theEMPLOYEEapp®

theEMPLOYEEapp API V3

Updated May 2023

Overview

The following documentation will cover the essentials of theEMPLOYEEapp V3 API. The API provides a set of functions and features to allow client interaction programmatically with theEMPLOYEEapp platform. In version 3 of the API, the platform has the capability for:

- CSV Uploads
- User Management
- Directory Contact Management
- Group Management

Endpoints

Available endpoints to interact with V3 API are:

Environment	Route
Production (US)	https://api.theemployeeapp.com/v3/
Production (EU)	https://api.eu-west-2.theemployeeapp.com/v3/
Beta	https://api.beta.theemployeeapp.com/v3/

CSV Uploads

Uploads via CSV files are one of the many ways to automate user management for your organization. The following features have been introduced in v3 CSV Uploads:

- Selective columns for Users and Groups
    - e.g. Users and Groups does not have to be enforced in a particular column. They can be set through a flag in the API call.
- Ability to for aggregated groups
    - e.g. Groups A, B, C can now be aggregated into a consolidated 4th group called: Group A-B-C. When publishing content, and you select group A-B-C, it will only render to those that are in all 3 groups (User must be in groups A and B and C, not A or B or C).

Method	Route
POST	/users/bulk-csv

Example Route: https://api.theemployeeapp.com/v3/users/bulk-csv

Request Body

Name	DataType	Description	Example
headerGroupNames *(optional for 2 column CSV, required for multi-column CSV)*	string	The names of the column that represents the group(s). Spaced with a single comma for each name.	Location, Department, Job Role
headerRegCodeName *(optional)*	string	The name of the column header that represents the unique identifier. If not set, the values in the first column of the CSV is used as the unique identifier.	EmployeeID
dontRemoveMembers *(optional)*	boolean	Whether or not to allow the csv job to remove any members if they no longer exist in the uploaded CSV. This applies to removing more than 10% of the users and is a safety mechanism to prevent mass removal of users unintentionally.. true - Do not allow removal of more than 10% of users false - Allow removal of more than 10% of users. *Note: To proceed with removal of more than 10% of your users, please contact your client success account representative to guide you through the steps and configuration for this process.*	true/false
csvType (optional)	string	Type of csv being uploaded	group
file (required)	csv file	csv file of members to upload.	Example.csv

headerUnion (optional)	string	The headerUnion flag allows for multiple groups to be combined into a new group. For example if the headers in the csv are: Location,Department. If the admin would like to create a new grouping of all their Locations and Departments such as New York and Sales, London and Sales, simply enter Location,Department. If you would like another header merge, please continue by separation of a comma. For Example, Location-Department,Department-SubDivision. These groupings will only be applied to users that are part of all groups. In the example above, the users would need to be part of both Location AND Department for them to be in the respective grouping. If the user was only in one group OR another, they would not be added to the grouping. In addition, this grouping is only applied to the users in the current CSV. Any user added to Location via the CMS will not be included in grouping, but the groupings will be selectable as groups to add a user to in the CMS post CSV upload.	Single: Location,Department This will result in all groups under the headers of **Location and Department** to be respectively merged. Multiple: Location-Department,Department-SubDivision
groupUnion (optional)		This is a more low level control of which groupings are created. If you did not want every single group under a header to be created as a grouping, you may individually select each group. For example, if you wanted New York and London, but they are both under location's header, here is your chance to declare New York-Sales Continue adding by separation of a comma. Such as	New York-Sales, London-Sales

		New York-Sales,London-Sales	
profile (optional)		This option is for accounts with multiple profiles. By using this field, the uploader of the csv may choose which profile they wish to exclusively set the csv data to. This is also the way to migrate groups from profile to profile. The entry of the profile name is case, character and space sensitive. Be sure to enter the exact name of the profile.	Profile-A B
partialUpdate (optional)		This feature is primarily designed for backwards compatibility with our CSV V2 Processing mode. It allows for multiple CSVs to be uploaded for a given account without blocking updates that trigger a 10% threshold of users being removed from the system. To achieve this, users need to be properly partitioned into groups that should not overlap significantly with other groups from the other CSVs. When this feature is enabled, the request will process changes only for the groups listed in the CSV with the users on that CSV. If users are expected to be in a particular group listed in the CSV, the users also need to be listed on that CSV otherwise they will be removed from the group and from the system (if the users are not in other groups on the CSV). For example: CSV 1 has groups A, B and C, CSV 2 should only have D,E, and F to prevent collisions and errors when processing CSVs with partialUpdate enabled.	true/false

headerMobileNumber *(optional)*	string	The name of the column header that represents the mobile number to receive SMS for that user. The number needs to be in E.164 format for US based numbers. The number would start with +1 (for US country code), then 3 digits for the area code, then 7 digits for the phone number with no spaces or symbols between the numbers. Example: +15557778888 +12128961200	MobileNumber

Request Parameters

Name	DataType	Description	Example
code (required)	string	API Key (Grant Code)	-

Responses

HTTP Code	Description	Schema
200	Success	jobId (String)
405	Invalid Input	No Content
Default	Error	Error

Status

Status for uploaded CSV Jobs can be obtained by running a request against the endpoints below:

Method	Route
GET	/jobs

Example Route: https://api.theemployeeapp.com/v3/jobs

Request Parameters

Name	DataType	Description	Example
code (required)	string	API Key (Grant Code)	-
limit (required)	string	Limit number of results to return	100
offset (required)	string	Number of results to skip	0
csvType (required)	string	Type of CSV uploaded	GROUP DIRECTORY

Responses

HTTP Code	Description	Schema
200	Success	result (String)
405	Invalid Input	No Content

Default	Error	Error

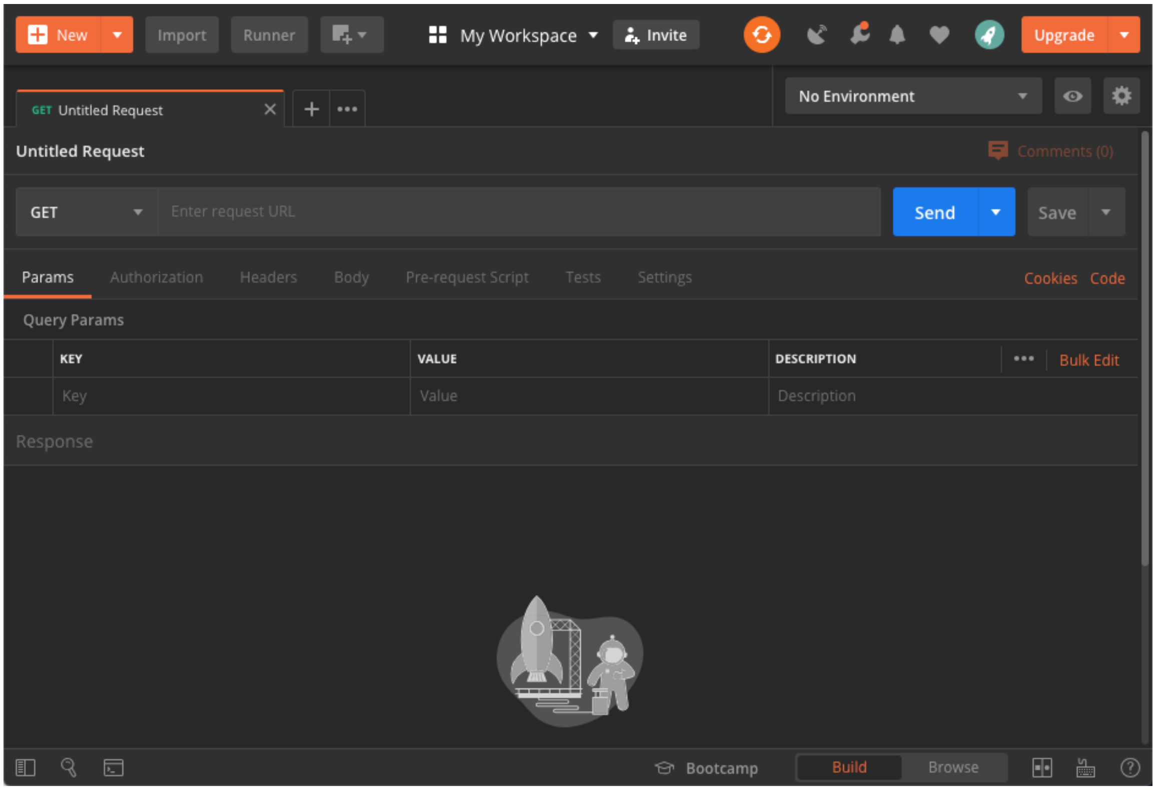
Example Usage

Double Column CSV

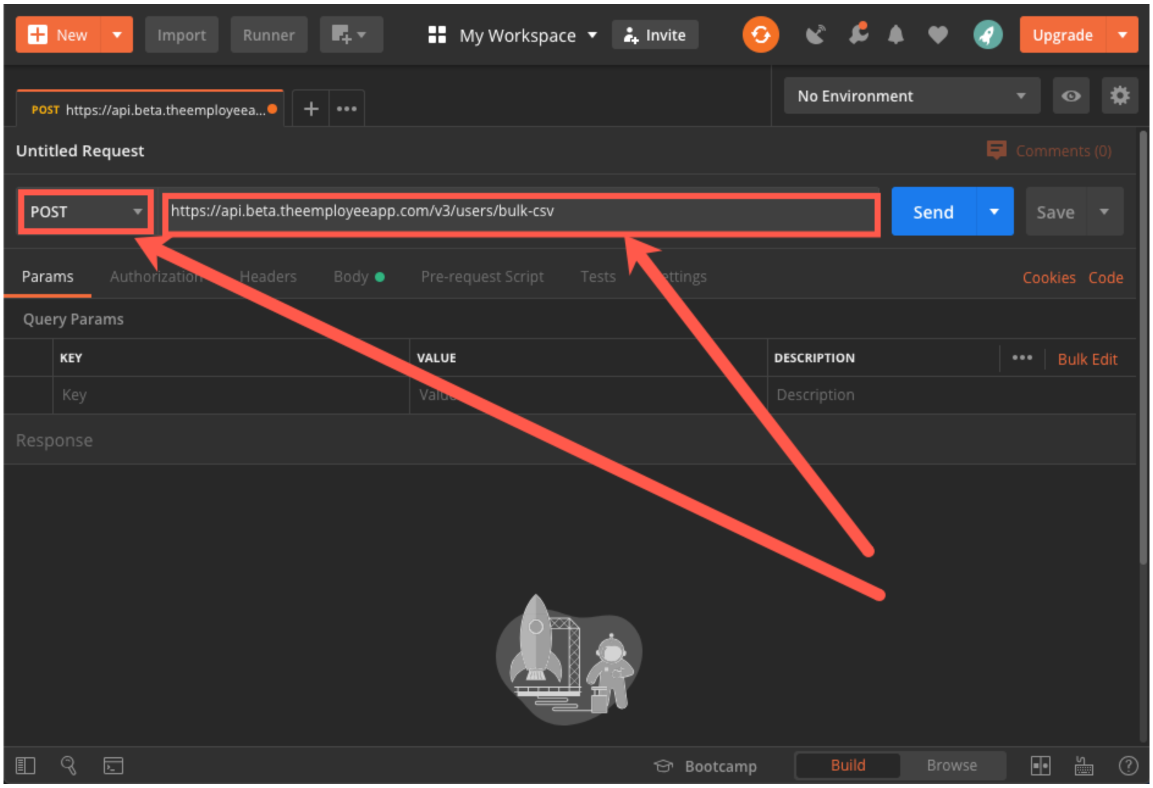
1. Create sample csv file with the following users and headers (regcode, groups):
   Note: Make sure to include Account Admin is in the CSV

```
regcode,department
Admin_DunderMifflin,Management
Michael_Scott,Management
Dwight_Schrute,Administrative
Pam_Beesly,Administrative
Jim_Halpert,Sales
Stanley_Hudson,Sales
Phyllis_Lapin,Sales
Angela_Martin,Accounting
Oscar_Gutierrez,Accounting
Toby_Flenderson,HR
```

