



Paycom REST API Companion Guide

HELP MENU MANUAL

TABLE OF CONTENTS

	Page Numbers
Table of Contents	<u>1</u>
Notes and Things to Consider.....	<u>5</u>
How to Connect.....	<u>5</u>
Retrieve Paginated Results.....	<u>6</u>
Sample Workflow.....	<u>7</u>
Error Handling.....	<u>13</u>
Maintaining API Users.....	<u>14</u>
Support.....	<u>18</u>



Employee Endpoints Group

Employee Lists

Route	Description
api/v1/employeeid	Return a list of employee identifiers.
api/v1/employeeirectory	Return a paginated list of employees and their demographics.
api/v1/employeeenewhire	Return a list of employees added in a specific date range.

Non-Sensitive Employee Information

Route	Description
api/v1/employee/:eocode	Return an employee master record.
api/v1/employee/:eocode/customfield	Return custom field information for an employee.
api/v1.1/employee/:eocode/change	Return an audit log of updates made to non-sensitive employee fields for a specified date range.
api/v1/employee/:eocode/photo	Return an encoded string representing the employee photo.
api/v1/employee/:eocode/tax	Return employee tax setup from Form 1.
api/v1/employee/:eocode/ratesbyallocation	Return employee rates by allocation information from Form 10.

Sensitive Employee Information

Route	Description
api/v1/employee/:eocode/sensitive	Return an employee master record along with sensitive data.
api/v1.1/employee/:eocode/sensitivechange	Return an audit log of updates made to employee fields for a specified date range along with sensitive changes.



Time and Attendance Endpoint Group

Route	Description
api/v1.1/punchimport	Add, edit, or delete time clock events to employee time cards
api/v1/employee/:eencode/punchaudit	Return historical information about punches for a specified date range.
api/v1/employee/:eencode/punchhistory	Return a list of employee punches.

New Hire Endpoint Group

Route	Description
api/v1/newhireids	Return a list of IDs of New Hires within a given time range.
api/v1/newhire/:new-hire-id	Returns the New Hires information for the given new hire id.
api/v1/newhire/:new-hire-id/customfield	Return custom field information for new hires.
api/v1/newhire/:new-hire-id/photo	Return an encoded string representing the new hire photo.

Position Management Endpoint Group

Route	Description
api/v1/positions/detail	Get a list of all position codes or a position code can be provided as a parameter to receive the details about the specific position.
api/v1/positions/levels	Get a list of all position level codes or a position level code can be provided as a parameter to receive the details about the specific position level.
api/v1/positions/seats	Get a list of all position seat numbers or a position seat number can be provided as a parameter to receive the details about the specific position seat.



Client Endpoint Group

Locations and Establishments

Route	Description
api/v1/cl/locations	Returns a list of all, or specific Company Locations configured for the client.
api/v1/cl/establishments	Returns a list of all, or specific Company Establishments configured for the client.

Labor Allocations

Route	Description
api/v1/cl/category	Return a list of all labor allocation categories or one if category is specified
api/v1/cl/category/:catcode/detail	Return information for one or all distributions for a specified labor allocation category. Add new distributions or update existing distributions.

Misc

Route	Description
api/v1/cl/earning	Return information about earnings.



Notes and Things to Consider

It is recommended to connect to the Paycom API for the first time using a testing suite such as Postman to allow you to manually step through the Paycom API. It will give you an idea of how the requests should look, and what responses you will receive.

In an effort to provide the best quality of service, some methods have access limitations, either in the form of timeouts between subsequent method calls or the limit of the number of calls that can be made in a 24-hour period. You should receive an error indicating if a limitation has been reached.

If you have any questions or are running into any problems, please contact the Implementation API Team.

How to Connect:

To begin with the API, you will receive an SID and Token from the Automation team. They will also need any public IP addresses you plan to access the API from.

<https://www.paycomonline.net/v4/rest/index.php>, this URL will be used to connect to the REST API. A header using basic authentication will also be necessary. The SID and Token will be used as the username and password respectively for authentication. You then base64 encode these strings separated by a colon ":", which allows for connection to the API. Paycom will also require any WAN IP addresses you will be reaching out to the API from for allowlisting.

