Table of Contents

Ulafix REST API Documentation

Introduction

The Ulafix API allows you to perform all the operations that you do with our web client.

Ulafix API is built using REST principles which ensures predictable URLs that make writing applications easy. This API follows HTTP rules, so a wide range of HTTP clients can be used to interact with the API.

Every resource is exposed as a URL. The URL of each resource can be obtained by accessing the API Root Endpoint.

API Root Endpoint: **http://***<domain >***/api/web/v1** *(for production server, replace <domain> with* ***ulafix.com****)*

Overview

HTTP Methods

Method	Description
GET	Used for retrieving resources.
POST	Used for creating resources and performing resource actions.
PUT or PATCH	Used for updating resources.
DELETE	Used for deleting resources.

Response

Responses will be in the JSON format by default. XML format is supported as well for which `application/xml` need to be specified in the respective request's `Accept` header.

Errors

Ulafix API uses HTTP status codes to indicate success or failure of an API call. In general, status codes in the `2xx` range mean success, `4xx` range mean there was an error in the provided information, and those in the `5xx` range indicate server side errors. Commonly used HTTP status codes are listed below:

HTTP Status Codes

- `200`: OK. Everything worked as expected.
- `201`: A resource was successfully created in response to a POST request. The Location header contains the URL pointing to the newly created resource.
- `204`: The request was handled successfully and the response contains no body content (like a DELETE request).
- `400`: Bad request. This could be caused by various actions by the user, such as providing invalid JSON data in the request body, providing invalid action parameters, etc.
- `401`: Authentication failed.
- `403`: The authenticated user is not allowed to access the specified API endpoint.

- `404`: The requested resource does not exist.
- `405`: Method not allowed. Please check the Allow header for the allowed HTTP methods.
- `415`: Unsupported media type. The requested content type or version number is invalid.
- `422`: Data validation failed (in response to a POST request, for example). Please check the response body for detailed error messages.
- `429`: Too many requests. The request was rejected due to rate limiting.
- `500`: Internal server error. This could be caused by internal program errors.

Providers

A provider object keeps track of all provider related information.

CREATE A PROVIDER : A new Provider will be created.	**POST** /users

PARAMETERS

`username`	**required**	Account login username of Provider
`email`	**required**	Email address of Provider
`password`	**required**	Account password
`pr_first_name`	**required**	First name of the provider
`pr_last_name`	**required**	Last name of the provider
`pr_company_name`	**required**	Name of the company the provider represents
`address_line1`	**required**	Provider street address
`address_line2`	optional	Provider street address additional
`city`	**required**	Provider city
`state`	**required**	Provider state
`zipcode`	**required**	Provider zipcode
`zipcode_id`	**required**	DB id of zipcode table corresponding to the `zipcode` above *(available in LIST response structure of **Zipcodes** API)*
`country_id`	**required**	DB id of country table *(available in LIST response structure of **Countries** API)*
`contact_number`	**required**	Provider mobile number
`role`	**required**	For Provider type user, role = 2 OR role = 'Provider'
`pr_company_description`	optional	Provider Company description
`pr_week_open_from`	optional	Provider working hour start time on weekdays
`pr_week_open_till`	optional	Provider working hour end time on weekdays
`pr_sat_open_from`	optional	Provider working hour start time on Saturdays
`pr_sat_open_till`	optional	Provider working hour end time on Saturdays
`pr_sun_open_from`	optional	Provider working hour start time on Sundays
`pr_sun_open_till`	optional	Provider working hour end time on Sundays
`pr_max_jobs`	optional	Max. no. of jobs provider can accept at a time *(defaults to 1)*
`pr_company_logo`	optional	Provider logo file
`pr_services[]`	**required**	DB id of service_type table for Providers' services; multiple values allowed *(available in LIST response structure of **Services** API)*
`latitude`	**required**	Latitude (in decimal) of Provider office location
`longitude`	**required**	Longitude (in decimal) of Provider office location
`web_url`	optional	