2. Open Postman



3. Set the request type to POST and enter the API endpoint:
    a. Production: https://api.theemployeeapp.com/v3/users/bulk-csv
    b. Beta: https://api.beta.theemployeeapp.com/v3/users/bulk-csv

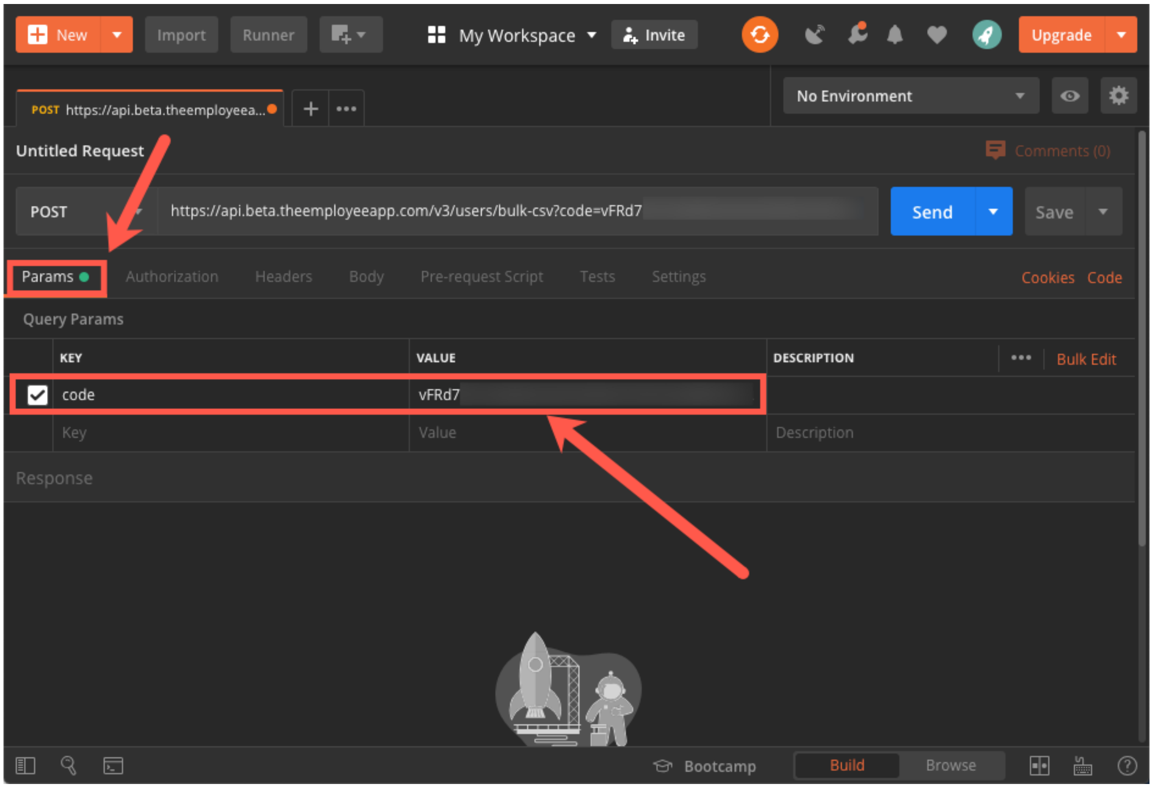


Untitled Request

POST https://api.beta.theemployeeapp.com/v3/users/bulk-csv

4. Ensure the following parameters are set:

   code: [Contact customercare@theemployeeapp.com to obtain a grant code]



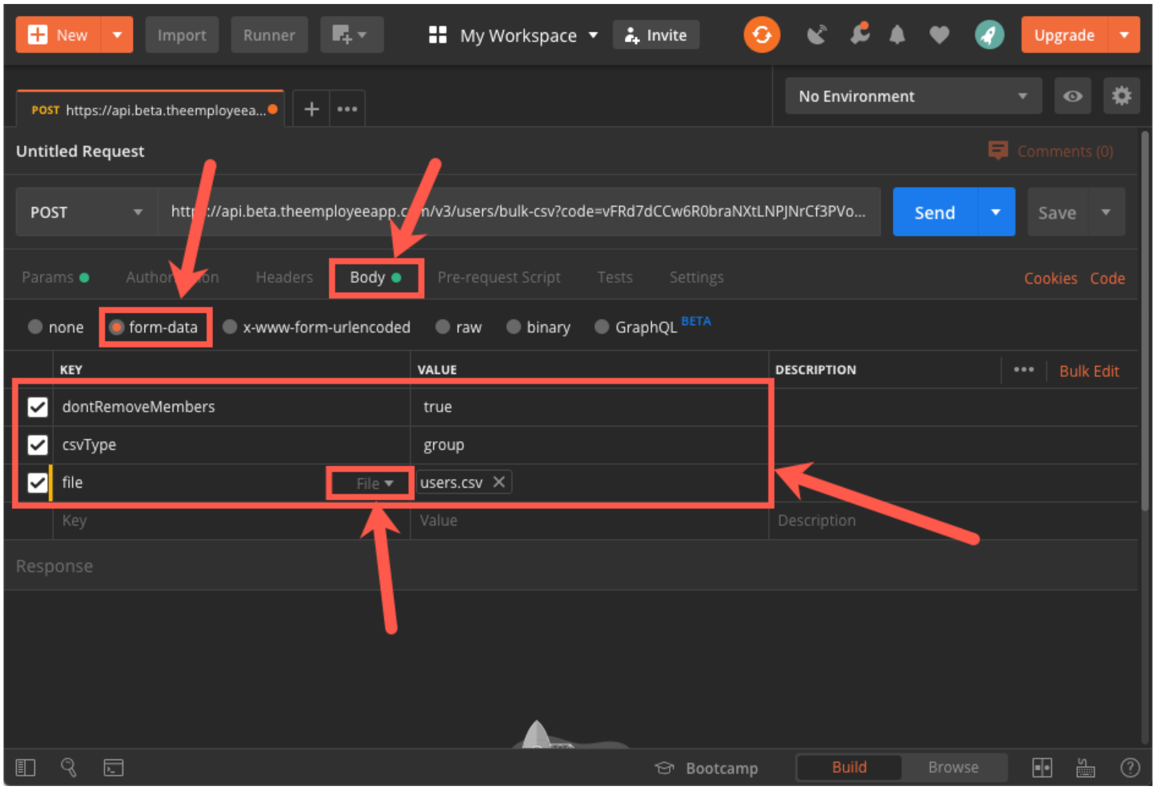
Untitled Request

POST https://api.beta.theemployeeapp.com/v3/users/bulk-csv?code=vFRd7[redacted]

Query Params

	KEY	VALUE	DESCRIPTION
☑	code	vFRd7[redacted]	
	Key	Value	Description

5. Select the file by clicking on **Body**. Then select form-data as the Body type. The following keys and values need to be added:
    a. dontRemoveMembers: true
    b. csvType: group
    c. file - Select dropdown for Key type to be File, then **Select File** from local machine

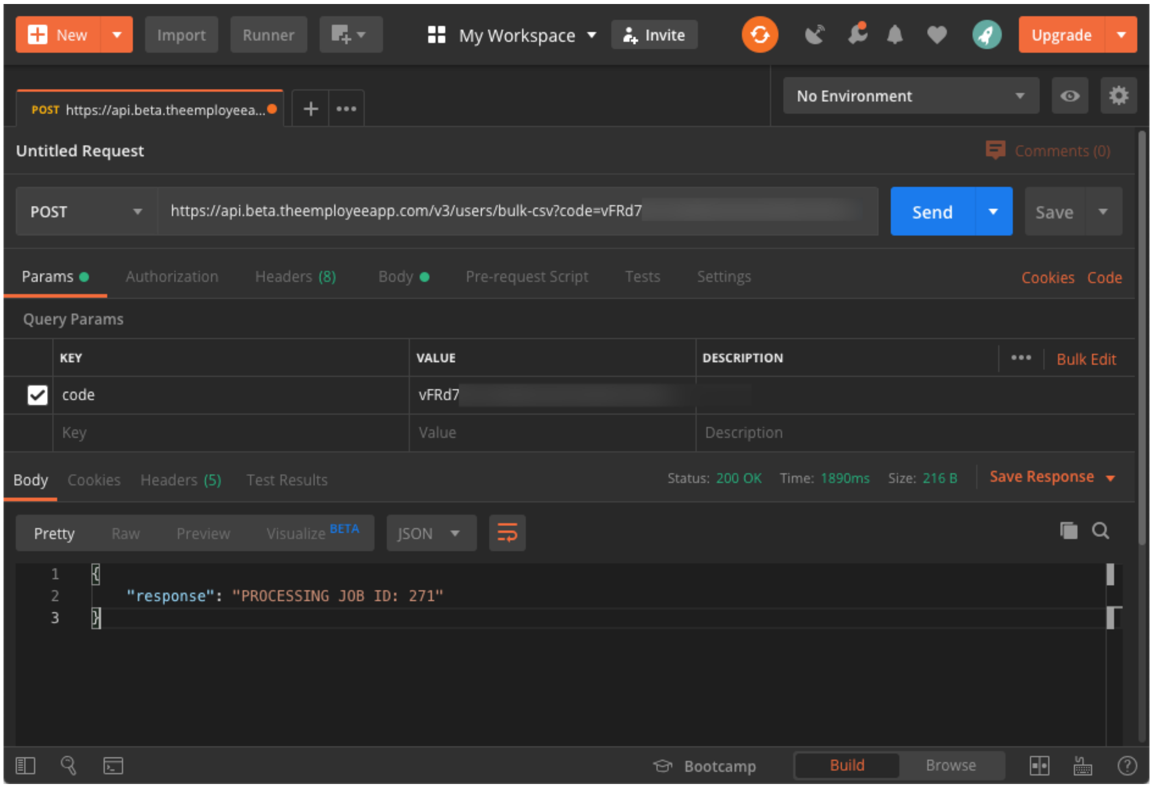


Untitled Request

POST http ://api.beta.theemployeeapp.c n/v3/users/bulk-csv?code=vFRd7dCCw6R0braNXtLNPJNrCf3PVo...

	KEY	VALUE	DESCRIPTION
☑	dontRemoveMembers	true	
☑	csvType	group	
☑	file File ▾	users.csv ✕	
	Key	Value	Description

6. Once all the information is filled out, press Send. A successful response should be returned with the Processing Job ID.



POST https://api.beta.theemployeea...