Below is an example cURL request for the employeeid call.

```
curl -X GET \
  https://www.paycomonline.net/v4/rest/index.php/api/v1/employeeid \
  -H 'Authorization: Basic **Base64 encoded sid:token here **' \
  -H 'Cache-Control: no-cache' \
  -H 'Content-Type: application/json' \
```

Retrieve Paginated Results

If pulling all employees, your results will more than likely return a partial result (status code 206 Partial Content) due to pagination. If this occurs, there are two options for full results retrieval.



The first option is to increase the results per page using the page size parameter, “pagesize”. Below is a sample URL for the employee directory call:

<https://api.paycomonline.net/v4/rest/index.php/api/v1/employeedirectory?pagesize=500>

Other useful headers returned are 'X-Max-Page-Size', which can be referenced for the max page size parameter that can be requested and 'X-Total-Count', which returns the total number of records.

The second option would be to reference the Link header where the next relative page URL as well as the previous page and first and last pages can be found. The Link header will look like the below example.

```
'Link':  
'<https://api.paycomonline.net/v4/rest/index.php/api/v1/employeedirectory?requestid=363f31f373e  
d912e073006bde24e7fc3&page=2>; rel="prev",  
https://api.paycomonline.net/v4/rest/index.php/api/v1/employeedirectory?requestid=363f31f373e  
d912e073006bde24e7fc3&page=1>; rel="first",  
<https://api.paycomonline.net/v4/rest/index.php/api/v1/employeedirectory?requestid=363f31f373e  
d912e073006bde24e7fc3&page=3>; rel="last",'
```



API Sample Workflow

Retrieve List of employees

<https://api.paycomonline.net/v4/rest/index.php/api/v1/employeeedirectory/>

This will return all employees in a JSON format, below is an example of an employee that is returned.

```
{ "result": true,
  "data": [
    {
      "eecode": "A001",
      "eename": "GREEN, ALEX",
      "firstname": "ALEX",
      "lastname": "GREEN",
      "gender": "1",
      "streetaddr": "4327 MAIN STREET",
      "cityaddr": "FT WORTH",
      "clockseq": "8765432",
      "eebadge": "140831",
      "zipcode": "55555",
      "homestate": "TX",
      "homephone": "",
      "eestatus": "A",
      "deptcode": "99900",
      "deptdesc": "heather test",
      "cat1": "1",
      "cat1desc": "1",
      "cat2": "CanbeAlpha1",
      "cat2desc": "state4",
      "cat3": "003",
      "cat3desc": "NewCode",
    }
  ],
  "errors": [],
  "errorCount": 0,
  "records": 1
}
```

Retrieve Employee Changes



<https://api.paycomonline.net/v4/rest/index.php/api/v1/employee/A002/change/>

For parameters such as time you will use a get parameter on the URL such as ?parameter=value This would bring the full employee change URL to:

<https://api.paycomonline.net/v4/rest/index.php/api/v1/employee/A002/change?startdate=UNIXTimeStamp&enddate=UNIXTimeStamp>

This is an example of pulling the changes for an employee with an ecode of A002. The ecode is a mandatory field as denoted in the endpoint documentation. Below is a sample JSON result of what Employee Change may return.

```
{
  "result": true,
  "data": [
    {
      "changedby": "hhtest",
      "changedesc": "Position Mgmt: Position Seat",
      "changetime": "2019-12-02T10:19:21-06:00",
      "changetype": "U",
      "childcode": "",
      "clockseq": "8765432",
      "ecode": "A001",
      "ename": "GREEN, ALEX",
      "new_value": "Sales Representative-3",
      "notes": "",
      "old_value": "CEO",
      "usetype": 1
    },
    {
      "changedby": "hhtest",
      "changedesc": "Position Mgmt: Position Family",
      "changetime": "2019-12-02T10:19:21-06:00",
      "changetype": "U",
      "childcode": "",
      "clockseq": "8765432",
      "ecode": "A001",
      "ename": "GREEN, ALEX",
      "new_value": "9128",
      "notes": "",
      "old_value": "9293",
      "usetype": 1
    }
  ]
}
```




```
],  
  "errors": [],  
  "errorCount": 0,  
  "records": 8  
}
```



Import Labor Allocation Codes

<https://api.paycomonline.net/v4/rest/index.php/api/v1/cl/category/:catcode/detail>

To PUT a new detail code into an existing labor allocation category or to PATCH an existing detail code you only need a simple body in combination with the above URL.

