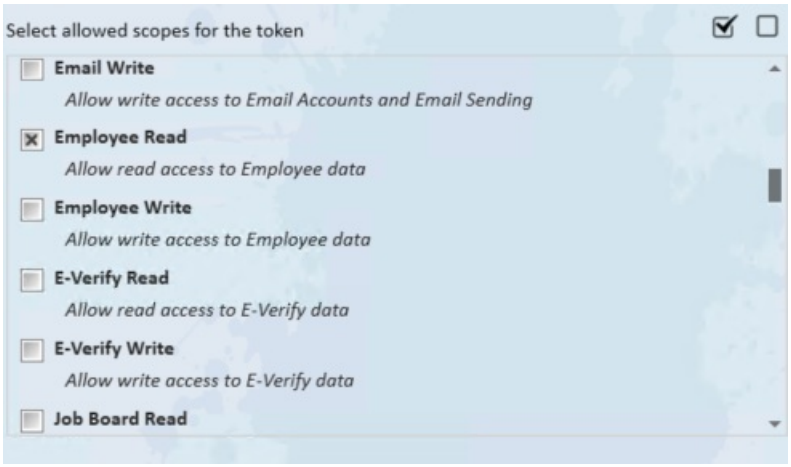
You have to generate your personal access tokens through the Enterprise user interface. You can follow these

steps.

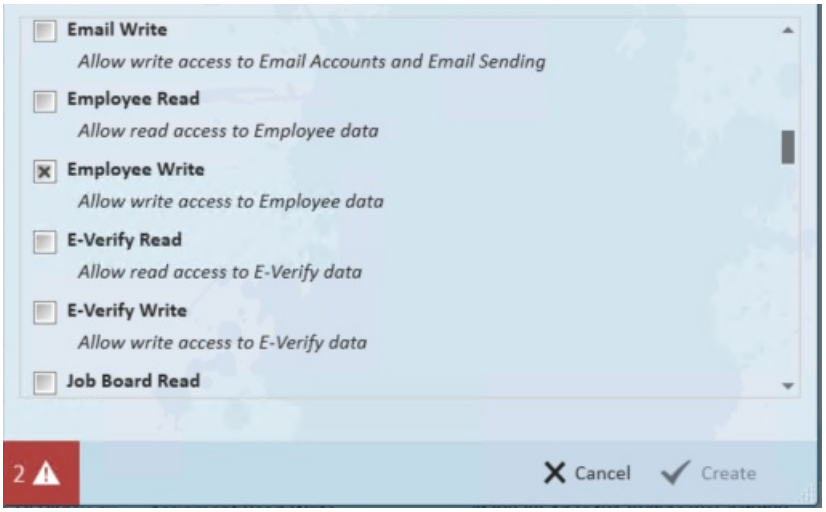
Note If you are using the token for a service or an application and do not wish to mirror a dedicated service representative account, you will have to create a new system TempWorks service representative and configure that account's permissions and hierarchy according to what you want your service or application to be able to see.

For security reasons, every personal access token must have an expiration date. After a token has expired, it may no longer be used to authenticate to the Open API, and you will need to generate a new token.

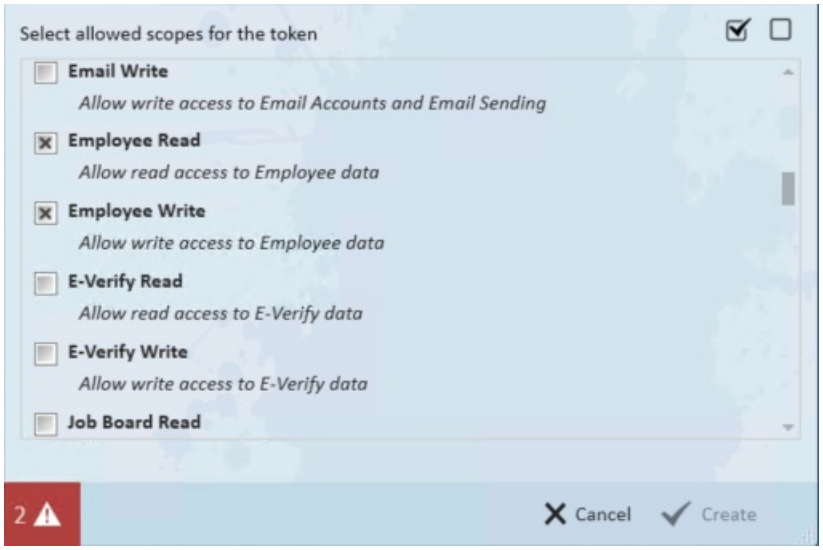
When you create your personal access tokens, you should carefully choose the right scopes in order to be able to use the API end points you need. If you want your token to have access only to GET endpoints for your record, select the checkbox for "{record name} Read". For example, we want to have access for Employee GET endpoints. You have to check "Employee Read":



If you want your token to have access to POST, PUT, PATCH and DELETE endpoints for your record, select the checkbox for "{record name} Write". For example, we want to have accesses for Employee POST and PUT endpoints. You have to check "Employee Write":



If you select both read and write, then your token will have access to all available endpoints for this record:



You can see all selected scopes here:

Authenticating to the API

You have to add your personal access token in Base64String to the x-tw-token header in your client which you use to make requests to the API V3:

Example in C# HttpClient:

```
using (var handler = new HttpClientHandler())
using (var client = new HttpClient(handler)
{
   BaseAddress = new Uri("https://api.ontempworks.com/")>
})
{
   var accountSid = "...";
   var authToken = "...";

   var tokenDetails = Encoding.UTF8.GetBytes($"{accountSid}:{authToken}");
   var base64Token = Convert.ToBase64String(tokenDetails);

   client.DefaultRequestHeaders.Add("x-tw-token", base64Token);

   HttpResponseMessage response = await client.PostAsync("your-enpoint", "your data");
}
```

If you cannot specify a header, you can pass the personal access token inBase64String as a tw-token query parameter:

Example in C# HttpClient:

```
using (var handler = new HttpClientHandler())
using (var client = new HttpClient(handler)
{
   BaseAddress = new Uri("https://api-canary.ontempworks.com/")
})
{
   var accountSid = "...";
   var authToken = "...";

   var tokenDetails = Encoding.UTF8.GetBytes($"{accountSid}:{authToken}");
   var base64Token = Convert.ToBase64String(tokenDetails);

   HttpResponseMessage response = await client.PostAsync($"your-enpoint?tw-token={base64Token}", "your data");
}
```

Note More information about how to use the authentication token can be found here.