Untitled Request

POST https://api.beta.theemployeeapp.com/v3/users/bulk-csv?code=vFRd7[redacted]

Query Params

	KEY	VALUE	DESCRIPTION
☑	code	vFRd7[redacted]	
	Key	Value	Description

Status: 200 OK Time: 1890ms Size: 216 B

```
{
    "response": "PROCESSING JOB ID: 271"
}
```

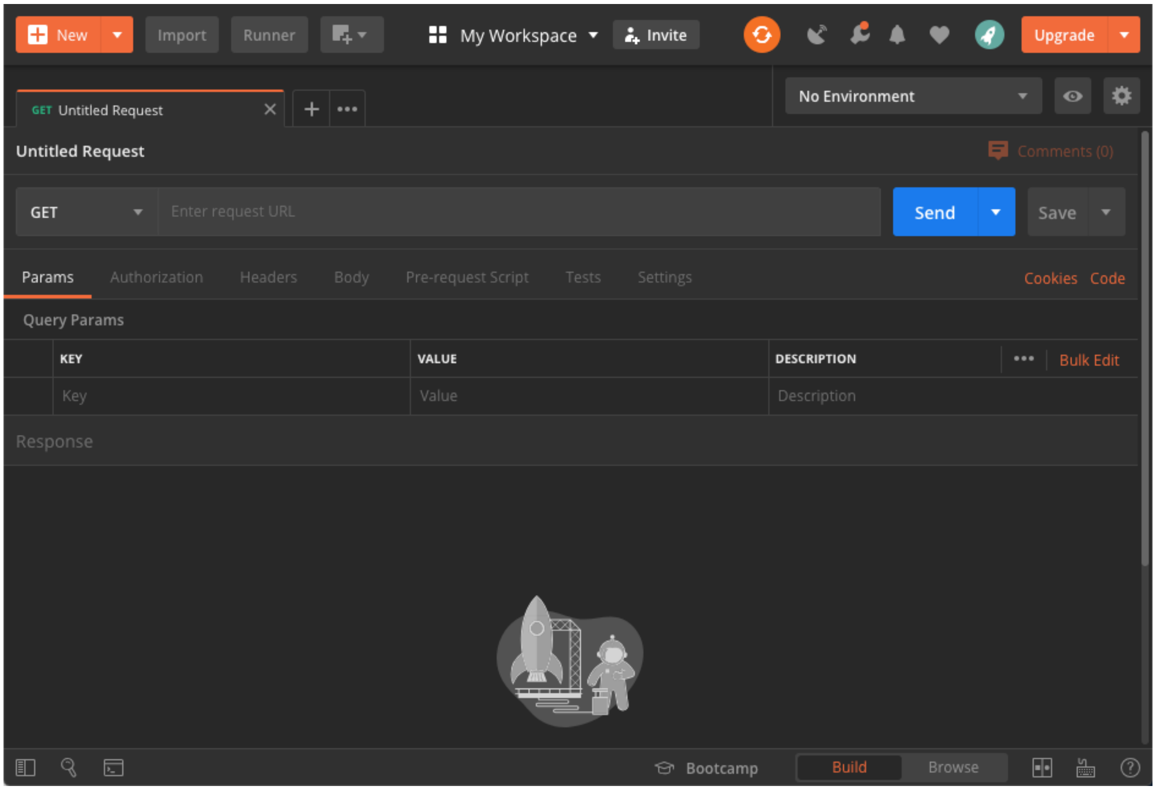
7. You have successfully uploaded a double column CSV.

Multiple Column CSV

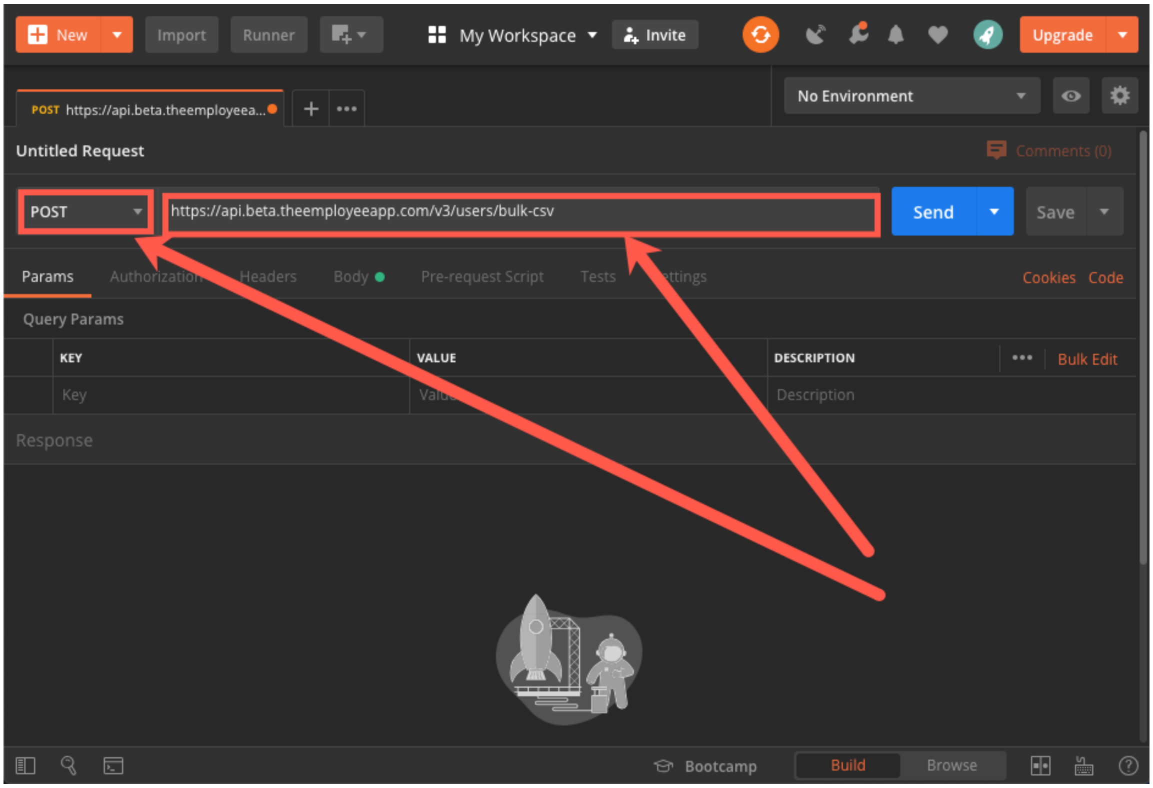
1. Create sample csv file with the following users and headers (regcode, groups):
   Note: Make sure to include Account Admin is in the CSV

```
Name,EmployeeID,Department,Location,Phone
Administrator,Admin_DunderMifflin,Management,Scranton,5555555000
Michael,Michael_Scott,Management,Scranton,5555555001
Dwight,Dwight_Schrute,Administrative,Scranton,5555555002
Pam,Pam_Beesly,Administrative,Scranton,5555555003
Jim,Jim_Halpert,Sales,Scranton,5555555004
Stanley,Stanley_Hudson,Sales,Scranton,5555555005
Phyllis,Phyllis_Lapin,Sales,Scranton,5555555006
Angela,Angela_Martin,Accounting,Scranton,5555555007
Oscar,Oscar_Gutierrez,Accounting,Scranton,5555555008
Toby,Toby_Flenderson,HR,Scranton,5555555009
```

2. Open Postman



3. Set the request type to POST and enter the API endpoint:
   a. Production: https://api.theemployeeapp.com/v3/users/bulk-csv
   b. Beta: https://api.beta.theemployeeapp.com/v3/users/bulk-csv



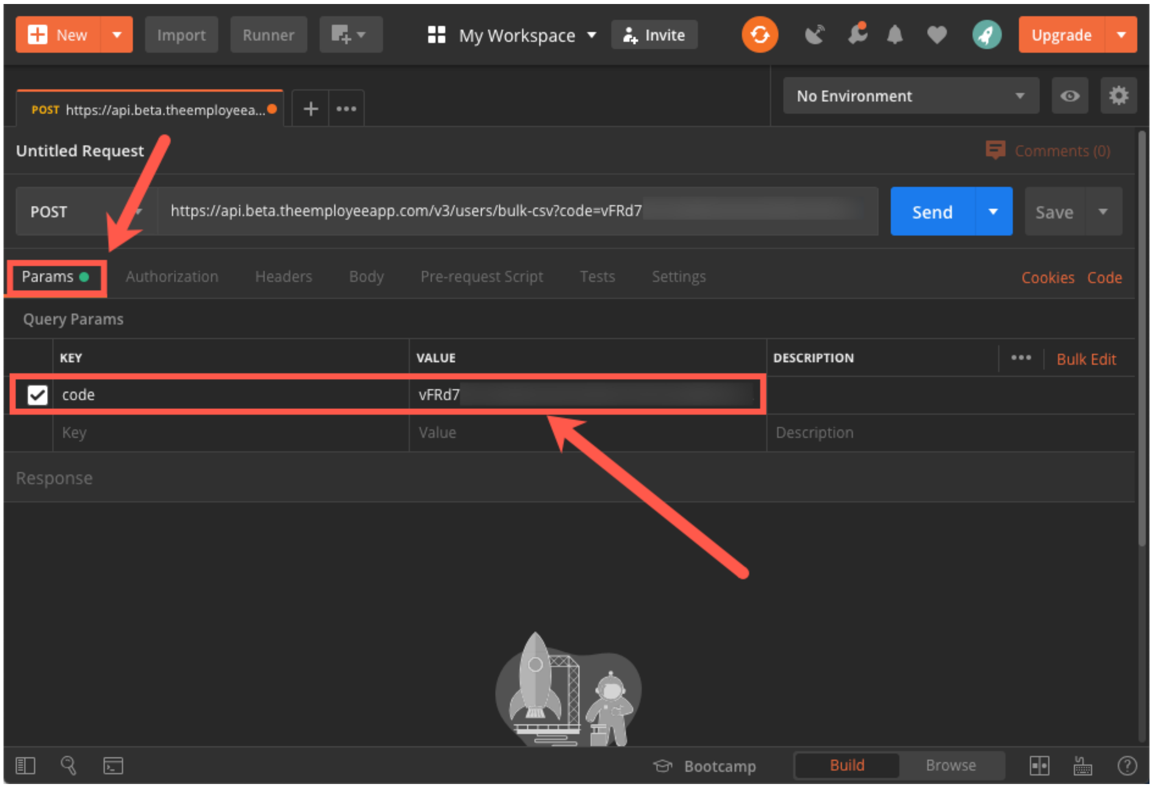
POST

https://api.beta.theemployeeapp.com/v3/users/bulk-csv

KEY	VALUE	DESCRIPTION
Key	Value	Description

4. Ensure the following parameters are set:

   code: [Contact customercare@theemployeeapp.com to obtain a grant code]



POST https://api.beta.theemployeea...

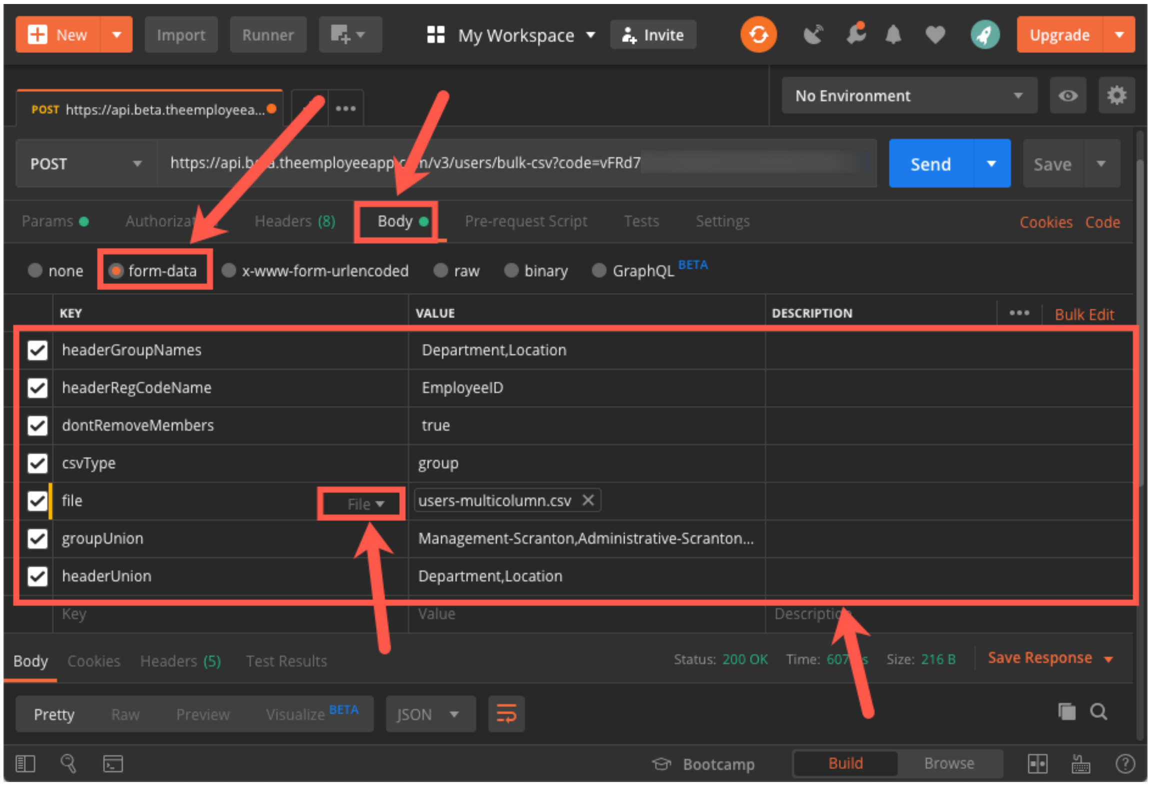
Untitled Request

POST https://api.beta.theemployeeapp.com/v3/users/bulk-csv?code=vFRd7[redacted]