Request Example:	**Response Example:**

```
curl -X POST \
  --url http://<domain>/api/web/v1/users \
  -H 'cache-control: no-cache' \
  -H 'content-type: multipart/form-data;
boundary=---011000010111000001101001' \
  -F username=api.provider \
  -F email=test@account.net \
  -F password=xpass123456 \
  -F pr_first_name=API \
  -F pr_last_name=Provider \
  -F 'pr_company_name=API Inc' \
  -F 'address_line1=12409 Lagoon Dr' \
  -F address_line2=null \
  -F city=Curtice \
  -F state=OH \
  -F zipcode_id=18749 \
  -F country_id=233 \
  -F contact_number=1123456780 \
  -F role=Provider \
  -F zipcode=43412 \
  -F 'pr_company_description=This is API
Provider' \
  -F pr_week_open_from=10:00 \
  -F pr_week_open_till=18:00 \
  -F pr_sat_open_from=null \
  -F pr_sat_open_till=null \
  -F pr_sun_open_from=null \
  -F pr_sun_open_till=null \
  -F pr_max_jobs=5 \
  -F pr_company_logo=@abcd.png \
  -F 'pr_services[]=7' \
  -F 'pr_services[]=12' \
  -F latitude=41.655400 \
  -F longitude=-83.246854 \
  -F web_url=http://google.com
```

```
HTTP/1.1 201 Created
Content-Type:application/json;charset=UTF-8
Location:http://<domain>/api/web/v1/users/73
{
    "username": "api.provider",
    "email": "test@account.net",
    "status": "Inactive",
    "role": "Provider",
    "account_activation_token":
"cNIO4lHbjd0oaJ8eAYJVN_BFJEQUfI_7_1450951807",
    "created_at": 1450951807,
    "updated_at": 1450951807,
    "id": 73,
    "role_numeric": 2,
    "status_numeric": 0,
    "profile": {
        "id": 29,
        "user_id": 73,
        "company_name": "API Inc",
        "contact_first_name": "API",
        "contact_last_name": "Provider",
        "company_logo_path":
"TfS0NGgb0f99a2GcypOSFhUaLFsk-h6W.png",
        "company_description": "This is API
Provider",
        "week_open_from": "10:00:00",
        "week_open_till": "18:00:00",
        "sat_open_from": null,
        "sat_open_till": null,
        "sun_open_from": null,
        "sun_open_till": null,
        "max_jobs": 5,
        "is_premium": 0,
        "created_at": 1450951808,
        "updated_at": 1450951808
    },
    "contacts": {
        "id": 71,
        "user_id": 73,
        "address_line1": "12409 Lagoon Dr",
        "address_line2": "",
        "city": "Curtice",
        "state": "OH",
        "country_id": 233,
        "zipcode_id": 18749,
        "latitude": "41.655400",
        "longitude": "-83.246854",
        "web_url": "http://google.com",
        "alt_email": null,
        "contact_number": "1123456780",
        "alt_contact_number": null,
        "country": "United States",
        "zip": "43412"
    }
}
```

ACTIVATE A PROVIDER : A Provider will be activated as part of registration.

PUT /users/activate

PARAMETERS

account_activation_token	**required**	Account activation token *(available in CREATE response structure)*
activation_code	**required**	Activation Code *(available in CREATE response structure)*

Request Example:

Response Example:

```
curl -X PUT \
  --url
http://<domain>/api/web/v1/users/activate \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/x-www-form-
urlencoded' \
  -d
'account_activation_token=cNIO4lHbjd0oaJ8eAY
JVN_BFJEQUfI_7_1450951807&activation_code=50
0539'
```

```
HTTP/1.1 200 OK
Content-Type:application/json;charset=UTF-8
{
    "username": "api.provider",
    "email": "test@account.net",
    "status": "Active",
    "role": "Provider",
    "account_activation_token": null,
    "created_at": 1450951807,
    "updated_at": 1451293791,
    "id": 73,
    "role_numeric": 2,
    "status_numeric": 0,
    "profile": {
        "id": 29,
        "user_id": 73,
        "company_name": "API Inc",
        "contact_first_name": "API",
        "contact_last_name": "Provider",
        "company_logo_path":
"TfS0NGgb0f99a2GcypOSFhUaLFsk-h6W.png",
        "company_description": "This is API
Provider",
        "week_open_from": "10:00:00",
        "week_open_till": "18:00:00",
        "sat_open_from": null,
        "sat_open_till": null,
        "sun_open_from": null,
        "sun_open_till": null,
        "max_jobs": 5,
        "is_premium": 0,
        "created_at": 1450951808,
        "updated_at": 1450951808
    },
    "contacts": {
        "id": 71,
        "user_id": 73,
        "address_line1": "12409 Lagoon Dr",
        "address_line2": "",
        "city": "Curtice",
        "state": "OH",
        "country_id": 233,
        "zipcode_id": 18749,
        "latitude": "41.655400",
        "longitude": "-83.246854",
        "web_url": "http://google.com",
        "alt_email": null,
        "contact_number": "1123456780",
        "alt_contact_number": null,
        "country": "United States",
        "zip": "43412"
    }
}
```

Users

A user object keeps track of all user related information.

CREATE A USER : A new User will be created. | **POST** /users

PARAMETERS

`username`	**required**	Account login username of User
`email`	**required**	Email address of User
`password`	**required**	Account password
`cl_first_name`	**required**	First name of the user
`cl_last_name`	**required**	Last name of the user
`cl_allow_calls`	**required**	User opt-in status for receiving calls on provided mobile number
`address_line1`	optional	User street address
`address_line2`	optional	User street address additional
`city`	**required**	User city
`state`	**required**	User state
`zipcode`	**required**	User zipcode
`zipcode_id`	**required**	DB id of zipcode table corresponding to the `zipcode` above *(available in LIST response structure of **Zipcodes** API)*
`country_id`	**required**	DB id of country table *(available in LIST response structure of **Countries** API)*
`contact_number`	**required**	User mobile number
`role`	**required**	For User type user, role = 1 OR role = 'User'
`form_make_id`	**required**	DB id of profiles_car_make table *(available in LIST response structure of **Car Makes** API)*
`form_model_id`	**required**	DB id of profiles_car_model table *(available in LIST response structure of **Car Models** API)*
`form_year_id`	**required**	DB id of profiles_car_model_year table *(available in LIST response structure of **Car Years** API)*