Common Exceptions

Standard HTTP status codes:

Status Code	Reason
400 Bad Request	The server can't process the request.
401 Unauthorized	Your authentication token is invalid or expired.
403 Forbidden	Your token doesn't have the necessary scope to access this endpoint or the service representative you have in your token is at a hierarchy level that does not allow this operation or the hierarchy you provided does not allow this operation.
404 Not Found	The endpoint you try to react is not found or the record you are querying is not visible to the service rep the token was issued for.
409 Conflict	The record already exists or already exists in a state such that the operation you are attempting is not possible.

Non-standard HTTP status codes:

Status Code	Reason
520 External Service Connection Error	We are unable to connect with an external service that is used as part of making your request. Most often this is because they are experiencing an outage.

Status Code	Reason
521 External Service Processing/Unknown Error	We experienced an unexpected error when performing an action against an external service. This may be due to unexpected validation or bad configuration.

Back to Top

Getting and Searching Job Board Orders

GET /Search/JobBoard

The endpoint will return the following response codes:

Status Code	Reason
200 Success	The request was successful, and the response body contains the representation requested.
400 Bad Request	Provided parameter value is not supported or does not exist. Parameters such as `countryCode`, `distanceUnitId`, `region` have preset possible ranges and will return a Bad Request along with a message in the response body indicating the problem.

For a Job Order to be returned in that list, it needs to be within the executing Service Rep's hierarchy. It also needs to be with an Order Status of either "Unfilled", "Unfilled - Replace", "Partially Filled" or "Master". The Skill Code used on the Job Order also needs to be set as "IsPublic" and the Order's Branch needs to be set as "IsWebPublic".

A sample GET request might look like this:

```
// GET JobBoardOrders
public async Task<PagedList<JobBoardOrder>> SearchJobBoardOrders(int sortBy, int countryCode, string textSearch
= null, string postalCode = null, string municipality = null, string region = null, double? latitude = null, double? longit
ude = null, int? distance = null, long? distanceUnitId = null, int skip = 0, int take = 1000)
{
    var result = await GetAsync<PagedList<JobBoardOrder>>(
          $"Searches/JobBoard",
          new Dictionary<string, string>{
                { "sortBy", sortBy.ToString() },
                { "countryCode", countryCode.ToString() },
                { "textSearch", textSearch },
                { "postalCode", postalCode },
                { "municipality", municipality },
                { "region",  region},
                { "latitude", latitude.ToString() },
                { "longitude", longitude.ToString() },
                { "distance",  distance.ToString() },
                { "distanceUnitId", distanceUnitId.ToString() },
                { "skip", skip.ToString() },
                { "take",  take.ToString() }
           });

    return result;
}

private async Task<HttpResponseMessage> GetAsync(string endpointUrl, IDictionary<string, string> queryParams =
null)
{
    var queryUrl = queryParams == null ? endpointUrl : QueryHelpers.AddQueryString(endpointUrl, queryParams);
    var request = new HttpRequestMessage(HttpMethod.Get, $"{_baseUrl}{queryUrl}");

    return await _httpClient.SendAsync(request);
}

private async Task<T> GetAsync<T>(string url, IDictionary<string, string> queryParams = null)
{
    var result = await GetAsync(url, queryParams);

    return await ConvertToModel<T>(result);
}

private async Task<T> ConvertToModel<T>(HttpResponseMessage response)
{
     if (response.IsSuccessStatusCode)
     {
        var json = await response.Content.ReadAsStringAsync();
        var result = JsonConvert.DeserializeObject<T>(json);

        return result;
     }
     throw new Exception();
}
```

The result is in the following JSON format:

```
{
"data":
    [
     {
      "jobOrderId": 0,
      "municipality": "ORLANDO",
      "region": "FL",
      "postalCode": "string",
      "dateOrderTaken": "2022-03-25T12:34:43.826Z",
      "jobTitle": "Job Title",
      "jobDescription": "Public Job Description",
      "jobDescriptionContentType": "text/html; charset=utf-8",
      "category": "Category Name",
      "orderType": "Daily Pay / Labor",
      "skills": [
        "Skill 1",
        "Skill 2"
      ],
      "distance": 12,
      "jobBoardUrl": "Url to the JobBoard detailing this JobOrder"
    },
   ...
  ],
  "totalCount": 999
}
```


Parameters

CountryCode STRING REQUIRED

You can get a list of all valid country codes that can be passed in as parameters.

```
// GET Countries
HttpResponseMessage response = await client.GetAsync("DataLists/countries");
```

The result is in the following JSON format:


```
{
"data":
   [
    {
     "countryCode": 999,
     "country": "CountryName",
     "countryCallingCode": 1
    },...
  ],
  "totalCount": 9
}
```


Location Based Search

There are three ways to run a location-based search for the Job Orders depending on the parameters of the request. They all require a "Distance" and "DistanceUnitId".

Distance INT and **DistanceUnitId** INT

The "Distance" is an integer representing the selected unit. For the "DistanceUnitId" we can use:

- 3 - (Miles)
- 4 - (Kilometers)

> *Note* Job Orders that don't fall within the location's distance radius are excluded.