Query Params

	KEY	VALUE	DESCRIPTION
☑	code	vFRd7[redacted]	
	Key	Value	Description

5. Select the file by clicking on **Body**. Then select form-data as the Body type. The following keys and values need to be added:
    a. headerGroupNames - Customize to your CSV
    b. headerRegCodeName - Customize to your CSV
    c. dontRemoveMembers: true
    d. csvType: group
    c. file - Select dropdown for Key type to be File, then **Select File** from local machine
    d. groupUnion:Management-Scranton,Administrative-Scranton,Sales-Scranton,Accounting-Scranton,HR-Scranton
    e. headerUnion: Department,Location



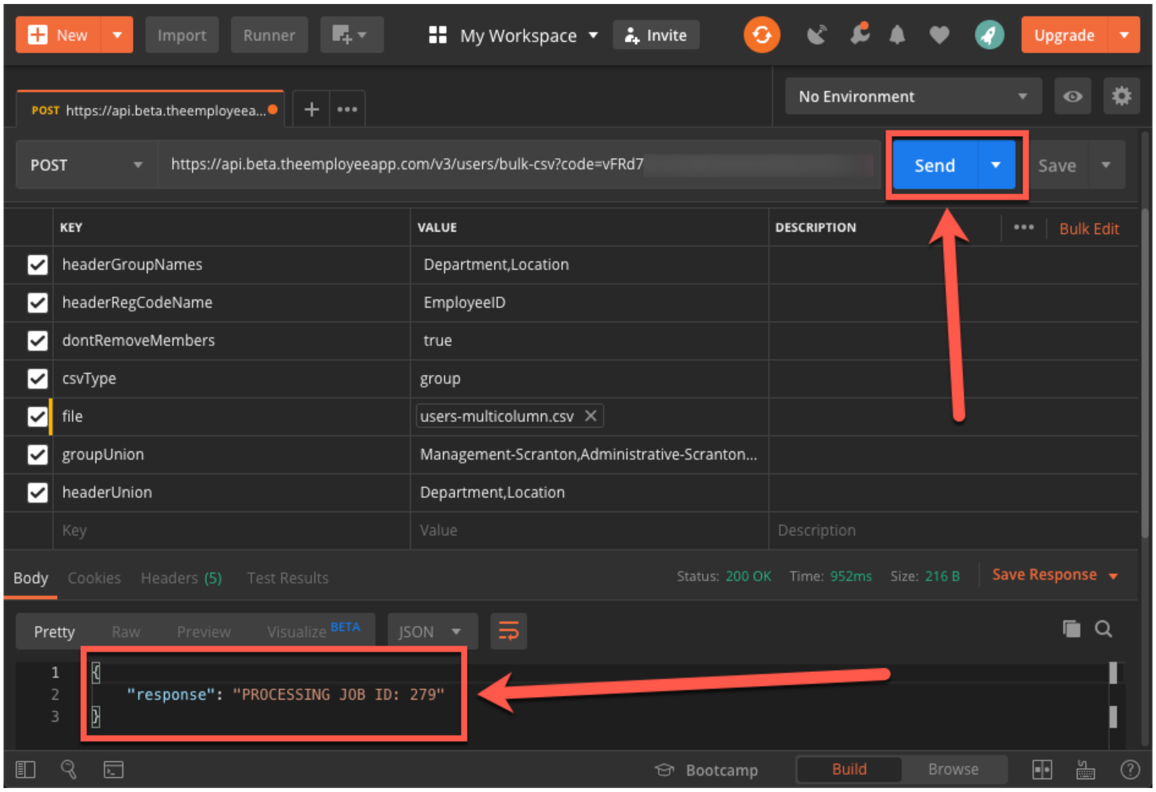
POST https://api.beta.theemployeea...

POST https://api.beta.theemployeeapp.com/v3/users/bulk-csv?code=vFRd7 [redacted]

	KEY	VALUE	DESCRIPTION
☑	headerGroupNames	Department,Location	
☑	headerRegCodeName	EmployeeID	
☑	dontRemoveMembers	true	
☑	csvType	group	
☑	file File ▾	users-multicolumn.csv ✕	
☑	groupUnion	Management-Scranton,Administrative-Scranton...	
☑	headerUnion	Department,Location	
	Key	Value	Description

Status: 200 OK Time: 607 ms Size: 216 B

6. Once all the information is filled out, press Send. A successful response should be returned with the Processing Job ID.



POST https://api.beta.theemployeea...

POST https://api.beta.theemployeeapp.com/v3/users/bulk-csv?code=vFRd7[redacted]

	KEY	VALUE	DESCRIPTION
☑	headerGroupNames	Department,Location	
☑	headerRegCodeName	EmployeeID	
☑	dontRemoveMembers	true	
☑	csvType	group	
☑	file	users-multicolumn.csv ✕	
☑	groupUnion	Management-Scranton,Administrative-Scranton...	
☑	headerUnion	Department,Location	
	Key	Value	Description

Status: 200 OK Time: 952ms Size: 216 B

```
{
    "response": "PROCESSING JOB ID: 279"
}
```

7. You have successfully uploaded a multiple column CSV.

User Management

Users can be managed via REST API requests. This encompasses the ability to create, read, update and delete users through this interface.

Method	Route
PUT	/user/create
PUT	/user/update
DEL	/user/delete
GET	/user/get
PUT	/user/profilephoto
DEL	/user/profilephoto
GET	/user/profilephoto

Example Route: https://api.theemployeeapp.com/v3/user/create

Create User

Method	Route
PUT	/user/create

Example Route: https://api.theemployeeapp.com/v3/user/create

User Request Body

Name	DataType	Description	Example
regCode (required)	string	Unique identifier for each user. Typically email address or EmployeeID.	JSMITH001 john@smith.com
firstName *(optional)*	string	First name of the user	John
lastName *(optional)*	string	Last name of the user	Smith
email *(optional)*	string	User email address	john@smith.com
mobileNumber *(optional)*	string	User's mobile number. Field should be formatted in the E164 format. Number should contain 1 (one) character and 11 (eleven) digits	+144466612345
password *(optional)*	string	User password, can be used to set default password or reset user password. Note: Once a password is set, the user will be automatically registered. The user will be able	P@$sw0rD

		to login and unable to go through the registration process.	
groups *(optional)*	Array string	Group names for the user. This is represented as an array of strings.	[“Group1”] [“Group1”,“Group2”]

Sample Request Body

JSON format

```
{
      "regCode": "JSMITH001",
      "firstName": "John",
      "lastName": "Smith",
      "email": "john@smith.com",
      "mobileNumber": "+144466612345",
      "password": "1234",
      "groups": ["Group 1", "Group 2"]
}
```

Request Parameters

Name	DataType	Description	Example
code (required)	string	API Key (Grant Code)	-

Responses

HTTP Code	Description	Schema
200	Success	-
400	Invalid Input	-
409	Duplicate User/Email	-

Update User

Method	Route
PUT	/user/update

Example Route: https://api.theemployeeapp.com/v3/user/update

User Request Body

Name	DataType	Description	Example
regCode (required)	string	Unique identifier for each user. Typically email address or EmployeeID.	JSMITH001 john@smith.com
updateFirstName *(optional)*	string	First name of the user	John
updateLastName *(optional)*	string	Last name of the user	Smith
updateEmail *(optional)*	string	User email address	john@smith.com
mobileNumber *(optional)*	string	User's mobile number. Field should be formatted in the E164 format. Number should contain 1 (one) character and 11 (eleven) digits	+144466612345
updatePassword *(optional)*	string	User password, can be used to set default password or reset user password. Note: Once a password is set, the user will be automatically registered if they are currently unregistered. The user will be	P@$sw0rD

		able to login and unable to go through the registration process.	
updateGroups *(optional)*	Array string	Group names for the user. This is represented as an array of strings.	[“Group1”] [“Group1”,“Group2”]

Sample Request Body

JSON format

```
{
      "regCode": "JSMITH001",
      "updateFirstName": "John",
      "updateLastName": "Smith",
      "updateEmail": "john@smith.com",
      "mobileNumber": "+14446661234",
      "updatePassword": "1234",
      "updateGroups": ["Group 1", "Group 2", "Group 3"]
}
```

Request Parameters

Name	DataType	Description	Example
code (required)	string	API Key (Grant Code)	-

Responses

HTTP Code	Description	Schema
200	Success	-
400	Invalid Input	-
409	Duplicate User/Email	-

Delete User

Method	Route
DEL	/user/delete

Example Route: https://api.theemployeeapp.com/v3/user/delete

User Request Body

Name	DataType	Description	Example
regCode (required)	string	Unique identifier for each user. Typically email address or EmployeeID.	JSMITH001 john@smith.com

Sample Request Body

JSON format

```
{
        "regCode": "JSMITH001"
}
```

Request Parameters

Name	DataType	Description	Example
code (required)	string	API Key (Grant Code)	-

Responses

HTTP Code	Description	Schema
200	Success	-
400	Invalid Input	-

Read User

Method	Route
GET	/user/get

Example Route: https://api.theemployeeapp.com/v3/user/get

Request Parameters

Name	DataType	Description	Example
code (required)	string	API Key (Grant Code)	-
regCode (required)	string	Unique identifier for each user. Typically email address or EmployeeID.	JSMITH001 john@smith.com

Responses

HTTP Code	Description	Schema
200	Success	-
400	Invalid Input	-

Add Photo

Method	Route
PUT	/user/profilephoto

Example Route: https://api.theemployeeapp.com/v3/user/profilephoto

User Request Body

Name	DataType	Description	Example
file (required)	file	The file to upload	Test.png Test.jpg
regCode (required)	string	Unique identifier for each user. Typically email address or EmployeeID.	JSMITH001 john@smith.com

Request Parameters

Name	DataType	Description	Example
code (required)	string	API Key (Grant Code)	-

Responses

HTTP Code	Description	Schema
200	Success	-
404	Not Found	-
401	Invalid Grant Code	-
500	Error adding the photo	-

Delete Photo

Method	Route
DEL	/user/profilephoto

Example Route: https://api.theemployeeapp.com/v3/user/profilephoto

User Request Body

Name	DataType	Description	Example
regCode (required)	string	Unique identifier for each user. Typically email address or EmployeeID.	JSMITH001 john@smith.com

Sample Request Body

JSON format

```
{
      "regCode": "JSMITH001"
}
```

Request Parameters

Name	DataType	Description	Example
code (required)	string	API Key (Grant Code)	-

Responses

HTTP Code	Description	Schema
200	Success	-
404	Not Found	-
401	Invalid Grant Code	-

500	Error deleting the photo	-

Read Photo

Method	Route
GET	/user/profilephoto

Example Route: https://api.theemployeeapp.com/v3/user/profilephoto

Request Parameters

Name	DataType	Description	Example
code (required)	string	API Key (Grant Code)	-
regCode (required)	string	Unique identifier for each user. Typically email address or EmployeeID.	JSMITH001 john@smith.com

