Search the knowledge bas

RECRUIT (/S/TOPIC/0TO0P000000OLNXWAI/... > ADVANCED (/S/TOPIC/0TO0P000000OLO6WAI...

API Integration Guide

The ApplicantStack API provides a way for developers to build easy, secure integrations between ApplicantStack and other systems, such as payroll systems or HRIS. The API is intended for your internal IT staff and trusted third-parties only. Your data is secured using a private API key.

Dec 27, 2019 • Knowledge

Title

API Integration Guide

Primary Product

ApplicantStack

Body

The ApplicantStack API provides a way for developers to build easy, secure integrations between ApplicantStack and other systems, such as payroll systems or HRIS. The API is intended for your internal IT staff and trusted third-parties only. Your data is secured using a private API key.
This document is designed for developers building integrations to ApplicantStack.

The API uses a REST interface with JSON as the input and the output. The API can be accessed using GET commands to read data and PUT commands to update it.

Authentication

You will need an access token and username to access the API. The access token is specific to each ApplicantStack customer. It can be found on the Settings --> Edit Settings. You will send the access token in the token parameter of the header.

```
“Token: mytoken”
```

Each access token may only be used in conjunction with its specific domain. Domain information is also available on the settings page. You will not send the domain information in a header. Instead, you will put the domain in the subdomain of the url.

```
MYDOMAIN.applicantstack.com (http://MYDOMAIN.applicantstack.com)
```

A Publisher name should be passed in as well. This helps the ApplicantStack customer to see who is making the changes to their account through the API. This should be the name of your software so that the customer will recognize it. The Publisher name is not configured anywhere, and you may use whatever you like. You will send the Publisher name through the Publisher parameter of the header.

```
“Publisher: MyName”
```

See the examples for more information.

Candidate/Application Records

To access candidate records use the following URL. This will bring you 100 candidates, including their field data. This will not include a candidate’s cover letters or resumes. The total number of pages will also be returned.

```
yourdomain.applicantstack.com/api/candidates (http://yourdomain.applicantstack.co
```

To access more records continue to make the above call with the additional getpage parameter. You can get all candidate records by repeating this process through all of the pages.

Related Articles

Setting Up the ADP Payroll Integration with ApplicantStack Onboard (/s/article/Setting-Up-the-ADP-Payroll-Integration-with-ApplicantStack-Onboard) 1.2K

Getting Started Guide (Recruit) (/s/article/Getting-Started-Guide-Recruit) 5.44K

Getting Started Guide (Onboard) (/s/article/Getting-Started-Guide-Onboard) 2.88K

Hiring Manager's Guide (Recruit) (/s/article/Hiring-Manager-s-Guide-Recruit) 2.48K

Administrator Setup Guide (Recruit) (/s/article/Administrator-Setup-Guide-Recruit) 1.68K

Trending Articles

API Integration Guide (/s/article/API-Integration-Guide)

Setup - Users (/s/article/Setup-Users)

Upgrade to Premium Texting (/s/article/Upgrade-to-Premium-Texting)

Getting Started Guide (Recruit) (/s/article/Getting-Started-Guide-Recruit)

Administrator Setup Guide (Recruit) (/s/article/Administrator-Setup-Guide-Recruit)

Contact Support

```
yourdomain.applicantstack.com/api/candidates/2 (http://yourdomain.applicantstack.
```

You may also filter your results.

```
yourdomain.applicantstack.com/api/candidates?stage=hired (http://yourdomain.appli
```

Available Candidate Filters:

```
**stage**    Receive only candidates that are at this stage
```

```
**score<**    Receive only candidates who score less than a given amount, includi
```

```
**score>**    Receive only candidates who scored higher than the given amount
```

To access an individual candidate record you will need the candidate serial. This is returned as part of the candidates objects above. This will return all of the candidate fields including the cover letter and questionnaire answers. In the example below a45a0b8q3afc is this candidate's serial number.

```
yourdomain.applicantstack.com/api/candidate/a45a0b8q3afc (http://yourdomain.appli
```

Job Records

You may access jobs in the same way you accessed candidates. This will bring you 100 jobs, including their field data. This will not include a job's listing, questionnaires, or scoring rules. The total number of pages will also be returned.

```
yourdomain.applicantstack.com/api/jobs (http://yourdomain.applicantstack.com/api/
```