Consider you want to import the new Job “Waiter”, code 1234, into Paycom. If Job is your category 2 in Paycom the full URL will look like this:

<https://api.paycomonline.net/v4/rest/index.php/api/v1/cl/category/2/detail>

Here is the Body you would PUT to the above URL:

```
{
  "hideonline": 0,
  "detailcode": "1234",
  "detaildesc": "Waiter",
  "glcode": "1234"
}
```

Now if you would like to PATCH the existing “Waiter” detail description to “Waiter OLD”, you would send a PATCH to the same URL but including the detail code to be PATCHed:

<https://api.paycomonline.net/v4/rest/index.php/api/v1/cl/category/2/detail/1234>

Here is the Body you would PATCH to the above URL:

```
{
  "hideonline": 0,
  "detailcode": "1234",
  "detaildesc": "Waiter OLD",
  "glcode": "1234"
}
```

Now the detail code 1234 in category 2 has a detail description of “Waiter OLD”



Import Time

<https://api.paycomonline.net/v4/rest/index.php /api/v1.1/punchimport>

To POST a new punch, hours event, or earnings to the timecard use the above URL.

Consider you want add a series of events to the timecard on June 1st 2020. Employee A001 punches in for the day to department 1234 at 8:00 am CST and out for the day at 5:00 CST. Employee B001 records 5 hours of Regular hours and records \$30 of tips. You could add this to the timecard using the following URL and Body combination:

<https://api.paycomonline.net/v4/rest/index.php /api/v1.1/punchimport>

Here is the Body you would POST to the above URL:

```
[
  {
    "eocode": "A001",
    "deptcode": "1234",
    "entrytype": 1,
    "punchtime": "1591016400",
    "punchtype": "ID",
    "timezone": "CST"
  },
  {
    "eocode": "A001",
    "entrytype": 1,
    "punchtime": "1591048800",
    "punchtype": "OD",
    "timezone": "CST"
  },
  {
    "eocode": "B001",
    "entrytype": 2,
    "hours": 5.0,
    "earncode": "R",
    "punchtime": "1590987600",
    "timezone": "CST"
  },
  {
    "eocode": "B001",
    "entrytype": 3,
    "dollaramount": 30,
  }
]
```



```
        "earncode":"TP1",  
        "punchtime":"1590987600",  
        "timezone":"CST"  
    }  
]
```

The values of fields such as Earning Code, Labor Allocation, and Department are unique per client in Paycom. Please reach out to your Paycom Specialist to determine what those unique values are. It is best practice when sending time data into Paycom to only include the fields that are required and that are being used to allocate the time event.

When using the V1.1 version of the punch import method, the response will include a Paycom assigned punchId for each punch item. It is best practice to store these punchIds on your end for future mapping or troubleshooting needs. It is also best practice to store a record of if the punch has been sent to Paycom or not. In the event a request to the endpoint fails, all punches in the request will be considered to have failed and not written to the timecard. A revision will need to be made and the entire batch will need to be resent.

Sample response for api/v1.1/punchimport:

```
{  
  "result": true,  
  "data": [  
    {  
      "punchId": 75323044,  
      ...  
      "makeUpTime": false,  
      "externalId": null  
    }  
  ],  
  "errors": [],  
  "errorCount": 0,  
  "records": 1  
}
```



Error Handling

Error handling should be included in your process. There could be times when, for unforeseen reasons, something may go wrong. In any given response, Paycom includes the result object. If there were any errors with the call itself, this object will be set to false, otherwise true if the call was successful.

Any errors that occurred will be returned as response codes, please check with the End Point Documentation for the specific error response codes based on what end point you are using.

Note: A response code of 200 for GET, and 201 for PUT or POST indicates a successful call of the API.