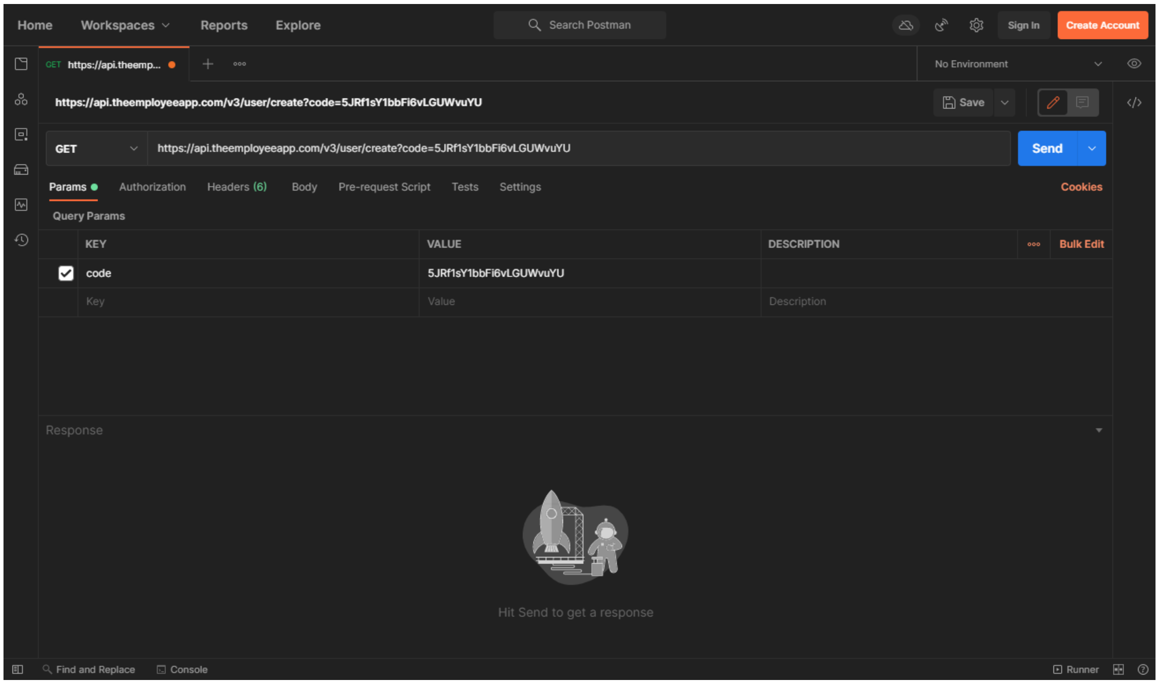
Responses

HTTP Code	Description	Schema
200	Success	-
404	Not Found	-
401	Invalid Grant Code	-
500	Error getting the photo	-

Example Usage

Create User

Request Parameters



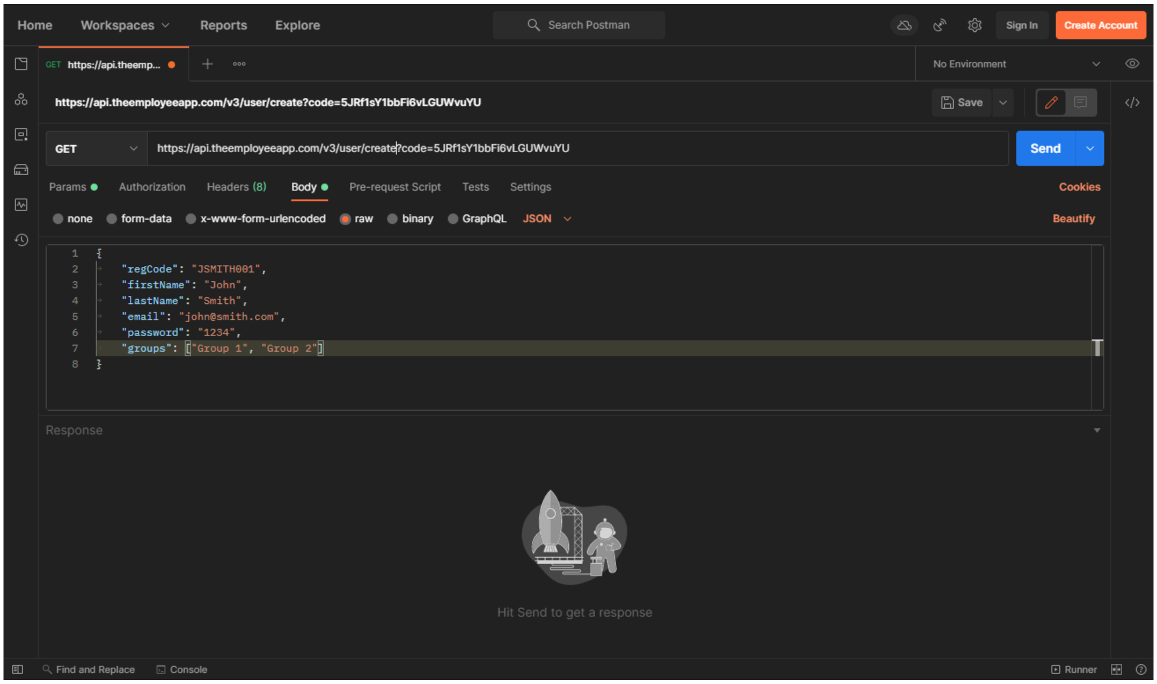
https://api.theemployeeapp.com/v3/user/create?code=5JRf1sY1bbFi6vLGUWvuYU

GET https://api.theemployeeapp.com/v3/user/create?code=5JRf1sY1bbFi6vLGUWvuYU

Query Params

	KEY	VALUE	DESCRIPTION
☑	code	5JRf1sY1bbFi6vLGUWvuYU	
	Key	Value	Description

Request Body (JSON)



https://api.theemployeeapp.com/v3/user/create?code=5JRf1sY1bbFi6vLGUWvuYU

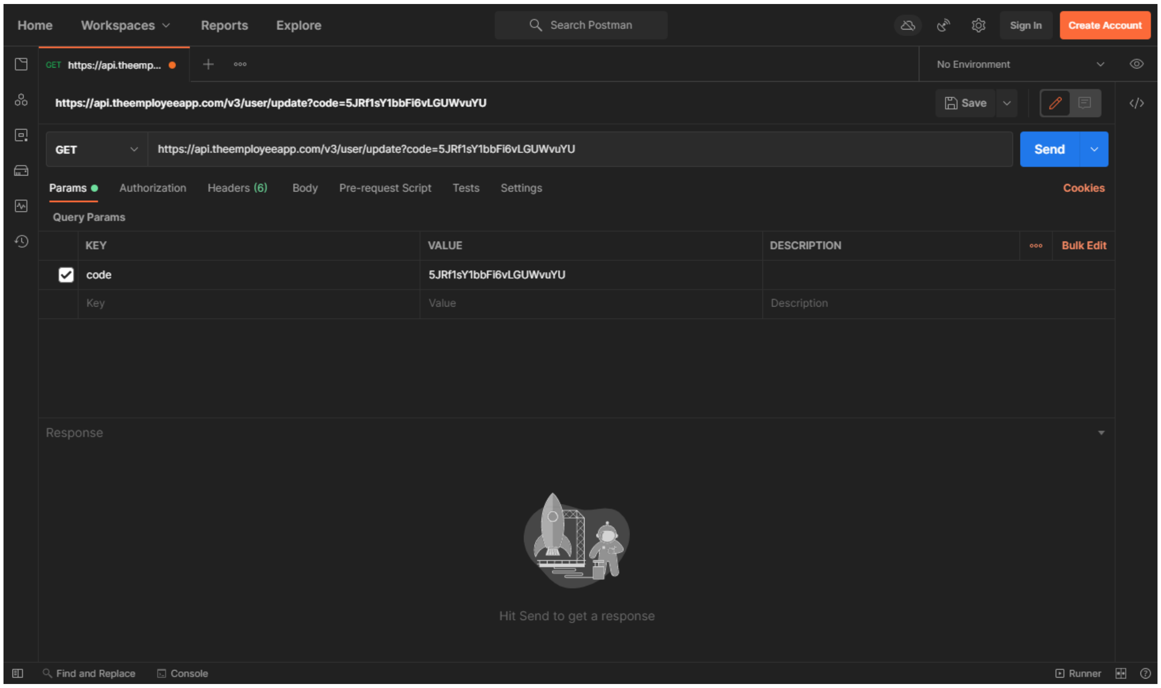
GET https://api.theemployeeapp.com/v3/user/create?code=5JRf1sY1bbFi6vLGUWvuYU