The following parameters are in order of priority:

- **Municipality**(*City*) STRING and **Region**(*State*) STRING
  - They determine the central location around which to search Job Board Orders for.
  - Depending on the "CountryCode" selected, Region(*State*) might not be required.
- **PostalCode**(*Zip Code*) STRING
  - If not passed in as a parameter, it will be assigned based on the combination of "CountryCode", "Municipality", and "Region" for that Country if applicable.
  - If it is passed in as a parameter, setting "Municipality" and "Region" is not required and will result in overwriting it by getting the first "PostalCode" available for that general location.
  - The "PostalCode" is then used for searching Job Orders for that specific location by combining it with "Distance" and "DistanceUnitId".
- **Latitude** LONG and **Longitude** LONG
  - If the "PostalCode" cannot be assigned due to lacking all of the above parameters, the "Latitude" and "Longitude" are used as a fallback.
  - They need to be specified together, following the decimal notation.
  - "Latitude" and "Longitude" have no effect if the "PostalCode" was assigned a value in any of the previous steps above.
- **TextSearch** STRING
  - "TextSearch" is a parameter that looks for matching words in Job Orders mainly by their description, title and skills to further refine the results.
  - It can be used with "SortBy".
- **SortBy** STRING REQUIRED
  - "SortBy" is used to order the results by specific criteria.
  - The available options are as follows:
    1. (Term Relativity Ranking): Used together with the "TextSearch" parameter to order by the most relevant matches descending. For instance, a Job Order containing the phrase "Forklift operator" five times will have a higher rank than another Job Order that only lists it twice, therefore it will be towards the end of the list.
    2. (Job Title): Order by the job title alphabetically.
    3. (Distance): Order by distance to the desired location, starting with the closest Job Orders.
    4. (Job Order Created): Orders by the Job Order's creation date and returns the most recent orders first.

Default Configurations

A set of predefined configurations affect the search results in different ways:

- **"CanSearchByLocation"** is enabled by default and allows narrowing down Job Orders by all the location based parameters from 2. If this is disabled, the location parameters are ignore and all Job Board Orders are returned instead.
- **"Job Description Fallback"** and **"Job Title Fallback"** are enabled by default and display the "Private" Job description and title, but only if the "Public" one is not set.
- **"Hide Old Dates"** is disabled by default, but if enabled, together with the "Old Date Cutoff" integer, it hides the "DateTimeOrderTaken" value for orders that pass that threshold.
- **"Hide Worksite"** is disabled by default, but when enabled it hides the "Municipality(City)", "Region(State)" and "PostalCode" of the JobOrder.