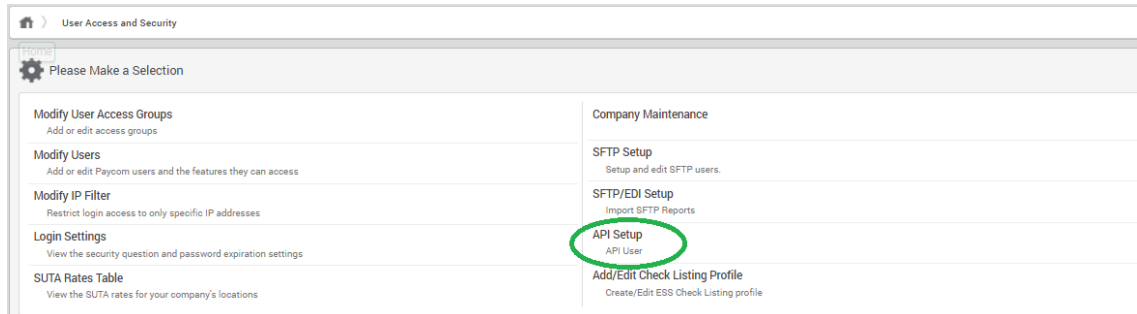
Updating and Maintaining Your API Users

Within Permission Profiles, ensure you have a profile for your IT personnel, and then grant them access to the API Setup screen. From the API tab, check the box for “API Setup” and determine whether users can create and edit permissions or access sensitive functions.

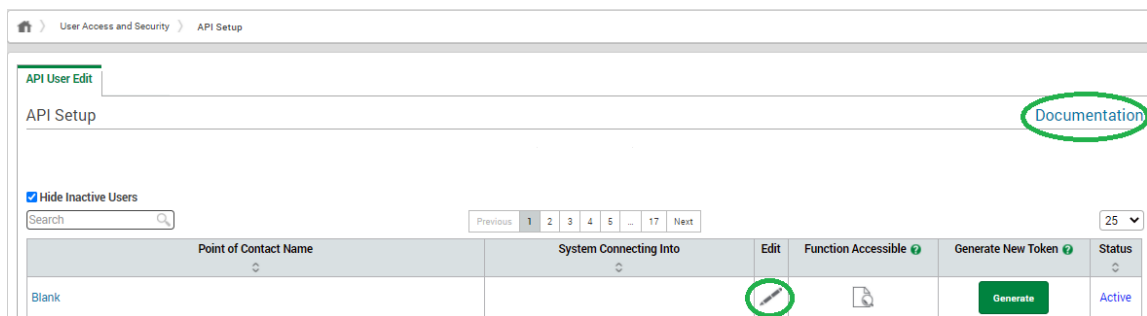
If you do not see this option available, please contact your dedicated specialist to request access

ACA, Benefits Administration and COBRA	Notification Center	SFTP	API	Sensitive Information	Labor Allocation	SSO
Employees	API Setup ? <input checked="" type="checkbox"/>					
Human Resources	Edit User Permissions ? <input checked="" type="checkbox"/>					
Payroll	Access to Sensitive Functions ? <input type="checkbox"/>					
Reports						
Talent Acquisition						
Talent Management						
Time Management						
User Options						

After your users have access, navigate to the API Setup screen by clicking “API Setup” from the User Access and Security menu.



On the API Setup screen, the details for each API user will display. If you need a new API user added, please reach out to your specialist.



To edit your API user click the pencil icon under Edit.



On the API Edit User screen, enter details such as the point of contact's name, description and email address. Then, check the appropriate boxes in the Functions section to determine which functions the user can access. For sensitive functions to display, "Access to Sensitive Functions" must be enabled in Permission Profiles. You can also generate a new token on this screen. When finished, click "Update"

User Access and Security > API Setup > API Edit User

API Edit User

* Indicates Required Field

Point of Contact Name *

Test

Point of Contact Description ?

Point of Contact Email *

automation@paycomonline.com

System Connecting Into ?

Active☒

Functions ?