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```json
{
    "regCode": "JSMITH001",
    "firstName": "John",
    "lastName": "Smith",
    "email": "john@smith.com",
    "password": "1234",
    "groups": ["Group 1", "Group 2"]
}
```

Update User

Request Parameters



https://api.theemployeeapp.com/v3/user/update?code=5JRf1sY1bbFi6vLGUWvuYU

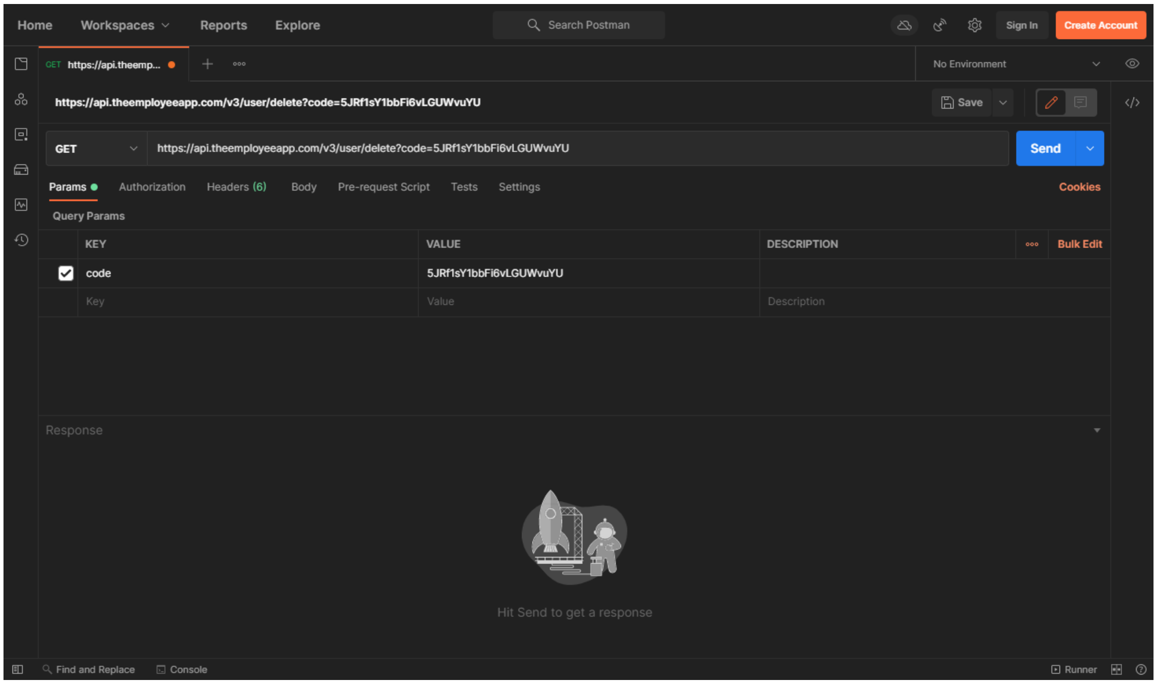
GET https://api.theemployeeapp.com/v3/user/update?code=5JRf1sY1bbFi6vLGUWvuYU

Query Params

	KEY	VALUE	DESCRIPTION
☑	code	5JRf1sY1bbFi6vLGUWvuYU	
	Key	Value	Description

Delete User

Request Parameters



GET https://api.theemp...

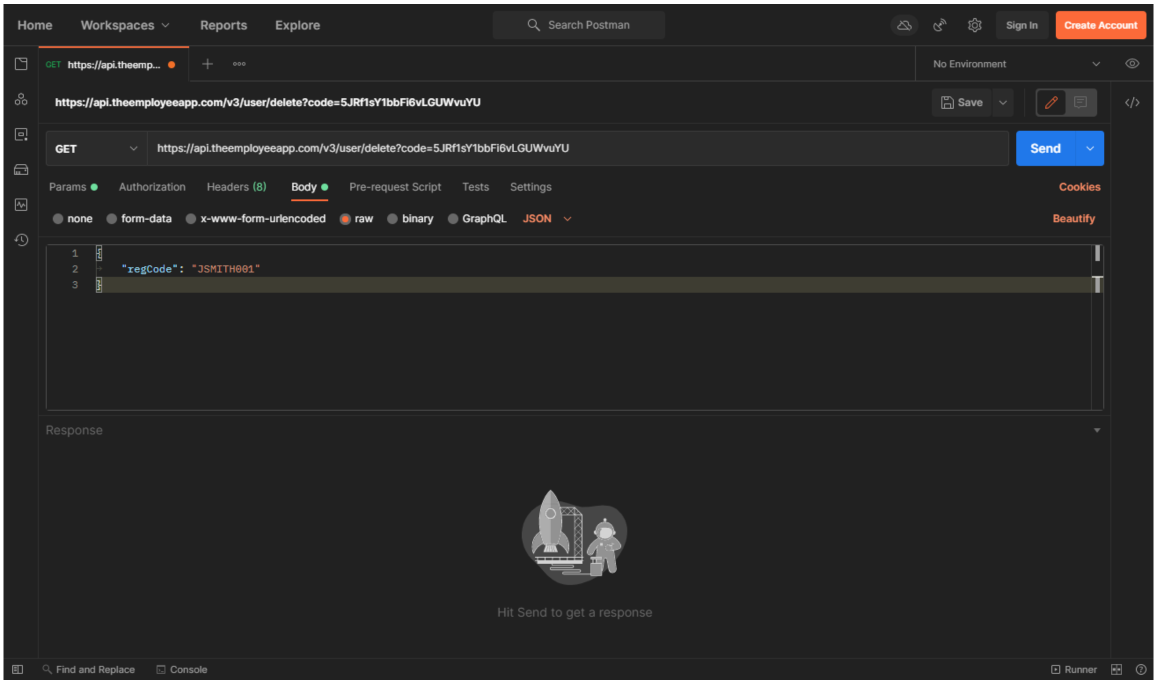
https://api.theemployeeapp.com/v3/user/delete?code=5JRf1sY1bbFi6vLGUWvuYU

GET https://api.theemployeeapp.com/v3/user/delete?code=5JRf1sY1bbFi6vLGUWvuYU

Query Params

	KEY	VALUE	DESCRIPTION
☑	code	5JRf1sY1bbFi6vLGUWvuYU	
	Key	Value	Description

Request Body (JSON)



GET https://api.theemp...

https://api.theemployeeapp.com/v3/user/delete?code=5JRf1sY1bbFi6vLGUWvuYU

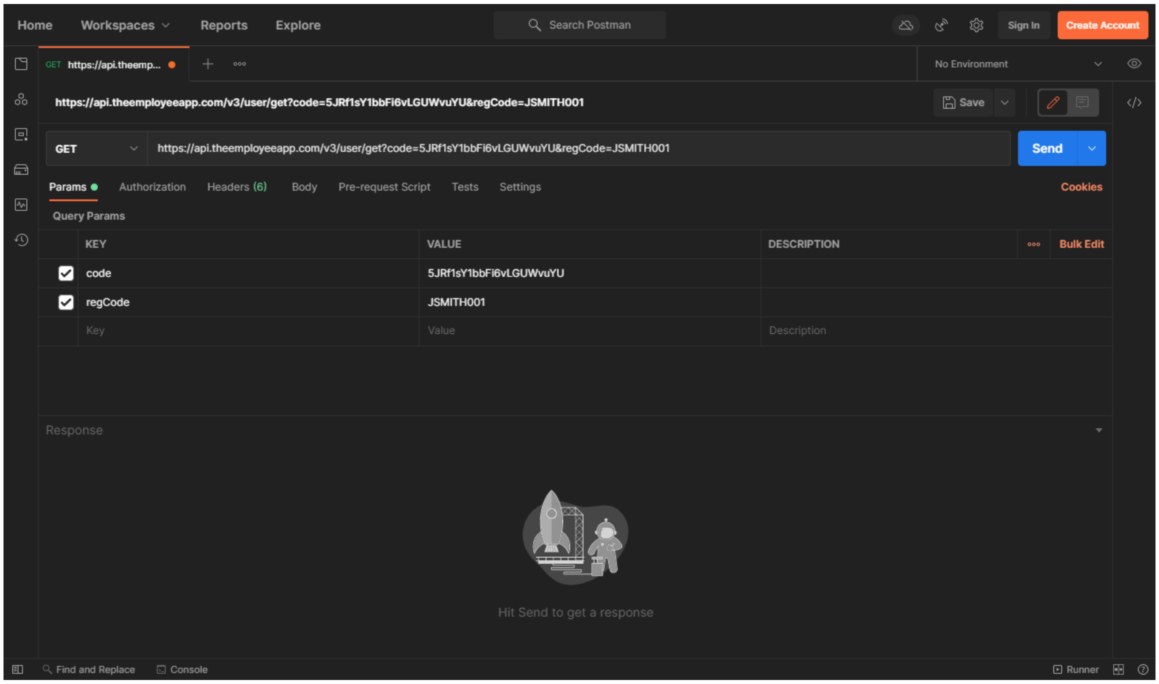
GET https://api.theemployeeapp.com/v3/user/delete?code=5JRf1sY1bbFi6vLGUWvuYU

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "regCode": "JSMITH001"
}
```

Read User

Request Parameters



https://api.theemployeeapp.com/v3/user/get?code=5JRf1sY1bbFi6vLGUWvuYU&regCode=JSMITH001

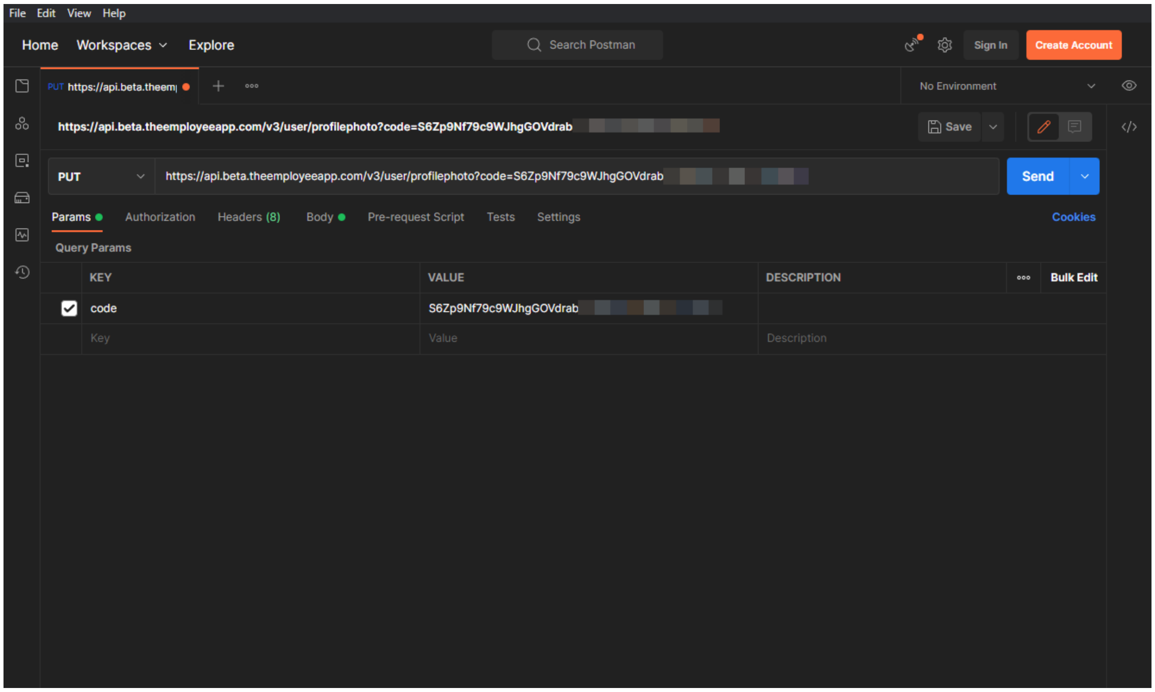
GET https://api.theemployeeapp.com/v3/user/get?code=5JRf1sY1bbFi6vLGUWvuYU&regCode=JSMITH001

Query Params

	KEY	VALUE	DESCRIPTION
☑	code	5JRf1sY1bbFi6vLGUWvuYU	
☑	regCode	JSMITH001	
	Key	Value	Description

Add Photo

Request Parameters



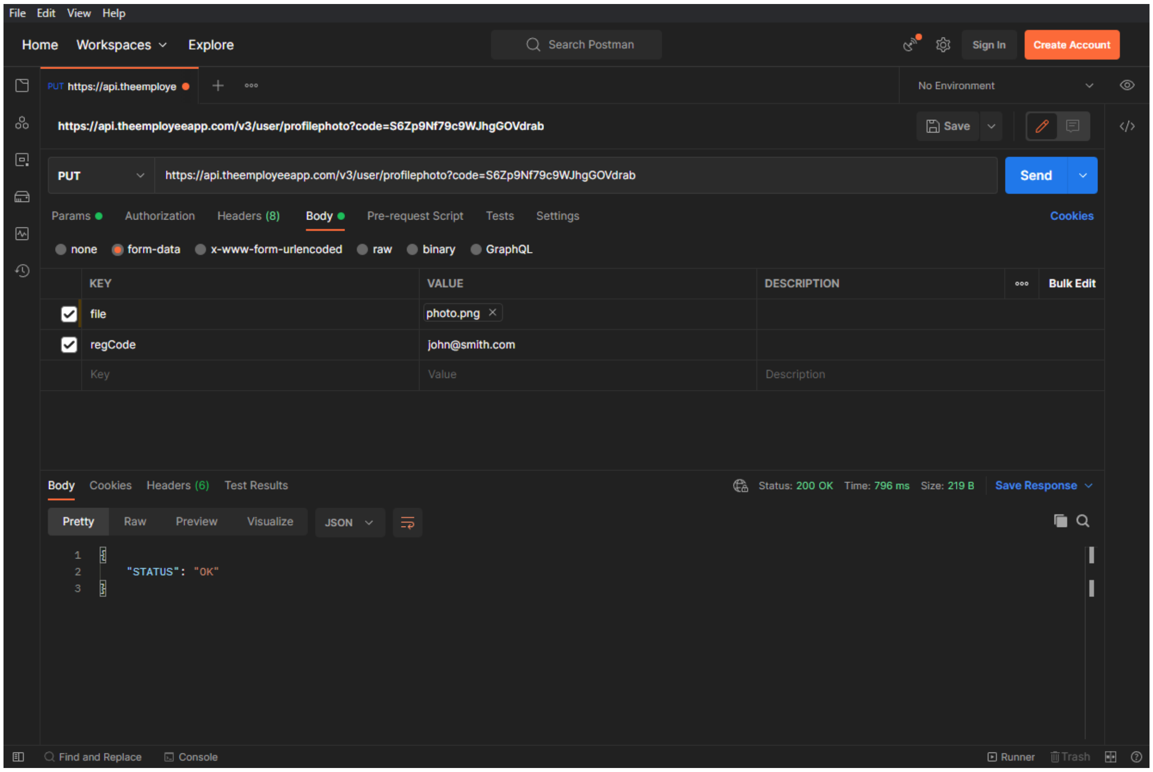
https://api.beta.theemployeeapp.com/v3/user/profilephoto?code=S6Zp9Nf79c9WJhgGOVdrab[redacted]

PUT https://api.beta.theemployeeapp.com/v3/user/profilephoto?code=S6Zp9Nf79c9WJhgGOVdrab[redacted]

Query Params

	KEY	VALUE	DESCRIPTION
☑	code	S6Zp9Nf79c9WJhgGOVdrab[redacted]	
	Key	Value	Description

Request Body (JSON)



https://api.theemployeeapp.com/v3/user/profilephoto?code=S6Zp9Nf79c9WJhgGOVdrab

PUT https://api.theemployeeapp.com/v3/user/profilephoto?code=S6Zp9Nf79c9WJhgGOVdrab

Params ● Authorization Headers (8) Body ● Pre-request Script Tests Settings

○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	KEY	VALUE	DESCRIPTION
☑	file	photo.png	
☑	regCode	john@smith.com	
	Key	Value	Description