A call to jobs will only return 100 results. If there are more than 100 jobs, you must call specific pages up to the total number of job pages available. Just like with candidate records, the number of jobs available will be returned with every call to jobs as NumPages:1

```
yourdomain.applicantstack.com/api/jobs/2 (http://yourdomain.applicantstack.com/ap
```

To get an individual job use the following. Note that a25a0b8ygups is the job serial number.

```
yourdomain.applicantstack.com/api/job/a25a0b8ygups (http://yourdomain.applicantst
```

Hire Records

If you subscribe to the Onboarding product you can use the api to access hires in a similar way to jobs and candidates. The URL below will bring you 100 hires, including their field data. This will not include a hire's questionnaire or form data. The total number of pages will also be returned.

```
yourdomain.applicantstack.com/o/api/hires (http://yourdomain.applicantstack.com/o
```

A call to hires will only return 100 results. If there are more than 100 hires, you must call specific pages up to the total number of pages available. Just like with candidate and job records, the number of hires available will be returned with every call as NumPages:1

```
yourdomain.applicantstack.com/o/api/hires/2 (http://yourdomain.applicantstack.com
```

To get an individual hire use the following URL. Note that e2g2xo659jvz is the hire serial number.

```
yourdomain.applicantstack.com/o/api/hire/e2g2xo659jvz (http://yourdomain.applican
```

Note that the hire and hires method URLs include '/o/' in the path to designate that they use the onboarding product.

Example

This is a sample candidates request using cURL

```
curl –H “Token: mytoken” https://mydomain.applicantstack.com/api/candidates (http
```

The following code is a working example that may be used to make a call to the API in PHP using cURL. This example is geared specifically to the candidates operation of the API. In order to get a single candidate you can set the operation to candidate and replace the code for setting the page number with code to insert the candidate serial number. The final variable $result is an object containing all of the returned data. Try `var_dump($result);` during development for a simple way to see the returned data.

```
function php_applicantstack_api(){
    $accesstoken = 'your_api_token';
    $publisher = 'ApplicantStack Jeff';

    $headers = array(
        "charset=\"utf-8\"",
        "Token: $accesstoken",
        "Publisher: $publisher "
    );
    $mydomain = "yourdomain";
    $action = 'GET';
    $operation = 'candidates';
    $pagenumber = '1';
    if($pagenumber){$operation .= '/'.$pagenumber;}

    $operationurl = "https://$ (https://$)mydomain.applicantstack.com/api/ (http:

    $curl = curl_init($operationurl);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_HTTPHEADER, $headers);

    $res = curl_exec($curl);

    if (curl_errno($curl)) {
        print "Error: " . curl_error($curl);
    }
    curl_close($curl);
    $result = json_decode($res);
}
```

PUT Commands

PUT commands allow you to update job, candidate and hire fields through the API.

You will be passing in a JSON encoded string representing an array of fields to be changed. You only need to pass fields that you wish to change. In a PUT, the serial number may be passed in the URL in the same way it is when getting a job or candidate. Alternatively, you may pass it in with the fields to change with the key ‘serial.’

To update a field for a job, candidate or hire you will need to get the name of the field and the value you wish to set. If the field you are changing is a dropdown, the value may be set by name, and a warning will be returned if the option specified was not found, and therefore not set.

Example

The sample PUT command below sends a new last name to the candidate specified by serial. Notice that the question marks are escaped. This is necessary to represent a json_encoded string properly. Also notice the publisher name, which will be noted in the candidates history.

```
curl –X PUT –H “Token: mytoken” –H “Publisher: yourname” –d {\”serial\”:\”a45a0b8
```