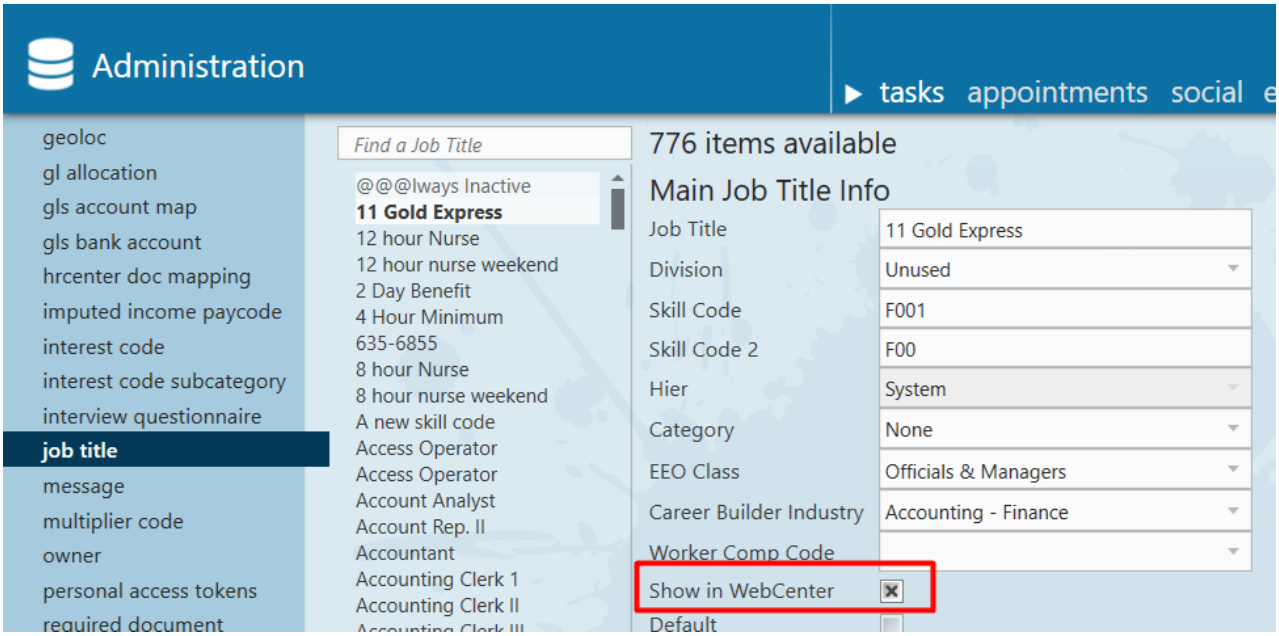
Back to Top

Getting Job Board Order Details for a Specific Order

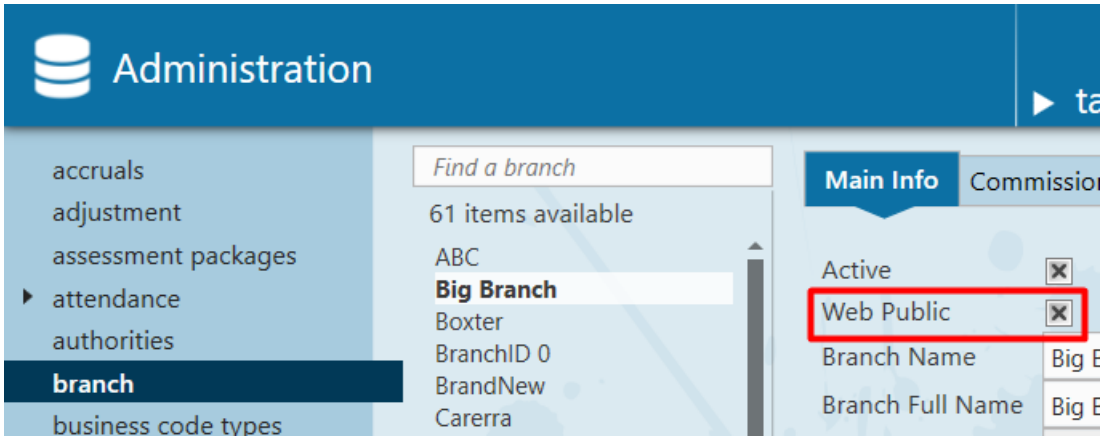
The Job Order Job Board Details API endpoint returns information about the job order that is meant to be shown on the Job Board. This endpoint returns more detailed information about the job order compared to what is within the Getting and Searching Job Board Orders article.

Whether or not a job order is suitable for showing on a job board is determined based on the following criteria:

- **DoNotPostToWeb config value**
  - 1- Order would not be available.
  - 0- Order would be available.
  - This value is taken into account when a job order is created.
  - Changing the config value will NOT affect the already created job orders.
- **Active job status**
  - Inactive job orders will NOT be returned.
- **Unfilled status**
  - Filled job orders will NOT be returned.
- **Public status of the associated job title**
  - Refers to the "Show In Web Center" checkbox within Enterprise > All Options > Administration >Job title.
  - TempWorks will only show job offers with "Show In Web Center" = 1.



- **Public Web Status of the associated branch**
  - Refers to the "Web Public" checkbox within Enterprise > All Options > Administration > Branch.
  - TempWorks will only show job offers with "Web Public" = 1.



GET /JobOrders/{jobOrderId}/jobBoardDetails

The above endpoint will return the following response codes:

Status Code	Reason
200 Success	The request was successful and the response body contains the representation requested.
404 Not Found	The specified job order does not exist or does not meet the criteria mentioned above.

Sample C# Request:

```
// GET JobOrderJobBoardDetails
public async Task<JobBoardOrder> GetJobOrderJobBoardDetails(long jobOrderId)
{
    var result = await GetAsync<JobBoardOrder>($"JobOrders/{jobOrderId}/jobBoardDetails");

    return result;
}

private async Task<HttpResponseMessage> GetAsync(string endpointUrl, IDictionary<string, string> queryParams
= null)
{
    var queryUrl = queryParams == null ? endpointUrl : QueryHelpers.AddQueryString(endpointUrl, queryParams);
    var request = new HttpRequestMessage(HttpMethod.Get, $"{_baseUrl}{queryUrl}");

    return await _httpClient.SendAsync(request);
}

private async Task<T> GetAsync<T>(string url, IDictionary<string, string> queryParams = null)
{
    var result = await GetAsync(url, queryParams);

    return await ConvertToModel<T>(result);
}

private async Task<T> ConvertToModel<T>(HttpResponseMessage response)
{
    if (response.IsSuccessStatusCode)
    {
        var json = await response.Content.ReadAsStringAsync();
        var result = JsonConvert.DeserializeObject<T>(json);

        return result;
    }
    throw new Exception();
}
```

Sample Response:

Note Some of the fields` information could be missing or different based on the applied configurations. For more information, please refer to the Job board Configurations article.

```
{
   "jobOrderId": 4295037574,
   "orderId": 4295037574,
   "dateOrderTaken": "2020-10-12T08:49:00",
   "jobDescription": "test",
   "jobDescriptionContentType": "text/plain",
   "jobTitle": "Accountant",
   "category": "General Office",
   "jobOrderType": "Direct Hire",
   "skills": [
      "12 hour Nurse",
      "8 hour nurse",
      "Light office duty",
      "1st Shift",
      "2nd Shift",
      "10961"
   ],
   "featured": null,
   "pay": 0.0000,
   "payFrequency": "Weekly",
   "payMin": 0.0000,
   "payMax": 0.0000,
   "payCurrency": null,
   "branchId": 1604,
   "branchName": "High Tech SE",
   "jobBoardUrl": "https://JobBoard.ontempworks.com/HiTechQA//jobs/details/4295037574",
   "educationSummary": null,
   "experienceSummary": null,
   "publishDateTime": "2020-10-12T13:50:00+00:00",
   "address": {
      "attentionTo": null,
      "street1": "439 Bergeron Ave",
      "street2": "",
      "municipality": "Eagan",
      "region": "MN",
      "postalCode": "55121",
      "country": null,
      "countryCode": null,
      "location": null,
      "dateAddressStandardized": null
   }
}
```


Additional Information and Purpose about Response Fields

- **Featured**
  - If set to true, job should be highlighted/starred.
- **PayFrequency**
  - Possible values:
    - Monthly
    - Bi-Monthly
    - Bi-Weekly
    - Weekly
    - Daily
- **PublishDateTime**
  - The date, time, and UTC offset are included.

- **JobDescriptionContentType**
  - This provides information about the content type information stored in the "JobDescription" field. This could be used to determine how/whether the information would be displayed.
  - Possible values:
    - text/plain
    - text/html; charset=utf-8

Obsolete Fields

- **OrderId**
  - Use the "JobOrderId" field instead.
- **JobBoardUrl**
- **PayCurrency**

Additional Job Order Information

Note In order to call the endpoints mentioned below, you would need to have an **order-read** scope.