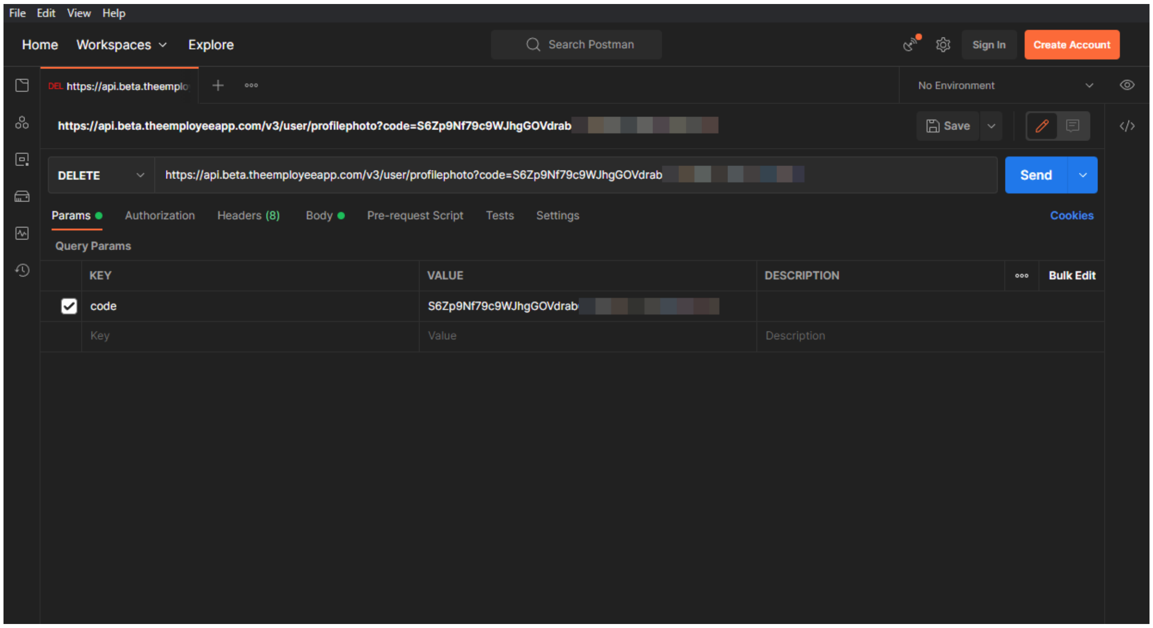
Body Cookies Headers (6) Test Results — Status: 200 OK Time: 796 ms Size: 219 B Save Response

Pretty Raw Preview Visualize JSON

```
{
    "STATUS": "OK"
}
```

Delete Photo

Request Parameters



https://api.beta.theemployeeapp.com/v3/user/profilephoto?code=S6Zp9Nf79c9WJhgGOVdrab[redacted]

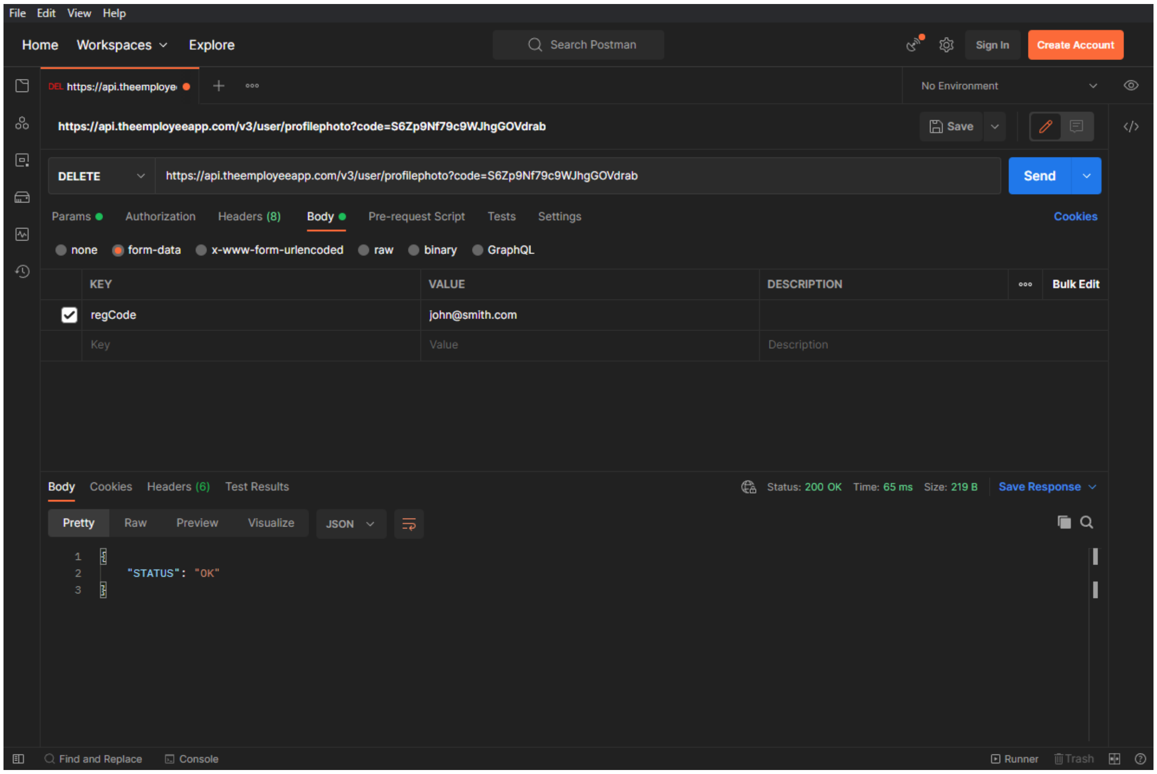
DELETE https://api.beta.theemployeeapp.com/v3/user/profilephoto?code=S6Zp9Nf79c9WJhgGOVdrab[redacted]

Params ● Authorization Headers (8) Body ● Pre-request Script Tests Settings

Query Params

	KEY	VALUE	DESCRIPTION
☑	code	S6Zp9Nf79c9WJhgGOVdrab[redacted]	
	Key	Value	Description

Request Body (JSON)



File Edit View Help
Home Workspaces Explore
Search Postman
Sign In
Create Account
DEL https://api.theemploye
No Environment
Save
Params ● Authorization Headers (8) Body ● Pre-request Script Tests Settings
Cookies
Bulk Edit
Body Cookies Headers (6) Test Results
Save Response
Pretty Raw Preview Visualize JSON
Find and Replace
Console
Runner
Trash

https://api.theemployeeapp.com/v3/user/profilephoto?code=S6Zp9Nf79c9WJhgGOVdrab

DELETE https://api.theemployeeapp.com/v3/user/profilephoto?code=S6Zp9Nf79c9WJhgGOVdrab

○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

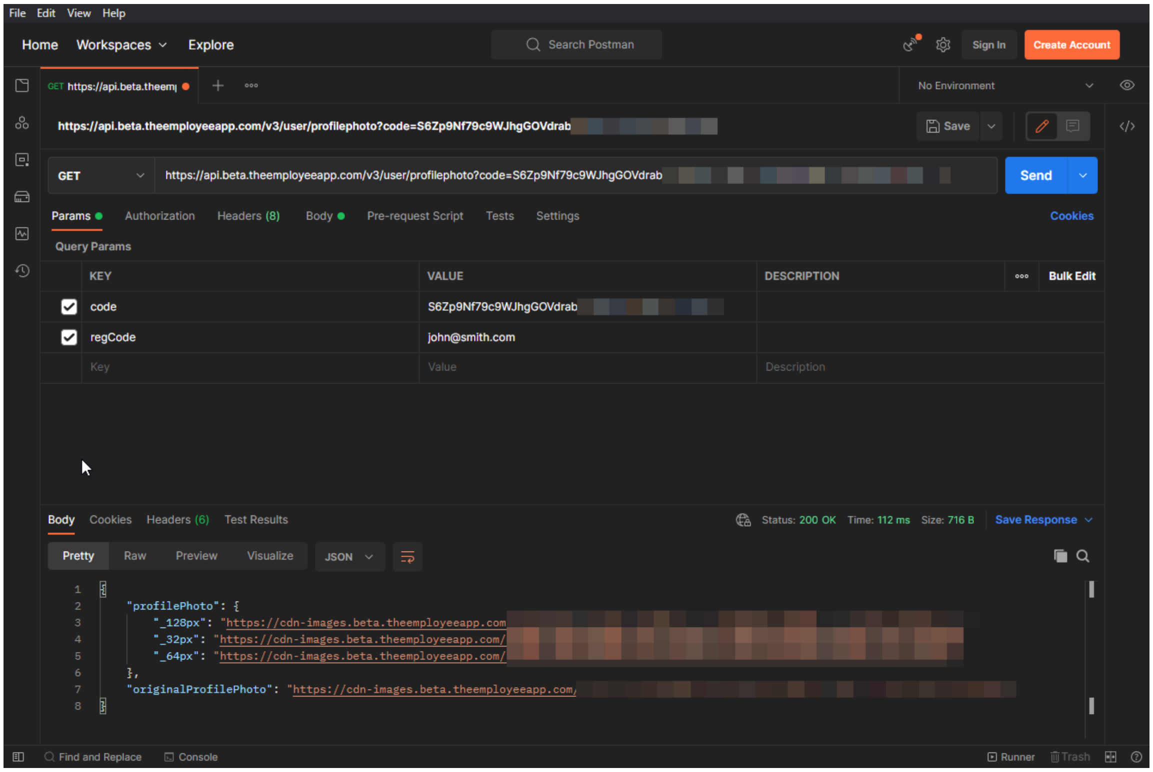
	KEY	VALUE	DESCRIPTION
☑	regCode	john@smith.com	
	Key	Value	Description

Status: 200 OK Time: 65 ms Size: 219 B

```
{
    "STATUS": "OK"
}
```

Read Photo

Request Parameters



https://api.beta.theemployeeapp.com/v3/user/profilephoto?code=S6Zp9Nf79c9WJhgGOVdrab[redacted]

GET https://api.beta.theemployeeapp.com/v3/user/profilephoto?code=S6Zp9Nf79c9WJhgGOVdrab[redacted]

Send

Query Params

	KEY	VALUE	DESCRIPTION
☑	code	S6Zp9Nf79c9WJhgGOVdrab[redacted]	
☑	regCode	john@smith.com	
	Key	Value	Description

Body Cookies Headers (6) Test Results

Status: 200 OK Time: 112 ms Size: 716 B Save Response

```
{
    "profilePhoto": {
        "_128px": "https://cdn-images.beta.theemployeeapp.com[redacted]
        "_32px": "https://cdn-images.beta.theemployeeapp.com/[redacted]
        "_64px": "https://cdn-images.beta.theemployeeapp.com/[redacted]
    },
    "originalProfilePhoto": "https://cdn-images.beta.theemployeeapp.com/[redacted]
}
```

Directory Contact Management

Directory Contacts can be managed via REST API requests. This encompasses the ability to create, read, update and delete directory contacts through this interface.