Search

☐ Select All

☒ REST-GET NEW HIRE PHOTO

☒ REST-GET EMPLOYEE IDS

☒ REST-GET EMPLOYEE PUNCH AUDIT

☒ REST-PUNCH IMPORT

☒ REST-PUNCH PATCH

☒ REST-PUNCH DELETE

☐ REST-CLOCKED IN/OUT STATUS

☒ REST-GET EMPLOYEE DIRECTORY

☒ REST-GET EMPLOYEE CUSTOM FIELDS

Show Selected Filters

IP Address

Add

10.0.0.1/32

Remove

SID

003868abe357b0325445b79f133331e207d7d033b891ab2f26507b12fb2de81e

Copy

Token

0337dd2ea4330e7ef877cd4002f9423bebbfb3bd0a4257ad061831e2d5db0784

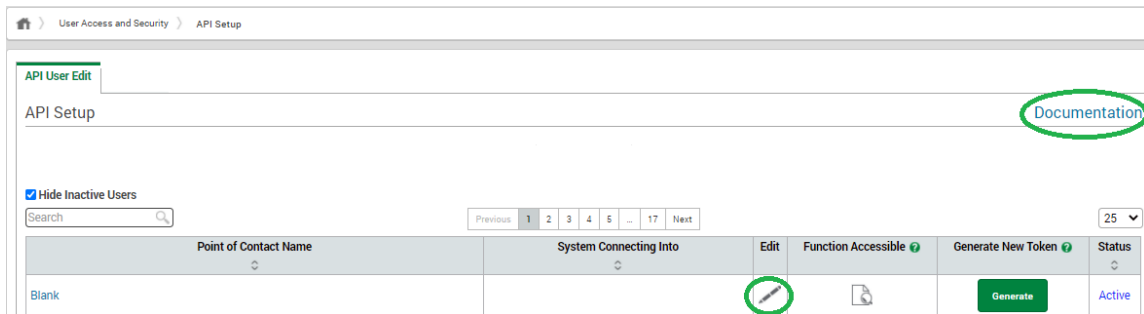
Copy

Generate Token

Update

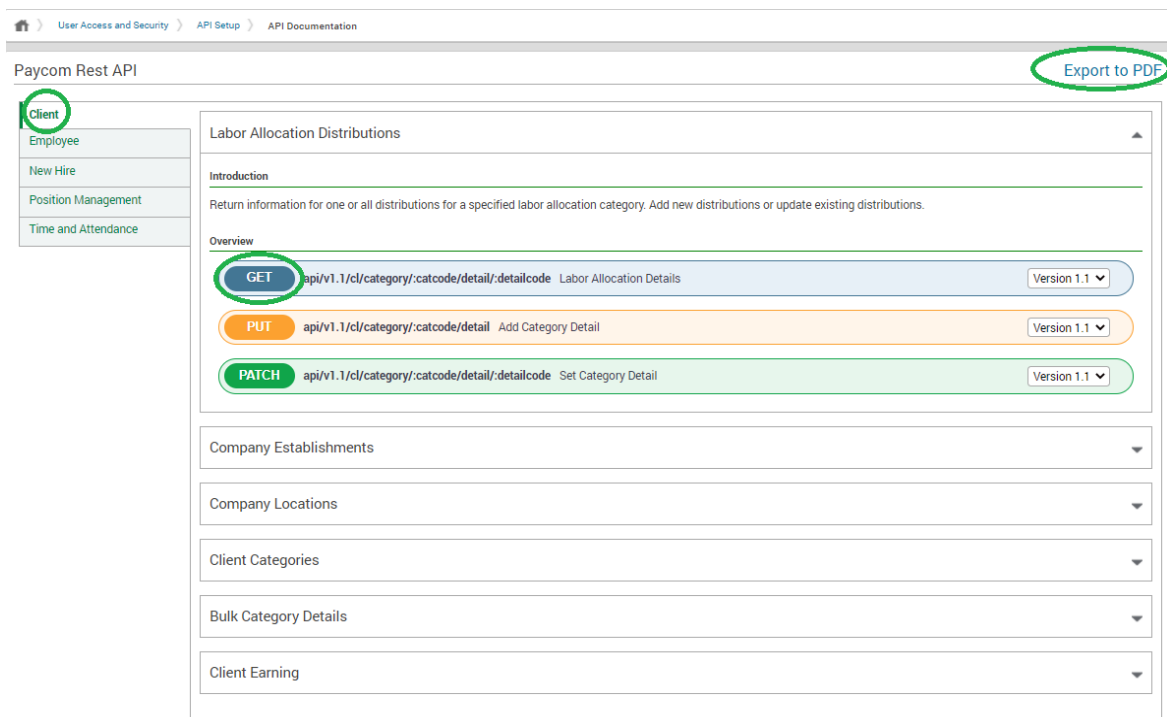


To view your API documentation, click “Documentation.”



On the API Documentation screen, use the tabs on the left side of the screen to view Client, Employee, New Hire, Position Management, and Time and Attendance documentation. To review the details for a specific type of documentation, click the arrow to expand the section and click “GET.”

To export all API documentation, click “Export to PDF.”





Support

The Technical Solutions Team can assist with any API issues or questions you might have. They can be reached by email at automation@paycomonline.com. When sending an email please include your client code or company name and as much information as possible.