Request Example:

```
curl --request POST \
  --url http://<domain>/api/web/v1/users \
  -H 'cache-control: no-cache' \
  -H 'content-type: multipart/form-data;
boundary=---011000010111000001101001' \
  -F username=api.user \
  -F email=abcdefg@account.net \
  -F password=passx123456 \
  -F cl_first_name=API \
  -F cl_last_name=User \
  -F cl_allow_calls=0 \
  -F 'address_line1=12409 Lagoon Drive' \
  -F address_line2=null \
  -F city=Curtice \
  -F state=OH \
  -F zipcode_id=18749 \
  -F country_id=233 \
  -F contact_number=1023456789 \
  -F role=User \
  -F zipcode=43412
  -F form_make_id=5
  -F form_model_id=33
  -F form_year_id=183
```

Response Example:

```
HTTP/1.1 201 Created
Content-Type:application/json;charset=UTF-8
Location:http://<domain>/api/web/v1/users/73
{
    "id": 68,
    "username": "api.user",
    "account_activation_token":
"sA6bXuhEqY7MKzGT68DwkZTEiEcRS-dd_1450774955",
    "email": "abcdefg@account.net",
    "role": "User",
    "status": "Inactive",
    "policy_agreed": 0,
    "created_at": 1450774955,
    "updated_at": 1450774955,
    "role_numeric": 1,
    "status_numeric": 0,
    "profile": {
        "id": 37,
        "user_id": 68,
        "client_first_name": "API",
        "client_last_name": "User",
        "client_description": null,
        "allow_calls": 0,
        "created_at": 1450774955,
```

```
        "updated_at": 1450774955
    },
    "contacts": {
        "id": 66,
        "user_id": 68,
        "address_line1": "12409 Lagoon Drive",
        "address_line2": "",
        "city": "Curtice",
        "state": "OH",
        "country_id": 233,
        "zipcode_id": 18749,
        "latitude": null,
        "longitude": null,
        "web_url": null,
        "alt_email": null,
        "contact_number": "1023456789",
        "alt_contact_number": null,
        "country": "United States",
        "zip": "43412"
    }
    "car_model_year_row_id": 8,
    "car": {
            "make": {
               "id": 12,
               "make_name": "Chevrolet",
               "status": 1
             },
             "model": {
               "id": 228,
               "model_name": "Cruze",
               "status": 1
             },
             "year": {
               "id": 1173,
               "year": "2013",
               "status": 1
             }
        }
}
```


ACTIVATE A USER : A User will be activated as part of registration.

PUT /users/activate

PARAMETERS

account_activation_token	**required**	Account activation token *(available in CREATE response structure)*
activation_code	**required**	Activation Code *(available in CREATE response structure)*

Request Example:

```
curl -X PUT \
  --url
http://<domain>/api/web/v1/users/activate \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/x-www-form-
urlencoded' \
  -d
'account_activation_token=sA6bXuhEqY7MKzGT68
DwkZTEiEcRSdd_1450774955&activation_code=245
457'
```

Response Example:


```
HTTP/1.1 200 OK
Content-Type:application/json;charset=UTF-8
{
    "id": 68,
    "username": "api.user",
    "account_activation_token": null,
    "email": "abcdefg@account.net",
    "role": "User",
    "status": "Active",
    "policy_agreed": 0,
    "created_at": 1450774955,
    "updated_at": 1450862598,
    "role_numeric": 1,
    "status_numeric": 1,
    "profile": {
        "id": 37,
        "user_id": 68,
```

```
        "client_first_name": "API",
        "client_last_name": "User",
        "client_description": null,
        "allow_calls": 0,
        "created_at": 1450774955,
        "updated_at": 1450774955
    },
    "contacts": {
        "id": 66,
        "user_id": 68,
        "address_line1": "12409 Lagoon Drive",
        "address_line2": "",
        "city": "Curtice",
        "state": "OH",
        "country_id": 233,
        "zipcode_id": 18749,
        "latitude": null,
        "longitude": null,
        "web_url": null,
        "alt_email": null,
        "contact_number": "1023456789",
        "alt_contact_number": null,
        "country": "United States",
        "zip": "43412"
    }
    "car_model_year_row_id": 8,
    "car": {
            "make": {
               "id": 12,
               "make_name": "Chevrolet",
               "status": 1
             },
             "model": {
               "id": 228,
               "model_name": "Cruze",
               "status": 1
             },
             "year": {
               "id": 1173,
               "year": "2013",
               "status": 1
             }
       }


}
```

Login

Ulafix allows REST clients to login with a user name and password using basic authentication.

LOGIN : A Provider/User will be logged in to the system.	**GET** /login

Request Example:

Simple Request

Most client software/program provides a simple mechanism for supplying a user name and password and will build the required authentication headers automatically. For example you can specify the -u argument with curl as follows:

```
curl -X GET \
      --url
http://<domain>/api/web/v1/login \
      -u fred:fred \
      -H 'cache-control: no-cache'
```

Supplying Basic Auth headers

If needed basic auth headers can be constructed and sent with the request. To do this the following steps needs to be performed:

- Build a string of the form *username*:*password*
- Base64 encode the string
- Supply an "Authorization" header with content "Basic " followed by the encoded string.