The following PHP example expands the previous example to include optional puts. Notice the $requestarray.

```
function php_put_applicantstack_api(){
    $accesstoken = 'your_api_token';
    $publisher = 'ApplicantStack Jeff';

    $headers = array(
        "charset=\"utf-8\"",
        "Token: $accesstoken",
        "Publisher: $publisher"
    );
    $mydomain = "yourdomain";
    $action = 'PUT';
    $operation = 'candidate';

    $operationurl = "https://$ (https://$)mydomain.applicantstack.com/api/ (http:
    $curl = curl_init($operationurl);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_HTTPHEADER, $headers);
    if($action == 'PUT'){
        $requestarray = array(
            'serial'=>'a45a0b8x7wfk',
            'source'=>'Facebook',
        );
        $request = json_encode($requestarray);
        $requestlength = strlen($request);
        $fh = fopen('php://temp', 'r+');
        fwrite($fh, $request);
        rewind($fh);
        curl_setopt($curl, CURLOPT_PUT, 1);
        curl_setopt($curl, CURLOPT_INFILESIZE, $requestlength);
        curl_setopt($curl, CURLOPT_INFILE, $fh);
    }
    $res = curl_exec($curl);

    if (curl_errno($curl)) {
        print "Error: " . curl_error($curl);
    }
    curl_close($curl);
    if($action == 'PUT'){fclose($fh);}
    $result = json_decode($res);
}
```

POST Commands

POST commands allow you to create new candidates or hires through the API. You will need to pass in a JSON encoded string representing an array of fields. When posting you will need to provide a value for all required fields for the object to be successfully created. If the field you are changing is a dropdown, the value may be set by name, and a warning will be returned if the option specified was not found, and therefore not set. Any fields set as required from the user interface will be required by the API. Any missing required fields will return an error message listing the fields that are missing.

When posting new candidates, you must also provide the "Job Serial" for the job you wish to attach the candidate to.

When posting new hires, you must also provide the "Manager" for the hire, which must match the full name of one of the users of your account.

Example

The PHP sample below will post a new candidate into the account with all required fields set.

```
function php_post_applicantstack_api(){
    $accesstoken = 'your_api_token';
    $publisher = 'ApplicantStack Jeff';
    $headers = array(
        "charset=\"utf-8\"",
        "Token: $accesstoken",
        "Publisher: $publisher "
    );
    $mydomain = "yourdomain";
    $action = 'POST';
    $operation = 'candidate';
    $operationurl = "https://$ (https://$)mydomain.applicantstack.com/api/ (http:
    $curl = curl_init($operationurl);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_HTTPHEADER, $headers);
    if($action == 'POST'){
        $requestarray = array(
            'Job Serial'=>'a2k0xovpgg1e',
            'First Name'=>'Jeff',
            'Last Name'=>'Schaefer',
            'Address 1'=>'2500 Regency Pkwy',
            'City'=>'Cary',
            'State'=>'NC',
            'Zip'=>'27518',
            'Phone'=>'919-508-6155',
            'Email'=>'jeff.schaefer@applicantstack.com (mailto:jeff.schaefer@appl
            'Source'=>'Facebook',
        );
        $requestdata = json_encode($requestarray);

        curl_setopt($curl, CURLOPT_POST, 1);
        curl_setopt($curl, CURLOPT_POSTFIELDS, $requestdata);
    }
    $res = curl_exec($curl);
    if (curl_errno($curl)) {
        print "Error: " . curl_error($curl);
    }
    curl_close($curl);
    $result = json_decode($res);
}
```

Posting Files

You may also pass in a resume or other attachment to be added to the candidate. To do this you must put your fields inside of the key 'json', and the file under the key 'file'. In php we will use the realpath function to pass the file into the POST. The example below is a copy of the example above with the file added to it. Note the difference in the assignment of $requestdata. It is also important to note that the json_encode is within the 'json' key, since the 'file' is not meant to be encoded.

```
function php_post_file_applicantstack_api(){
    $accesstoken = 'your_api_token';
    $publisher = 'ApplicantStack Jeff';
    $headers = array(
        "charset=\"utf-8\"",
        "Token: $accesstoken",
        "Publisher: $publisher "
    );
    $mydomain = "yourdomain";
    $action = 'POST';
    $operation = 'candidate';
    $operationurl = "https://$ (https://$)mydomain.applicantstack.com/api/ (http:
    $curl = curl_init($operationurl);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_HTTPHEADER, $headers);
    if($action == 'POST'){
        $requestarray = array(
            'Job Serial'=>'a2k0xovpgg1e',
            'First Name'=>'Jeff',
            'Last Name'=>'Schaefer',
            'Address 1'=>'2500 Regency Pkwy',
            'City'=>'Cary',
            'State'=>'NC',
            'Zip'=>'27518',
            'Phone'=>'919-508-6155',
            'Email'=>'jeff.schaefer@applicantstack.com (mailto:jeff.schaefer@appl
            'Source'=>'Facebook',
        );
        $requestdata = array(
            'json'=>json_encode($requestarray),
            'file'=>'@' . realpath('robots.txt'),
        );
        curl_setopt($curl, CURLOPT_POST, 1);
        curl_setopt($curl, CURLOPT_POSTFIELDS, $requestdata);
    }
    $res = curl_exec($curl);
    if (curl_errno($curl)) {
        print "Error: " . curl_error($curl);
    }
    curl_close($curl);
    $result = json_decode($res);
}
```