For more authentication information, please refer to Authenticating to the API.

If you would like to show more information about the job order, there are several endpoints that return additional job order details:

Job order Summary

- GET /JobOrders/{jobOrderId}/summary

Sample Response

```
{
   "branchId": 1604,
   "branch": "High Tech SE",
   "jobOrderStatusId": 19,
   "jobOrderStatus": "Unfilled",
   "isActive": true,
   "positionsRequired": 1,
   "positionsFilled": 0,
   "customerId": 4294969450,
   "customerName": "Aardvark Industries Inc",
   "departmentName": "Aardvark Department 1",
   "jobTitleId": 1,
   "jobTitle": "Accountant"
}
```

Job Order information

- GET /JobOrders/{jobOrderId}

Sample Response


```
{
   "jobOrderId": 4295037574,
   "branchId": 1604,
   "branch": "High Tech SE",
   "jobOrderTypeId": 1,
   "jobOrderType": "Direct Hire",
   "jobTitleId": 1,
   "jobTitle": "Accountant",
   "jobDescription": "test",
   "payRate": 0.0000,
   "billRate": 0.0000,
   "jobOrderStatusId": 19,
   "jobOrderStatus": "Unfilled",
   "isActive": true,
   "positionsRequired": 1,
   "positionsFilled": 0,
   "customerId": 4294969450,
   "customerName": "Aardvark Industries Inc",
   "departmentName": "Aardvark Department 1",
   "jobOrderDurationId": 33,
   "jobOrderDuration": "Indef",
   "dateOrderTaken": "2020-10-12T08:49:00",
   "startDate": "2020-10-13T00:00:00",
   "supervisorContactId": null,
   "supervisorFirstName": null,
   "supervisorLastName": null,
   "supervisorOfficePhoneCountryCallingCode": null,
   "supervisorOfficePhone": null,
   "doNotAutoClose": false,
   "usesTimeClock": false,
   "usesPeopleNet": false,
   "notes": null,
   "alternateJobOrderId": null,
   "dressCode": "Business Professional",
   "safetyNotes": null,
   "directions": "South on Hwy 220 to Bergeron Ave. exit",
   "serviceRepId": 1234,
   "serviceRep": "Rep Name",
   "salesTeamId": 2,
   "salesTeam": "dwood",
   "publicJobTitle": null,
   "publicJobDescription": "",
   "publicPostingDate": "2020-10-12T13:50:00+00:00",
   "doNotPostPublicly": false,
   "publicJobDescriptionContentType": "",
   "publicEducationSummary": null,
   "publicExperienceSummary": null,
   "showPayRate": true,
   "showWorksiteAddress": true
}
```


Back to Top

Job Board Configurations

At the moment, the TempWorks API does not have an existing endpoint that can be used for retrieving or clearing Job Board configurations. However, TempWorks does have a set of default settings that affect the Job Board endpoints in different ways:

- **"CanSearchByLocation"** is enabled by default and allows narrowing down Job Orders by all the location-based parameters. If this is disabled, the location parameters are ignored, and all Job Board Orders are returned instead.
- **"Job Description Fallback"** and **"Job Title Fallback"** are enabled by default and display the "Private" job description and title, but only if the "Public" one is not set on the Job Order itself.
- **"Hide Old Dates"** is disabled by default, but if enabled, together with the "Old Date Cutoff" integer, they hide the "DateTimeOrderTaken" value for orders that pass that threshold. For instance, enabling it and setting the cutoff to 20, will make it so orders placed more than 20 days ago do not display the date on which they were taken.
- **"Hide Worksite"** is disabled by default, but when enabled it hides the "Municipality(City)", "Region(State)" and "PostalCode" of the Job Order when calling the Search Job Board endpoint. These parameters are still available and won't be ignored in the filtering.

Back to Top

Creating an Employee Record

There are two ways to create an employee. These include using the API endpoint or through uploading a resume which will parse the resume information and add it to the employee record.

Creating an Employee Record Using the Endpoint

The following is the API endpoint that will be used to create the employee record:

- POST /Employees

The above endpoint will return the following response codes:

Status Code	Reason
201 Created	The employee is created successfully. The response will include the new employee Id.
403 Forbidden	You did not provide a hierarchy ("HierId") and the service rep you are authenticated as is at a hierarchy level that does not allow employee creation or the hierarchy you provided does not allow employee creation.
409 Conflict	An employee with the same "GovernmentPersonalId" already exists.

The minimum information you have to pass in this endpoint is employee first name, last name, region (state), and branch.

JSON Format:

```
{
  "firstName": "string",
  "lastName": "string",
  "governmentPersonalId": "string",
  "bypassAddressValidationService": true,
  "street1": "string",
  "street2": "string",
  "municipality": "string",
  "region": "string",
  "branchId": 0,
  "postalCode": "string",
  "primaryEmailAddress": "string",
  "primaryPhoneNumber": "string",
  "primaryPhoneNumberCountryCallingCode": 0,
  "cellPhoneNumber": "string",
  "cellPhoneNumberCountryCallingCode": 0,
  "countryCode": 0
}
```

Example C# Request:

```
using (var handler = new HttpClientHandler())
using (var client = new HttpClient(handler) {
    BaseAddress = new Uri("https://api.ontempworks.com/")
})
{
    client.DefaultRequestHeaders.Authorization = //Add your credentions here

    var employee = new Employee()
    {
        FirstName = "John",
        LastName = "Doe",
        BranchId = 1666,
        BypassAddressValidationService = false,
        CellPhoneNumber = "8876544321",
        CellPhoneNumberCountryCallingCode = 1,
        CountryCode = null,
        GovernmentPersonalId = "722433176",
        Municipality = "Eagan",
        PostalCode = "55122",
        PrimaryEmailAddress = "example@tempworks.com",
        PrimaryPhoneNumber = "87765443212",
        PrimaryPhoneNumberCountryCallingCode = 1,
        Region = "MN",
        Street1 = "123 Main St",
        Street2 = "Suite 205"
    };


    var data = new StringContent(JsonConvert.SerializeObject(employee), Encoding.UTF8, "application/json");

    HttpResponseMessage response = await client.PostAsync("/Employees", data);
}
```