For example, the string "fred:fred" encodes to "ZnJlZDpmcmVk" in base64, so you would make the request as follows.

```
curl -X GET \
      --url
http://<domanin>/api/web/v1/login \
      -H 'authorization: Basic ZnJlZDpmcmVk'
\
      -H 'cache-control: no-cache'
```

Response Example:

```
HTTP/1.1 200 OK
Content-Type:application/json;charset=UTF-8
{
    "id": 73,
    "username": "api.provider",
    "access_token":
"Y5PM1dLFq7kdtcuPKsUBjaXdzj5gfQ82",
    "status": "Active",
    "status_numeric": 1,
    "role": "Provider",
    "role_numeric": 2
}

** All subsequent API calls (that require
authorization) need to send the above
access_token value using HTTP Bearer Auth
method for authorization.
```

Forgot Password

Ulafix allows REST clients to option to reset forgotten password using a two-step process:

1. Generate request to reset a password and receive a password reset token.
2. Send password reset token and new password, to actually reset the password in the system.

REQUEST PASSWORD RESET : A reset password request is generated — **POST** /password

PARAMETERS

`email`	**required**	Registered email id for account whose password will be reset

Request Example:

```
curl -X POST \
  --url http://<domain>/api/web/v1/password \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/x-www-form-urlencoded' \
  -d email=account%40testmail.com
```

Note: The parameter-value pairs must be sent to the API server as HTTP application/x-www-form-urlencoded.

In the above request example syntax, this is achieved by setting `content-type` header (`-H`) to `application/x-www-form-urlencoded` and the parameters along with their values are sent as data (`-d`).

Response Example:

```
HTTP/1.1 200 OK
Content-Type:application/json;charset=UTF-8
{
    "password_reset_token":
"2fMGc5LRneRJNu2FHeKlBy2Owq",
    "status": "Email sent with instructions
for resetting the password."
}

** The value of password_reset_token in
the above response needs to be used in the
next step to call the actual Reset Password
end-point along with new password value.
```

RESET PASSWORD : Actual reset password is done — **PUT** /password

PARAMETERS

`password_reset_token`	**required**	The password reset token received in previous API response
`password`	**required**	The new password that will be set for the account

Request Example:

```
curl -X PUT \
  --url http://<domain>/api/web/v1/password \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/x-www-form-urlencoded' \
  -d
'password_reset_token=2fMGc5LRneRJNu2FHeKlBy2Owq&password=abcd123456'
```

Note: The parameter-value pairs must be sent to the API server as HTTP application/x-www-form-urlencoded.

In the above request example syntax, this is achieved by setting `content-type` header (`-H`) to `application/x-www-form-urlencoded` and the parameters along with their values are sent as data (`-d`).

Response Example:

```
HTTP/1.1 200 OK
Content-Type:application/json;charset=UTF-8
{
    "status": "Password reset successfully"
}
```

Member Profile

Ulafix allows REST clients to perform various operations related to the Member's profile.

EDIT PROVIDER PROFILE : A Provider profile will be updated.	**POST** /users/*{user-id}*

PARAMETERS

`pr_first_name`	**required**	First name of the provider
`pr_last_name`	**required**	Last name of the provider
`pr_company_name`	**required**	Name of the company the provider represents
`address_line1`	**required**	Provider street address
`address_line2`	optional	Provider street address additional
`city`	**required**	Provider city
`state`	**required**	Provider state
`zipcode`	**required**	Provider zipcode
`zipcode_id`	**required**	DB id of zipcode table corresponding to the `zipcode` above *(available in LIST response structure of **Zipcodes** API)*
`country_id`	**required**	DB id of country table *(available in LIST response structure of **Countries** API)*
`contact_number`	**required**	Provider mobile number
`pr_company_description`	optional	Provider Company description
`pr_week_open_from`	optional	Provider working hour start time on weekdays
`pr_week_open_till`	optional	Provider working hour end time on weekdays
`pr_sat_open_from`	optional	Provider working hour start time on Saturdays
`pr_sat_open_till`	optional	Provider working hour end time on Saturdays
`pr_sun_open_from`	optional	Provider working hour start time on Sundays
`pr_sun_open_till`	optional	Provider working hour end time on Sundays
`pr_max_jobs`	optional	Max. no. of jobs provider can accept at a time *(defaults to 1)*
`pr_company_logo`	optional	Provider logo file
`pr_services[]`	**required**	DB id of service_type table for Providers' services; multiple values allowed *(available in LIST response structure of **Services** API)*
`latitude`	**required**	Latitude (in decimal) of Provider office location
`longitude`	**required**	Longitude (in decimal) of Provider office location
`web_url`	optional	Provider Web URL

Additional Info:

Authorization: REQUIRED

This end-point uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X POST \
  --url http://<domain>/api/web/v1/users/73 \
  -H 'authorization: Bearer 3451bJzwIqQjucbcpsHFu' \
  -H 'cache-control: no-cache' \
```

Response Example:

```
HTTP/1.1 200 OK
Content-Type:application/json;charset=UTF-8
{
    "id": 73,
    "username": "api.provider",
    "email": "test@account.net",
    "status": "Active",
    "role": "Provider",
    "account_activation_token": null,
    "created_at": 1450951807,
    "updated_at": 1450951807,
    "role_numeric": 2,
    "status_numeric": 1,
    "profile": {
        "id": 29,
        "user_id": 73,
        "company_name": "API Inc",
        "contact_first_name": "API",
```

```
  -H 'content-type: multipart/form-data;
boundary=---011000010111000001101001' \
  -F pr_first_name=API \
  -F pr_last_name=Provider \
  -F 'pr_company_name=API Inc' \
  -F 'address_line1=12409 Lagoon Dr' \
  -F address_line2=null \
  -F city=Curtice \
  -F state=OH \
  -F zipcode_id=18749 \
  -F country_id=233 \
  -F contact_number=1123456780 \
  -F role=Provider \
  -F zipcode=43412 \
  -F 'pr_company_description=Edited API
Provider' \
  -F pr_week_open_from=10:00 \
  -F pr_week_open_till=18:00 \
  -F pr_sat_open_from=null \
  -F pr_sat_open_till=null \
  -F pr_sun_open_from=null \
  -F pr_sun_open_till=null \
  -F pr_max_jobs=5 \
  -F pr_company_logo=@abcd.png \
  -F 'pr_services[]=7' \
  -F 'pr_services[]=12' \
  -F latitude=41.655400 \
  -F longitude=-83.246854 \
  -F web_url=http://google.com
```

Note: The parameter-value pairs must be sent to the API server as HTTP multipart/form-data.

In the above request example syntax, this is achieved by setting `content-type` header (`-H`) to `multipart/form-data` and the parameters along with their values are sent as form data (`-F`).
Also, the user_id value passed in URL must be same as that of the user whose access token is being sent as authorization.

```
        "contact_last_name": "Provider",
        "company_logo_path":
"TfS0NGgb0f99a2GcypOSFhUaLFsk-h6W.png",
        "company_description": "Edited API
Provider",
        "week_open_from": "10:00:00",
        "week_open_till": "18:00:00",
        "sat_open_from": null,
        "sat_open_till": null,
        "sun_open_from": null,
        "sun_open_till": null,
        "max_jobs": 5,
        "is_premium": 0,
        "created_at": 1450951808,
        "updated_at": 1450951808
    },
    "contacts": {
        "id": 71,
        "user_id": 73,
        "address_line1": "12409 Lagoon Dr",
        "address_line2": "",
        "city": "Curtice",
        "state": "OH",
        "country_id": 233,
        "zipcode_id": 18749,
        "latitude": "41.655400",
        "longitude": "-83.246854",
        "web_url": "http://google.com",
        "alt_email": null,
        "contact_number": "1123456780",
        "alt_contact_number": null,
        "country": "United States",
        "zip": "43412"
    },
    "services": [
        {
            "id": 75,
            "service_type_id": 7,
                    ...
        },
        ...
    ],
}
```

EDIT USER PROFILE : A User profile will be updated. | **POST** /users/*{user-id}*

PARAMETERS

`cl_first_name`	**required**	First name of the user
`cl_last_name`	**required**	Last name of the user
`cl_allow_calls`	**required**	User opt-in status for receiving calls on provided mobile number
`address_line1`	**required**	User street address
`address_line2`	optional	User street address additional
`city`	**required**	User city
`state`	**required**	User state
`zipcode`	**required**	User zipcode
`zipcode_id`	**required**	DB id of zipcode table corresponding to the `zipcode` above *(available in LIST response structure of **Zipcodes** API)*
`country_id`	**required**	DB id of country table *(available in LIST response structure of **Countries** API)*

`contact_number`	**required**	User mobile number
`car_model_year_row_id`	**required**	It is mandatory field to get update the existing Car Make/Model/Year entry. e.g `car_model_year_row_id = <id>` *(Please refer to the User list API to get the* `car_model_year_row_id` *)* Please note if `car_model_year_row_id is Null or Empty` then it will insert a new record for that user into user_car table.
`form_make_id`	**required**	DB id of profiles_car_make table *(available in LIST response structure of **Car Makes** API)*
`form_model_id`	**required**	DB id of profiles_car_model table *(available in LIST response structure of **Car Models** API)*
`form_year_id`	**required**	DB id of profiles_car_model_year table *(available in LIST response structure of **Car Years** API)*

Additional Info:

Authorization: REQUIRED

This end-point uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

authorization: Bearer ***<token>***

Request Example:

```
curl --request POST \
  --url http://<domain>/api/web/v1/users/68
\
  -H 'authorization: Bearer
3451bJzwIqQjucbcpsHFu' \
  -H 'cache-control: no-cache' \
  -H 'content-type: multipart/form-data;
boundary=---011000010111000001101001' \
  -F cl_first_name=API \
  -F cl_last_name=User \
  -F cl_allow_calls=0 \
  -F 'address_line1=12409 Lagoon Drive' \
  -F address_line2=null \
  -F city=Curtice \
  -F state=OH \
  -F zipcode_id=18749 \
  -F country_id=233 \
  -F contact_number=1023456789 \
  -F role=User \
  -F zipcode=43412
```

Note: The parameter-value pairs must be sent to the API server as HTTP multipart/form-data.

In the above request example syntax, this is achieved by setting `content-type` header (`-H`) to `multipart/form-data` and the parameters along

Response Example:

```
HTTP/1.1 200 OK
Content-Type:application/json;charset=UTF-8
{
    "id": 68,
    "username": "api.user",
    "account_activation_token": null,
    "email": "abcdefg@account.net",
    "role": "User",
    "status": "Active",
    "policy_agreed": 1,
    "created_at": 1450774955,
    "updated_at": 1450774955,
    "role_numeric": 1,
    "status_numeric": 1,
    "profile": {
        "id": 37,
        "user_id": 68,
        "client_first_name": "API",
        "client_last_name": "User",
        "client_description": null,
        "allow_calls": 0,
        "created_at": 1450774955,
        "updated_at": 1450774955
    },
    "contacts": {
        "id": 66,
        "user_id": 68,
        "address_line1": "12409 Lagoon Drive",
        "address_line2": "",
        "city": "Curtice",
        "state": "OH",
        "country_id": 233,
        "zipcode_id": 18749,
        "latitude": null,
        "longitude": null,
        "web_url": null,
        "alt_email": null,
        "contact_number": "1023456789",
        "alt_contact_number": null,
        "country": "United States",
        "zip": "43412"
```

with their values are sent as form data (`-F`).
Also, the user_id value passed in URL must be same as that of the user whose access token is being sent as authorization.


```
    },
    "car_model_year_row_id": 8,
    "car": {
            "make": {
               "id": 12,
               "make_name": "Chevrolet",
               "status": 1
             },
             "model": {
               "id": 228,
               "model_name": "Cruze",
               "status": 1
             },
             "year": {
               "id": 1173,
               "year": "2013",
               "status": 1
             }
    }
}
```


UPDATE PASSWORD : Change password of member (user, provider) | **PUT** /users/*{user-id}*/password

PARAMETERS

`New_password`	**required**	The new password that will be set for the account

Request Example:

```
curl -X PUT \
  --url
http://<domain>/api/web/v1/users/22/password \
  -H 'authorization: Bearer
qwe12bJzwIqQjucbcpsaqHFu' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/x-www-form-
urlencoded' \
  -d  'new_password=abcd123456'
```

Response Example:

```
HTTP/1.1 200 OK
Content-Type:application/json;charset=UTF-8
{
    "status": "Password updated
successfully"
}
```

Provider Search & Full Profile

Ulafix allows REST clients to perform Provider Search based on criteria like car make/model/year, zipcodes, schedule, service type etc.

SEARCH : Search providers based on search criteria		GET /companies/search
PARAMETERS (QUERY STRING)		
`service_type_id`	**required**	DB id of service_type table for services *(available in LIST response structure of **Services** API)*
`search_radius`	optional	Integer value of radius of search from given zipcode
`zipcode`	**required**	DB id of zipcode table *(available in LIST/SEARCH response structure of **Zipcode** API)*
`from_schedule`	**required**	Schedule from date-time in specific format (e.g. 2016-01-27 15:00)
`to_schedule`	**required**	Schedule to date-time in specific format (e.g. 2016-01-27 16:00)
`car_make_id`	**required**	DB id of car_make table for car makes *(available in LIST response structure of **Car Makes** API)*
`car_model_id`	**required**	DB id of car_model table for car models *(available in LIST response structure of **Car Models** API)*
`car_year_id`	**required**	DB id of car_model_year table for car years *(available in LIST response structure of **Car Years** API)*

Request Example:

```
curl -X GET \
      --url
http://<domain>/api/web/v1/companies/search?service_type_id=7&search_radius=100&zipcode=38896&from_schedule=2016-01-27
15:00&to_schedule=2016-01-27
16:00&car_make_id=2&car_model_id=15&car_year_id=123 \
      -H 'cache-control: no-cache'
```

Using the `fields` parameters, you may also specify which fields should be included in the result.

For example, the URL

`http://<domain>/api/web/v1/countries/search?service_type_id=7&search_radius=`*`100`*`&zipcode=`*`92102`*`&from_schedule=`*`2016-01-27 15:00`*`&to_schedule=`*`2016-01-27 16:00`*`&car_make_id=`*`2`*`&car_model_id=`*`15`*`&car_year_id=`*`123`*`&`**`fields`**`=`*`user_id,account`*

will only return the `user_id` and `account` fields in the response.

Response Example:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "user_id": 22,
            "company_name": "Greenbird Media Inc.",
            "contact_first_name": "Nick",
            "contact_last_name": "Kondoori",
            "company_logo_path":
"http://<domain>/frontend/assets/uploads/company/logo/22/sqr__toALO3LEoGtBN53GJPPLbJ7G_VYa9Tc.jpg",
            "company_description": "Description...",
            "week_open_from": "09:00:00",
            "week_open_till": "17:00:00",
            "sat_open_from": "09:30:00",
            "sat_open_till": "13:30:00",
            "sun_open_from": null,
            "sun_open_till": null,
            "max_jobs": 2,
            "is_premium": 1,
            "created_at": 1439278924,
            "updated_at": 1450426471,
            "account": {
                "id": 22,
                "username": "souvik.provider",
                "account_activation_token": null,
                "email": "souvik@greenbirdit.com",
                "role": "Provider",
                "status": "Active",
                "policy_agreed": 1,
                "created_at": 1439278924,
                "updated_at": 1451389645,
                "role_numeric": 2,
                "status_numeric": 1
            },
            "contacts": {
```

```
                    "id": 20,
                    "user_id": 22,
          "address_line1": "7860 Mission Center Court",
                    "address_line2": "Suite# 103",
                    "city": "San Diego",
                    "state": "CA",
                    "country_id": 233,
                    "zipcode_id": 38896,
                    "latitude": "32.773896",
                    "longitude": "-117.155067",
                    "web_url": "http://abc.com",
                    "alt_email": null,
                    "contact_number": "1879054967",
                    "alt_contact_number": null,
                    "country": "United States",
                    "zip": "92102"
                },
                "services": [
                    {
                        "id": 32,
                        "service_type_id": 12,
                        "status": "Active",
                        "created_at": 1440076257,
                        "status_numeric": 1,
                        "service_name": "Transmission
Repair"
                    },
                    {
                        "id": 33,
                        "service_type_id": 14,
                        "status": "Active",
                        "created_at": 1440076257,
                        "status_numeric": 1,
                        "service_name": "Engine Service"
                    },
                    {
                        "id": 34,
                        "service_type_id": 4,
                        "status": "Active",
                        "created_at": 1440076257,
                        "status_numeric": 1,
                        "service_name": "Towing"
                    },
                    {
                        "id": 35,
                        "service_type_id": 7,
                        "status": "Active",
                        "created_at": 1440076337,
                        "status_numeric": 1,
                        "service_name": "Oil Change"
                    }
                ],
                "jobCount": "11",
                "reviewCount": "4"
            },
            ...
             ...
            {
                "user_id": 24,
                 ...
            }
        ]
}
```

PROVIDER FULL PROFILE : Get profile details of a particular provider	**GET** /companies/{user_id}

Request Example:

```
curl -X GET \
      --url
http://<domain>/api/web/v1/companies/22 \
      -H 'cache-control: no-cache'
```

Using the `fields` parameters, you may also specify which fields should be included in the result.
For example, the URL
`http://<domain>/api/web/v1/companies/22&`**`fields`**`=user_id,account`
will only return the `user_id` and `account` fields in the response.

Using the `expand` parameters, you may also specify which additional related fields should be added in the result. This end-point supports following two expand parameters:

- **reviews**: includes user reviews for the provider with same structure as **Reviews API** response
- **jobs**: includes jobs details of the provider
- **jobs.bids:** includes jobs details of provider along with bid information for each job. So `jobs` and `jobs.bids` should not be used together in same request

Example usage:
`http://<domain>/api/web/v1/companies/22&`**`expand`**`=reviews,jobs.bids`
will add `reviews` and `jobs.bids` details in the response structure.

Response Example:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
{
    "user_id": 22,
    "company_name": "Greenbird Media Inc.",
    "contact_first_name": "Nick",
    "contact_last_name": "Kondoori",
       ...
    "account": {
        "id": 22,
        "username": "souvik.provider",
              ...
    },
    "contacts": {
        "id": 20,
        "user_id": 22,
        "address_line1": "7860 Mission Center
Court",
              ...
    },
    "services": [
        {
            "id": 32,
            "service_type_id": 12,
                    ...
        },
        ...
              ...
    ],
    "jobCount": "11",
    "reviewCount": "4",
}
```

EXPAND Parameter Response Example:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
{
    "user_id": 22,
    "company_name": "Greenbird Media Inc.",
    "contact_first_name": "Nick",
    "contact_last_name": "Kondoori",
     ...
    "account": {
        "id": 22,
        "username": "souvik.provider",
         ...
    },
    "contacts": {
        "id": 20,
        "user_id": 22,
        "address_line1": "7860 Mission Center",
         ...
    },
    "services": [
        {
            "id": 32,
            "service_type_id": 12,
              ...
        },
        ...
```

```
            ...
        ],
        "jobCount": "11",
        "reviewCount": "4",
        "reviews": [
            {
                "id": 16,
                "review_for": 22,
                "review_by_id": 46,
                "review_by": "Tester Integrated",
                "job_id": 29,
                "rating": "5.00",
                "review": "Excellent services!",
                "for_job_type": "Transmission Repair",
                "timestamp": 1443179948
            },
            ...
            ...
        ],
        "jobs.bids": [
            {
                "id": 141,
                "user_id": 22,
                "job_by_id": 1,
                "job_by": "GBIT Souvik",
                "job_id": 52,
                "job_service_type_id": 7,
                "job_service_type": "Oil Change",
                "job_zipcode_id": 38897,
                "job_zipcode": "92103",
                "search_radius": 100,
                "job_datetime": "2016-01-08 12:30:00",
                "job_name": "API job#2 050116",
                "job_details": "API job#2 050116",
                "distance_from_job_zipcode": "3.615777",
                "distance_unit": "mi",
                "status_numeric": 2,
                "status": "Bid",
                "job_timestamp": 1451989973,
                "bids": [
                    {
                        "id": 75,
                        "parent_bid_id": 0,
                        "job_id": 52,
                        "job_created_by": 1,
                        "job_bid_by": 22,
                        "bid_amount": "150.00",
                        "suggested_job_start_datetime":
"2016-01-08 10:30:00",
                        "duration": "1.50",
                        "comments": "2nd Bid",
                        "is_accepted": 0,
                        "created_at": 1452085267
                    },
                    ...

                ]
            },
            ...
            ...
        ]
}
```

User (Client) Service Requests

Ulafix allows REST clients to create a service request from a User.