Method	Route
POST	/directory/directory-contact/create
PUT	/directory/directory-contact/:uid
DEL	/directory/directory-contact/:uid
DEL	/directory/directory-contact/photo/:uid
GET	/directory/directory-contact/photo/:uid
PUT	/directory/directory-contact/photo/:uid
POST	/directory/directory-contact/send-report

Example Route: https://api.theemployeeapp.com/v3/directory/directory-contact/create

Create Directory Contact

Method	Route
POST	/directory/directory-contact/create

Example Route: https://api.theemployeeapp.com/v3/directory/directory-contact/create

User Request Body

Name	DataType	Description	Example
firstName (required)	string	First name of the contact	John
lastName (required)	string	Last name of the contact	Smith
middleName *(optional)*	string	Middle name of the contact	James
name *(optional)*	string	Full name of the contact	John James Smith
directoryNames *(optional)*	Array string	Directory Names. This is represented as an array of strings.	[“Test1”] [“Test1”, “Test2”]
directories *(optional)*	Array string	Directory IDs. This is represented as an array of strings.	[“62g0e88799a88528209e0a5e”] [“62f0e83799a88528209e0a5e”,“62f0e85799a88528209e0e5e”]
thumbnailSource *(optional)*	string	The source of the thumbnail	“cms” or “api”

contactInfo *(optional)*	object	Contact information for the contact	Details in rows below
contactInfo - title *(optional)*	string	Title of the contact	“Engineer”
contactInfo - company *(optional)*	string	Company of the contact	“TEA”
contactInfo - address1 *(optional)*	string	Main address of the contact	“1 Test Lane”
contactInfo - address2 *(optional)*	string	Additional address info for the contact	“Building 1”
contactInfo - address3 *(optional)*	string	Additional address info for the contact	“Room 123”
contactInfo - city *(optional)*	string	City of the contact	“New Brunswick”
contactInfo - state *(optional)*	string	State of the contact	“NJ”
contactInfo - zip *(optional)*	string	Zip of the contact	“09876”
contactInfo - country *(optional)*	string	Country of the contact	“USA”
contactInfo - email *(optional)*	object	Email of the contact	“Work”: “test@test.com”
contactInfo - phone *(optional)*	object	Phone of the contact	“Mobile”: “9087089876”

contactInfo - additionalData *(optional)*	array	Any additional data for the contact	“Birthday”: “09/08/95”

Sample Request Body

JSON format

```
{
       "firstName": "John",
       "middleName": "James",
       "lastName": "Smith",
       "directoryNames": ["Test1"],
       "contactInfo": {
         "title": "Software Engineer",
         "company": "Test",
         "address1": "1 Test Lane",
         "city": "Test City",
         "state": "NJ",
         "zip": "08867",
         "country": "USA",
         "email": {
           "work": "test@test.com"
         },
         "phone": {
           "mobile": "9087089876"
         },
         "additionalData": [
           {
             "key": "birthday",
             "value": "07/24/96"
           }
         ]
       }
}
```

Request Parameters

Name	DataType	Description	Example
code (required)	string	API Key (Grant Code)	-

Responses

HTTP Code	Description	Schema
200	Success	-
401	Invalid grant code/access token	-
401	Duplicate Contact	-
500	Error creating the contact	

Update Directory Contact

Method	Route
PUT	/directory/directory-contact/:uid

Example Route:
https://api.theemployeeapp.com/v3/directory/directory-contact/62f53a3147b3ce3bd7bb12e5

User Request Body

Name	DataType	Description	Example
firstName (required)	string	First name of the contact	John
lastName (required)	string	Last name of the contact	Smith
middleName *(optional)*	string	Middle name of the contact	James
name *(optional)*	string	Full name of the contact	John James Smith
directoryNames *(optional)*	Array string	Directory Names. This is represented as an array of strings.	[“Test1”] [“Test1”, “Test2”]
directories *(optional)*	Array string	Directory IDs. This is represented as an array of strings.	[“62g0e88799a88528209e0a5e”] [“62f0e83799a88528209e0a5e”,“62f0e85799a88528209e0e5e”]

thumbnailSource *(optional)*	string	The source of the thumbnail	“cms” or “api”
contactInfo *(optional)*	object	Contact information for the contact	Details in rows below
contactInfo - title *(optional)*	string	Title of the contact	“Engineer”
contactInfo - company *(optional)*	string	Company of the contact	“TEA”
contactInfo - address1 *(optional)*	string	Main address of the contact	“1 Test Lane”
contactInfo - address2 *(optional)*	string	Additional address info for the contact	“Building 1”
contactInfo - address3 *(optional)*	string	Additional address info for the contact	“Room 123”
contactInfo - city *(optional)*	string	City of the contact	“New Brunswick”
contactInfo - state *(optional)*	string	State of the contact	“NJ”
contactInfo - zip *(optional)*	string	Zip of the contact	“09876”
contactInfo - country *(optional)*	string	Country of the contact	“USA”
contactInfo - email *(optional)*	object	Email of the contact	“Work”: “test@test.com”

contactInfo - phone *(optional)*	object	Phone of the contact	“Mobile”: “9087089876”
contactInfo - additionalData *(optional)*	array	Any additional data for the contact	“Birthday”: “09/08/95”

Sample Request Body

JSON format

```
{
      "firstName": "John",
      "middleName": "James",
      "lastName": "Smith",
      "directoryNames": ["Test1"],
      "contactInfo": {
        "title": "Software Engineer",
        "company": "Test",
        "address1": "1 Test Lane",
        "city": "Test City",
        "state": "NJ",
        "zip": "08867",
        "country": "USA",
        "email": {
          "work": "test@test.com"
        },
        "phone": {
          "mobile": "9087089876"
        },
        "additionalData": [
          {
            "key": "birthday",
            "value": "07/24/96"
          }
        ]
      }
}
```

Request Parameters

Name	DataType	Description	Example
code (required)	string	API Key (Grant Code)	-

Responses

HTTP Code	Description	Schema
200	Success	-
401	Invalid grant code/access token	-
401	Duplicate Contact	-
500	Error creating the contact	

Delete Directory Contact

Method	Route
DEL	/directory/directory-contact/:uid

Example Route:
https://api.theemployeeapp.com/v3/directory/directory-contact/62f53a3147b3ce3bd7bb12e5

Request Parameters

Name	DataType	Description	Example
code (required)	string	API Key (Grant Code)	-

Responses

HTTP Code	Description	Schema
200	Success	-
401	Invalid grant code/access token	-
500	Error deleting the contact	

Delete Directory Contact Photo

Method	Route
DEL	/directory/directory-contact/photo/:uid

Example Route:
https://api.theemployeeapp.com/v3/directory/directory-contact/photo/62f53a3147b3ce3bd7bb12e5

Request Parameters

Name	DataType	Description	Example
code (required)	string	API Key (Grant Code)	-

Responses

HTTP Code	Description	Schema
200	Success	-
401	Invalid grant code/access token	-
500	Error deleting the contact photo	

Read Directory Contact Photo

Method	Route
GET	/directory/directory-contact/photo/:uid

Example Route:
https://api.theemployeeapp.com/v3/directory/directory-contact/photo/62f53a3147b3ce3bd7bb12e5

Request Parameters

Name	DataType	Description	Example
code (required)	string	API Key (Grant Code)	-

Responses

HTTP Code	Description	Schema
200	Success	-

Add Directory Contact Photo

Method	Route
PUT	/directory/directory-contact/photo/:uid

Example Route:
https://api.theemployeeapp.com/v3/directory/directory-contact/photo/62f53a3147b3ce3bd7bb12e5

User Request Body

Name	DataType	Description	Example
file (optional)	file	The file to upload	Test.png Test.jpg
imageUrl (optional)	string	Optional imageUrl link to upload	https://images.com/Test.png https://images.com/Test.jpg

Request Parameters

Name	DataType	Description	Example
code (required)	string	API Key (Grant Code)	-

Responses

HTTP Code	Description	Schema
200	Success	-

404	Not Found	-
401	Invalid Grant Code	-
500	Error adding the photo	-

Send Directory Contact Report

Method	Route
POST	/directory/directory-contact/send-report

Example Route: https://api.theemployeeapp.com/v3/directory/directory-contact/send-report

User Request Body

Name	DataType	Description	Example
email (required)	file	The email to send the report to	test@test.com
directoryId (optional)	string	The directory ID to get the contacts from	62f53a3147b3ce3bd7bb12e5
directoryName (optional)	string	The directory name to get the contacts from	Test1

Request Parameters

Name	DataType	Description	Example
code (required)	string	API Key (Grant Code)	-

Responses

HTTP Code	Description	Schema
200	Success	-
404	Not Found	-
401	Invalid Grant Code/access token	-
401	No directory provided	
401	No email provided	
401	Invalid directory provided	
500	Error adding the photo	-

Group Management

Directory Contacts can be managed via REST API requests. This encompasses the ability to create, read, update and delete directory contacts through this interface.

Method	Route
POST	/group
PUT	/group/:uid
GET	/group/:uid
DEL	/group/:uid
GET	/group/account/:uid
GET	/group/cfp/:uid

Example Route: https://api.theemployeeapp.com/v3/group

Create Group

Method	Route
POST	/group

Example Route: https://api.theemployeeapp.com/v3/group

User Request Body

Name	DataType	Description	Example
name (required)	string	The name of group	TestGroup
contentFeedProfiles (required)	Array string	The content feed profiles the group is tied to. This is represented as an array of strings.	[“6493d6b74f672b41e701ed60”] [“6493d6b74f672b41e701ed60”,“6493d6b74f672b41e701ed60”]
permissions *(optional)*	Array object	The permissions that tie a user/group to the group. This is represented as an array of objects.	[{“type”: “EDIT”, “user”: “6493d6b74f672b41e701ed60”}]

Sample Request Body

JSON format

```
{
	"name": "TestGroup",
	"contentFeedProfiles": ["6493d6b74f672b41e701ed60"],
	"permissions": [{"type": "EDIT", "user": "6493d6b74f672b41e701ed60"}]
}
```

Request Parameters

Name	DataType	Description	Example
code (required)	string	API Key (Grant Code)	-

Responses

HTTP Code	Description	Schema
200	Success	-
401	Invalid access token/grant code	-
400	Permission type invalid	-
400	Permission missing data	-
400	Group already exists	-
500	Error creating the group	-

Update Group

Method	Route
PUT	/group/:uid

Example Route: https://api.theemployeeapp.com/v3/group/:uid

User Request Body

Name	DataType	Description	Example
name *(optional)*	string	The name of group	TestGroup
contentFeedProfiles *(optional)*	Array string	The content feed profiles the group is tied to. This is represented as an array of strings.	[“6493d6b74f672b41e701ed60”] [“6493d6b74f672b41e701ed60”,“6493d6b74f672b41e701ed60”]
permissions *(optional)*	Array object	The permissions that tie a user/group to the group. This is represented as an array of objects.	[{“type”: “EDIT”, “user”: “6493d6b74f672b41e701ed60”}]

Sample Request Body

JSON format

```
{
      "name": "TestGroup",
      "contentFeedProfiles": ["6493d6b74f672b41e701ed60"],
      "permissions": [{"type": "EDIT", "user": "6493d6b74f672b41e701ed60"}]
}
```

Request Parameters

Name	DataType	Description	Example
code (required)	string	API Key (Grant Code)	-

Responses

HTTP Code	Description	Schema
200	Success	-
401	Invalid access token/grant code	-
400	Group with empty name	-
404	Group not found	-
400	Group already exists	-
500	Error updating the group	-

Read Group

Method	Route
GET	/group/:uid

Example Route: https://api.theemployeeapp.com/v3/group/62f53a3147b3ce3bd7bb12e5

Request Parameters

Name	DataType	Description	Example
code (required)	string	API Key (Grant Code)	-

Responses

HTTP Code	Description	Schema
200	Success	-
401	Invalid access token/grant code	-
500	Error getting the group	-

Delete Group

Method	Route
DEL	/group/:uid

Example Route: https://api.theemployeeapp.com/v3/group/62f53a3147b3ce3bd7bb12e5

Request Parameters

Name	DataType	Description	Example
code (required)	string	API Key (Grant Code)	-

Responses

HTTP Code	Description	Schema
200	Success	-
401	Invalid access token/grant code	-
500	Error deleting the group	-

Read Group by Account

Method	Route
GET	/group/account/:uid

Example Route: https://api.theemployeeapp.com/v3/group/account/62f53a3147b3ce3bd7bb12e5

Request Parameters

Name	DataType	Description	Example
code (required)	string	API Key (Grant Code)	-
limit *(optional)*	int	How many results to return (default 10)	10
offset *(optional)*	int	The page number for results (default 0)	0

Responses

HTTP Code	Description	Schema
200	Success	-
401	Invalid access token/grant code	-
500	Error getting the group by account	-

Read Group by CFP

Method	Route
GET	/group/cfp/:uid

Example Route: https://api.theemployeeapp.com/v3/group/cfp/62f53a3147b3ce3bd7bb12e5

Request Parameters

Name	DataType	Description	Example
code (required)	string	API Key (Grant Code)	-
limit *(optional)*	int	How many results to return (default 10)	10
offset *(optional)*	int	The page number for results (default 0)	0

Responses

HTTP Code	Description	Schema
200	Success	-
401	Invalid access token/grant code	-
500	Error getting the group by cfp	-