Responses

Once you send your request through cURL you will get a JSON encoded string response. Included in this response will be a ‘Method Result’ item that will display either success or failure. If the result is failure you should get an error item as well. If your request was a candidates or jobs GET then you will receive a list of up to 100 items in response. You will also get a numpages indicating how many pages may be accessed. If you try to reach a page out of range, the system will simply return no results. If you request a candidate or job GET you will receive more detailed information about the candidate or job requested. PUTs or POSTs will only respond with a Method Result and Errors or warnings as applicable.

Sample Candidates Response

```
stdClass Object
(
    [Method Result] => Success
    [Page] => 1
    [NumPages] => 2
    [TotalCount] => 167
    [Candidates] => Array
        (
            [0] => stdClass Object
                (
                    [Candidate Serial] => a45a0b8x7wfk
                    [Job Serial] => a25a0b8ygups
                    [Job Name] => Account Manager [SAMP-1001]
                    [First Name] => Michael
                    [Last Name] => Turner
                    [Address 1] => 3128 Queensferry Lane (Sample)
                    [Address 2] =>
                    [State] => NC
                    [City] => Raleigh
                    [Zip] => 27616
                    [Date of Birth] => 01/01/2013
                    [Phone] => (919)508-6155
                    [Email] => jeff.schaefer@applicantstack.com (mailto:jeff.scha
                    [Source] =>
                    [Rating] => Potential
                    [Disposition] =>
                    [Desired Salary] =>
                    [Date Available] =>
                    [Stage] => Do Not Pursue
                    [Score] => KO
                    [Create Date] => 02/12/2013 10:58 AM
                )
. . .

        )

)
```

Sample Candidate Response

```
stdClass Object
(
    [Method Result] => Success
    [Candidate Serial] => a45a0b8x7wfk
    [Job Serial] => a25a0b8ygups
    [Job Name] => Account Manager [SAMP-1001]
    [First Name] => Michael
    [Last Name] => Turner
    [Address 1] => 3128 Queensferry Lane (Sample)
    [Address 2] =>
    [State] => NC
    [City] => Raleigh
    [Zip] => 27616
    [Date of Birth] => 01/01/2013
    [Phone] => (919)508-6155
    [Email] => jeff.schaefer@applicantstack.com (mailto:jeff.schaefer@applicantst
    [Source] =>
    [Rating] => Potential
    [Disposition] =>
    [Desired Salary] =>
    [Date Available] =>
    [Stage] => Do Not Pursue
    [Score] => KO
    [Create Date] => 02/12/2013 10:58 AM
    [Cover Letter] =>
    [Questionnaires] => Array
        (
            [0] => stdClass Object
                (
                    [Questionnaire Name] => Accounting Manager Questions
                    [Questionnaire Serial] => a75a0b8nv9nx
                    [Submit Date] => 02/12/2013 10:58 AM
                    [Score] => KO
                    [Questions] => Array
                        (
                            [0] => stdClass Object
                                (
                                    [Question] => Are you authorized to work in t
                                    [Value] => Yes
                                )

                            [1] => stdClass Object
                                (
                                    [Question] => Do you have a bachelors degree?
                                    [Value] => Yes
                                )

                            [2] => stdClass Object
                                (
                                    [Question] => How many years experience do yo
                                    [Value] => 0-1 Years
                                )

                            [3] => stdClass Object
                                (
                                    [Question] => What minimum salary do you requ
                                    [Value] => 60k
                                )

                        )

                )

        )

)
```

Sample PUT candidate response with Warnings

```
stdClass Object
(
    [Method Result] => Success
    [Warnings] => Array
        (
            [0] => Field with name disposition does not contain option Bad Option
        )

)
```