Body Parameters

- **FirstName** STRING REQUIRED
    - The first name of the employee.
    - Min length: 1 characters.
    - Max length: 50 characters.
- **LastName** STRING REQUIRED
    - The last name of the employee.
    - Min length: 1 characters.
    - Max length: 50 characters.
- **BranchId** INT REQUIRED
    - The ID of the branch the employee will be linked to.
- **BypassAddressValidationService** BOOLEAN REQUIRED
    - If true, address validation will be bypassed, and the address will save as long as all other validation passes.
    - Default value: true.
- **CellPhoneNumber** STRING OPTIONAL
    - A cell phone number.
    - Max length: 255 characters.
- **CellPhoneNumberCountryCallingCode** INT
    - The country calling code the cell phone number will be associated with.
    - Required only when the cell phone number is specified.
- **CountryCode** INT REQUIRED
    - The ID of the country.
    - Default value: Country code of the branch.
- **GovernmentPersonalId** STRING OPTIONAL
    - The government identifier of the employee.
    - (SSN)This value would be an SSN when the country code is 840 (US), NI No for 826 (UK), and SIN for 124 (Canada).
    - The country code of the request will define which is used.
- **Municipality** STRING OPTIONAL
    - City of the employee.
    - Max length: 50 characters.
- **PostalCode** STRING OPTIONAL
    - Zip code of the employee.
    - Max length: 10 characters.
- **PrimaryEmailAddress** STRING OPTIONAL
    - The primary email address for the employee.
    - Max length: 255 characters.
- **PrimaryPhoneNumber** STRING OPTIONAL
    - The primary phone number for the employee.

    - Max length: 255 characters.
- **PrimaryPhoneNumberCountryCallingCode** INT OPTIONAL
    - The country calling code the phone number will be associated with.
    - Required only when the phone number is specified.
- **Region** STRING REQUIRED
    - ID of the state.
    - Max length: 5 characters.
- **Street1** STRING OPTIONAL
    - Address of the employee.
    - Max length: 50 characters.
- **Street2** STRING OPTIONAL
    - Address2 of the employee.
    - Max length: 50 characters.

Creating an Employee Record Through Uploading a Resume

The following is the API endpoint that is used for the uploading of the resume:

- POST /Employees/Resume

The endpoint will return the following response codes:

Status Code	Reason
201 Created	The employee is created successfully. The response will include the new employee Id.
403 Forbidden	You cannot insert master table data into the system at this hierarchy level. (In Enterprise pick a Hier level that has MasterTableDataAllowed = 1 in dbo.HierType).
521 Web Server Is Down	The resume parsing service is encountering server issues.

The resume is attached using the standard file transfer approach of multi part form data.

The employee details can be passed over headers and this will override the data inside the resume.

To find out what all information can be passed in headers and what headers to use, see the headers section below.

Example with C# HttpClient:

```
using (var handler = new HttpClientHandler())
using (var multipartFormContent = new MultipartFormDataContent())
using (var client = new HttpClient(handler)
{
   BaseAddress = new Uri("https://api.ontempworks.com/")
})
{
   client.DefaultRequestHeaders.Authorization = //Add your credentions here
   client.DefaultRequestHeaders.Add("HowHeardOfId", "1");
   client.DefaultRequestHeaders.Add("FirstName", "John");
   client.DefaultRequestHeaders.Add("LastName", "Doe");
   client.DefaultRequestHeaders.Add("UseStandardizedAddressIfExists", "true");

   var filename = "resume.pdf";
   var filePath = @$"D:\{filename}";
   var fileStreamContent = new StreamContent(File.OpenRead(filePath));
   fileStreamContent.Headers.ContentType = new MediaTypeHeaderValue("application/pdf");
   multipartFormContent.Add(fileStreamContent, name: "resume", fileName: filename);

   HttpResponseMessage response = await client.PostAsync("/employees/resume", multipartFormContent);
}
```

All allowed types for uploading resume are:

- **pdf:** MediaTypeHeaderValue("application/pdf")
- **.doc:** MediaTypeHeaderValue("application/msword")
- **.docx:** MediaTypeHeaderValue("application/vnd.openxmlformats-officedocument.wordprocessingml.document")
- **.txt:** MediaTypeHeaderValue("text/plain")
- **.rtf:** MediaTypeHeaderValue("application/rtf")

Example with C# RestSharp:

```
var filename = "resume.pdf";
var filePath = @$"D:\{filename}";

var client = new RestClient("https://api.ontempworks.com/");
client.AddDefaultHeader("HowHeardOfId", "1");
client.AddDefaultHeader("FirstName", "John");
client.AddDefaultHeader("LastName", "Doe");
client.AddDefaultHeader("UseStandardizedAddressIfExists", "true");

client.Authenticator = //Add your credentions here

var request = new RestRequest("/employees/resume", Method.Post);
request.AddFile("file", filePath);

var response = await client.ExecuteAsync(request);
```

Note The maximum file size is up to 2MB.

Headers

Optionally, you can pass employee details as headers:

- **HowHeardOfId** STRING
  - ID for record.
  - Where did you hear about us information.
- **FirstName** STRING
  - Override the first name parsed from the resume.
  - The first name of the employee.
  - Min length: 1 characters.
  - Max length: 50 characters.
- **LastName** STRING
  - Override the last name parsed from the resume.
  - The last name of the employee.
  - Min length: 1 characters.
  - Max length: 50 characters.
- **UseStandardizedAddressIfExists** BOOLEAN
  - If this is set to true, TempWorks will use the standardized address who comes back from the standardization service.

Warning If set to false and address standardization is set to REQUIRED and the address can't be standardized, you will get an error: "Unable to parse an address or the address is not complete enough to be standardized. Unable to proceed as address standardization is required."

Check for Existing Employee by Email Address

The Search Email Address API endpoint retrieves a list of entities with detailed information about them.

In every result object, we have filed “OriginTypeId“. From this filed, we understand the record type. In this case, we are looking for “OriginTypeId“ equal to 1 which means "Employee". When we make the request, we can make a filter to get the result only for an employees.

More information can be found within the "Parameters" section below.

- GET /Search/EmailAddress

Example C# Request:

```
using (var handler = new HttpClientHandler())
using (var client = new HttpClient(handler)
{
   BaseAddress = new Uri("https://api-canary.ontempworks.com/")
})
{
   client.DefaultRequestHeaders.Authorization = //Add your credentions here

   var query = new Dictionary<string, string>()
   {
      ["SearchString"] = "John",
      ["OriginTypeFilter"] = "1",
      ["Skip"] = "0",
      ["Take"] = "10",
      ["OrderBy"] = "EmailAddress",
      ["OrderByAscending"] = "false"
   };

   HttpResponseMessage response = await client.GetAsync(QueryHelpers.AddQueryString("/search/emailaddress", qu
ery));
   string responseBody = await response.Content.ReadAsStringAsync();
}
```

Example Response:

```
{
 "data": [
   {
    "emailAddress": "string",
    "firstName": "string",
    "lastName": "string",
    "name": "string",
    "isPrimary": true,
    "originId": 0,
    "originTypeId": 0,
    "branchName": "string",
    "governmentPersonalId": "string"
   }
 ],
 "totalCount": 0
}
```

Parameters

- **SearchString** STRING REQUIRED
  - The string used to search.
  - Returns any matches on names or emails which start with the string provided.
- **OriginTypeFilter** INT OPTIONAL
  - The string used to search.
  - Returns any matches on names or emails which start with the string provided.
  - Filter for the entity types.
  - Pass 1 for Employee.
- **Skip** INT OPTIONAL
  - The number of items to skip.
  - Used for paging.

    - Minimum value: 0.
    - Default value: 0.
- **Take** INT OPTIONAL
    - The number of items to take.
    - Used for paging.
    - Minimum value: 1.
    - Default value: 1000.
- **OrderBy** STRING OPTIONAL
    - The column to order by.
    - Use any column provided in the results.
- **OrderByAscending** BOOLEAN OPTIONAL
    - The "OrderBy" will be ascending when true.
    - Default value: true.

Back to Top

Creating a Web Candidate Order Candidate Record

Creating candidates for Job Orders can be done through the following endpoint:

- POST /JobOrders/{id:long}/jobBoardCandidate

The endpoint will return the following response codes:

Status Code	Reason
201 Created	The specified employee was added as a candidate and the Id of the candidate is returned.
404 Not Found	The specified job order id or employee id does not exist or is not visible.
409 Conflict	The passed employee is already a candidate for that job order.

The API endpoint requires the Job Order ID, and the ID of the Employee. It also supports passing in Google Analytics parameters to track how much traffic was generated from different platforms and gauge effectiveness of your campaigns.

Note For more info on the analytics parameters, you can read Collect campaign data with custom URLs - Analytics Help.

A sample POST request might look like this:

```
// POST JobBoardCandidate
public async Task<SimpleApiResult> CreateJobOrderCandidate(long employeeId, long jobOrderId, string utmSource, s
tring utmCampaign, string utmMedium)
{
  var body = new CreateCandidate
  {
      EmployeeId = employeeId,
      UtmSource = utmSource,
      UtmCampaign = utmCampaign,
      UtmMedium = utmMedium
  };

  return await PostAsync<SimpleApiResult>($"JobOrders/{jobOrderId}/jobBoardCandidate", body);
}

private async Task<T> PostAsync<T>(string endpointUrl, object body)
{
   var result = await PostAsync(endpointUrl, body);
   return await ConvertToModel<T>(result);
}

private async Task<HttpResponseMessage> PostAsync(string endpointUrl, object body)
{
  var request = new HttpRequestMessage(HttpMethod.Post, $"{_baseUrl}{endpointUrl}");
  request.Content = new StringContent(JsonConvert.SerializeObject(body), Encoding.UTF8, "application/json");

  return await SendAsync(request);
}

private async Task<T> ConvertToModel<T>(HttpResponseMessage response)
{
    if (response.IsSuccessStatusCode)
    {
        var json = await response.Content.ReadAsStringAsync();
        var result = JsonConvert.DeserializeObject<T>(json);

        return result;
    }
    throw new Exception();
}

public class SimpleApiResult
{
   public int JobOrderCandidateId { get; set; }
}
```

The result is in the following JSON format, returning the newly created job order candidate ID if successful:

```
{
 "jobOrderCandidateId": 999
}
```

The requirements for creating job order candidates are:

- The executing service rep has to have visibility of the job order and employee being passed.
- There must not already exist a candidate with the specified job order and employee id's. If the employee already is a candidate, the API will return a 409 Conflict response.

Usually before an employee can be made a candidate, they must be vetted by an assignment restriction session and have no results, or all of the results approved. This endpoint bypasses the assignment restriction validation.

Candidates created through this endpoint are always created with a status that indicates that they applied from the web. This status is intended to indicate both how they got added as a candidate and that they may require additional vetting as they have not been run through our assignment restriction system.

Parameters

JobOrderId LONG REQUIRED

- Part of the route.
- Specifies the job order for which the candidate will be created.