Sample Jobs Response

```
stdClass Object
(
    [Method Result] => Success
    [Page] => 1
    [NumPages] => 1
    [TotalCount] => 7
    [Jobs] => Array
        (
            [0] => stdClass Object
                (
                    [Job Serial] => a25a0b82pemj
                    [Job Name] => Account Manager
                    [Stage] => On Hold
                    [Department] => Sales
                    [Job Category] => 4 Sales
                    [Job Type] => Exempt (salaried)
                    [Campaign Start Date] => 02/12/2013
                    [Salary Range] => 40k-60k

                    [Location] => Cary, NC
                    [Create Date] => 02/12/2013 01:15 PM
                )
. . .
        )

)
```

Sample Job Response

```
stdClass Object
(
    [Method Result] => Success
    [Job Serial] => a25a0b82pemj
    [Job Name] => Account Manager
    [Stage] => On Hold
    [Job Listing] =>
    [Department] => Sales
    [Job Category] => 4 Sales
    [Job Type] => Exempt (salaried)
    [Campaign Start Date] => 02/12/2013
    [Salary Range] => 40k-60k
    [Manual Labor Involved] => boxing
    [Location] => Cary,NC
    [Create Date] => 02/12/2013 01:15 PM
)
```

Sample PUT Job Response

```
stdClass Object
(
    [Method Result] => Success
)
```

Sample PUT Job Response with error

```
stdClass Object
(
    [Method Result] => Failure
    [Error Message] => No job found with serial given
)
```

Stage Change Notifications

With the stage change notification mechanism you can develop real-time systems that work with your ApplicantStack data. For example, you could develop a method that looks for Hired candidates and then loads their information into your internal or third-party HRIS or Payroll system.

You can subscribe to stage change events by entering an endpoint URL in Setup->Settings. When you are subscribed to stage change events, the system will post to your endpoint each time a candidate or hire (in Onboarding) is moved to a new stage. The post will include the name of the new stage, the candidate or hire serial, the user who initiated the stage change (if applicable), and the date of the stage change.

If a user initiated the stage change, the "User Serial" field will contain the ID of the user and the "User Type" field will be "USER". If the stage change was initiated by the system, the "User Serial" field will be blank and the "User Type" field will be "SYSTEM".

For performance reasons, the stage change notification will only include summary information, not the full candidate or hire record. Typically when you receive the notification you would then use the candidate or hire serial to pull the full candidate or hire information from the system.

The stage change notification will contain a "secret" value that is unique for your ApplicantStack account. You can specify this secret in Setup->Settings or have the system generate it for you. The secret is sent with all notification posts so you know that they originated from ApplicantStack.

Sample Stage Change Notification

```
stdClass Object
(
    [Object Type] => candidate
    [Event Type] => stagechange
    [Domain] => mycompany
    [Secret] => m58hqrje1xx5w4ku
    [Event Date] => 2016-09-08T15:58:13-04:00
    [Stage] => Interview
    [Candidate Serial] => a4g2xo6ji234
    [User Serial] => a1g2xo6xoezu
    [User Type] => USER
)
```

Errors

Request type error while fetching data Indicates that there was a problem getting the GET or PUT data. This often occurs with PUTs when characters are not escaped properly.

No Job/Candidate found with serial given Indicates that the serial number you passed in did not match an object within the account.

Job/Candidate does not belong to this account Indicates that the serial number refers to a job that this account does not have access to.

Error while writing job/candidate record This is an internal error. Please contact ApplicantStack.

Candidate job does not exist

Warnings

Field with name NAME does not contain option OPTION. The option that you passed in is not available for this field. Usually this is caused by a spelling error.

Field with name NAME was not found. The field you are trying to change does not exist. Check your spelling.

Notes

In the php examples above we use json_decode($result) and we get a stdClass object. It is often preferable to use the optional second parameter json_decode($result,true) to get an associative array instead.

More Examples

.NET 4.0 Example (https://help.applicantstack.com/hc/en-us/articles/115000786233--NET-4-0-API-Example)

URL Name

API-Integration-Guide

Last Published Date

12/27/2019 9:45 PM

Advanced
(/s/topic/0TO0P000000olO6WAI/a...

Privacy Policy (https://partner.swipeclock.info)