EmployeeId LONG REQUIRED

- The ID of the employee.
- A list of employees can be found at search/employees.

UtmSource STRING

- Identify the advertiser, site, publication, etc. that is sending traffic to your property. For example: google, newsletter4, billboard.
- Max length: 255 characters.

UtmCampaign STRING

- The individual campaign name, slogan, promo code, etc. for a product.
- Max length: 255 characters.

UtmMedium STRING

- The advertising or marketing medium. For example: cpc, banner, email newsletter.
- Max length: 255 characters.

Back to Top

Constructing an HRCenter Application URL

HRCenter allows you to build a custom registration link with a list of predefined values. These values, passed in via query string, will be used as initial (and in some places final) values to tailor the registration process to your business needs. Examples of what can be done are auto-populate certain fields or adding the applicant as a candidate to a job order once they complete the registration process.

Register Link Format

The following is a link format example:

- https://hrcenter.ontempworks.com/{*localeId*}/{*tenantName*}?{*queryStringParameters*}

The following are the parameters within the link:

- **localeId**
  - The language HRCenter should be displayed in.
  - HRCenter only supports the English and Spanish languages.
  - Use **en** for English and **es** for Spanish.
- **tenantName**
  - The name of the HRCenter tenant.
  - Please note that this most likely is different than the Open API tenant.
- **queryStringParameters**
  - Optional query string parameters that can be passed to alter the user registration process.
  - Click here to go to the list of supported parameters.
- **Sample URL**
  - https://hrcenter.ontempworks.com/en/MyTenantName?firstName=Test&lastName=Test
  - The sample URL mentioned above would display the HRCenter website in English (**en** locale ID was provided) and pre-populate the "First Name" and "Last Name" fields in the register form.

Supported Query String Parameters

Note The field associated with the query parameter marked with an asterisk (*) may not be visible to applicants. If these fields are not visible, values provided via query string will NOT be used when the user is being registered. What fields are visible can be controlled via configurations accessible in HRCenter Admin.

If the HRCenter tenant is configured in HRCenter Admin to run a custom procedure after registration, all query string parameters will be passed to that procedure for use within the procedure.

- **branchId**
  - The ID of the branch the employee would be created in.
  - If provided, the dropdown for branch selection in the Register Page would be hidden and this value CANNOT be modified.
  - The value from this parameter would be used to add the created user to the specified branch.
- **workflowName**
  - The name of the workflow to be assigned to the applicant upon registration.
  - If provided, the dropdown for branch selection would be hidden and value CANNOT be modified in the registration page.
  - The value from this parameter would be used for registration.
- **workflowId**

    - This parameter has the same purpose and effect as **workflowName**, but expects a workflow ID.
    - If both **workflowName** and **workflowId** are provided:
        - **workflowId** would be used.
        - A check would be performed to determine if the supplied ID is valid.
        - If **workflowId** is valid and corresponds to an existing workflow, this value would be used when the user is registered.
        - If **workflowId** not is valid, **workflowName** would be used when the user is being registered.
- **username**
    - If provided, this parameter would auto-populate the "Username" field in the register form.
    - Users will be able to override the value provided for the **UserName** field by editing the field when registering.
- **emailAddress** *
    - If provided, this parameter would auto-populate the "Email Address" field in the register form.
    - Users will be able to override the value provided for "Email Address" by editing the field when registering.
- **phoneNumber** *
    - If provided, this would auto-populate the "Phone Number" field in the register form.
    - You could provide a partial number (for example the area code) or a full phone number.
    - Users will be able to override the value provided for "Phone Number" by editing the field when registering.
- **howHeardOfId** *
    - If provided, the dropdown for "How Did You Hear About Us" would be pre-populate using this value.
    - Users will be able to override the value provided for "How Did You Hear About Us" by editing the field when registering.
- **firstName**
    - If provided, this would auto-populate the "First Name" field in the register form.
    - The "First Name field" would be visible, and this value could be changed by the user who is registering.
- **middleName** *
    - If provided, this would auto-populate the "Middle Name" field in the register form.
    - Users will be able to override the value provided for "Middle Name" by editing the field when registering.
- **lastName**
    - If provided, this would auto-populate the "Last Name" field in the register form.
    - The "Last Name" field would be visible, and this value could be changed by the user who is registering.
- **postalCode** *
    - If provided, this would auto-populate the "Postal Code" field in the register form.
    - Users will be able to override the value provided for "Postal Code" by editing the field when registering.
    - The postal code is only used for geolocation and if you do not want to allow HRCenter access to your device's location.

- **employmentCategoryId** *
  - If provided, the dropdown for "Employment Category" would be pre-populate using this value.
  - Users will be able to override the value provided for "Employment Category" by editing the field when registering.
- **orders**
  - Represents a comma-joined list of job order IDs (e.g **1,2,3**).
  - These values would be used to add a user as candidate for the supplied job order(s).
  - If the user opening this link is logged in, the existing user would be used.
  - If the user opening this link is NOT logged in, the newly created user would be used after the registration is completed.

The following parameters are used to pass Google Analytics campaign data:

- **utm_source**
  - Represents the supplied job(s) source (e.g., TempWorks, Indeed).
  - This is optional and it is used if **orders** query parameter is passed, or you have a setup to run a custom procedure after registration.
  - If it is provided, this value would be saved along ALL supplied jobs.
- **utm_campaign**
  - Represents the supplied job(s) campaign (e.g., March 2022).
  - This is optional and it is used if **orders** query parameter is passed, or you have a setup to run a custom procedure after registration.
  - If it is provided, this value would be saved along ALL supplied jobs.
- **utm_medium**
  - Represents the supplied job(s) medium (e.g., email).
  - This is optional and it is used if **orders** query parameter is passed, or you have a setup to run a custom procedure after registration.
  - If it is provided, this value would be saved along ALL supplied jobs.

Back to Top