CREATE SERVICE REQUEST : A new service request (job) will be created. | **POST** /client-jobs

PARAMETERS

`user_id`	**required**	id of User under whose account service request will be created; *(available in response structure of **Login** API on successful login)*
`job_name`	**required**	Job name for easy identification
`job_dt_choice_from`	**required**	Schedule from date-time in specific format (e.g. 2016-01-27 15:00)
`job_dt_choice_to`	**required**	Schedule to date-time in specific format (e.g. 2016-01-27 16:00)
`form_service_type`	**required**	DB id of service_type table for services *(available in LIST response structure of **Services** API)*
`form_zipcode`	**required**	DB id of zipcode table *(available in LIST/SEARCH response structure of **Zipcode** API)*
`search_radius`	optional	Integer value of radius of search from given zipcode
`job_description`	optional	Job description
`form_make_id`	**required**	DB id of car_make table for car makes *(available in LIST response structure of **Car Makes** API)*
`form_model_id`	**required**	DB id of car_model table for car models *(available in LIST response structure of **Car Models** API)*
`car_model_year_id`	**required**	DB id of car_model_year table for car years *(available in LIST response structure of **Car Years** API)*

Additional Info:

Authorization: REQUIRED

This end-point uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X POST \
  --url http://<domain>/api/web/v1/client-jobs \
  -H 'authorization: Bearer 5bJzwIqQjucbcxHFu' \
  -H 'cache-control: no-cache' \
  -H 'content-type: multipart/form-data; boundary=---011000010111000001101001' \
  -F user_id=1 \
  -F 'job_name=New API job#1 2701' \
  -F 'job_dt_choice_from=2016-01-27 16:00' \
  -F 'job_dt_choice_to=2016-01-27 18:00' \
  -F form_service_type=7 \
  -F form_zipcode=38897 \
  -F search_radius=10 \
  -F 'job_description=New API job descrip' \
  -F form_make_id=2 \
  -F form_model_id=15 \
  -F car_model_year_id=122
```

Response Example:

```
HTTP/1.1 201 Created
Content-Type:application/json;charset=UTF-8
Location:http://<domain>/api/web/v1/client-jobs/61
```


```
{
    "id": 61,
    "user_id": 1,
    "service_type_id": 7,
    "job_name": "New API job#1 2701",
    "job_description": "New API job descrip",
    "job_datetime": null,
    "job_dt_choice_from":"2016-01-27 16:00:00",
    "job_dt_choice_to": "2016-01-27 18:00:00",
    "zipcode_id": 38897,
    "search_radius": 10,
    "status": 1,
    "created_at": 1453888861,
    "updated_at": 1453888861,
    "status_text": "STATUS_CURRENT_JOBS",
    "service_type": "Oil Change",
    "car": {
        "car_model_year_id": 122,
        "car_model_year": "1993",
        "car_model_id": 15,
        "car_model": "Spider",
        "car_make_id": 2,
        "car_make": "Alfa Romeo"
    },
    "zipcode": "92103",
    "jobcode": 1453888861,
    "bid_win_by": null,
    "bid_win": null,
    "bid_count": "0"
```

Note: The parameter-value pairs must be sent to the API server as HTTP multipart/form-data.

In the above request example syntax, this is achieved by setting `content-type` header (`-H`) to `multipart/form-data` and the parameters along with their values are sent as form data (`-F`).
Also, the user_id value must be the same as that of the user whose access token is being sent as authorization.

```
}
```

RESEND SERVICE REQUEST : A new service request (job) will be created while deleting the existing job whose id is passed in the URL.

POST /client-jobs/*{job-id}*

PARAMETERS

`user_id`	**required**	id of User under whose account service request will be created; *(available in response structure of **Login** API on successful login)*
`job_name`	**required**	Job name for easy identification
`job_dt_choice_from`	**required**	Schedule from date-time in specific format (e.g. 2016-01-27 15:00)
`job_dt_choice_to`	**required**	Schedule to date-time in specific format (e.g. 2016-01-27 16:00)
`form_service_type`	**required**	DB id of service_type table for services *(available in LIST response structure of **Services** API)*
`form_zipcode`	**required**	DB id of zipcode table *(available in LIST/SEARCH response structure of **Zipcode** API)*
`search_radius`	optional	Integer value of radius of search from given zipcode
`job_description`	optional	Job description
`form_make_id`	**required**	DB id of car_make table for car makes *(available in LIST response structure of **Car Makes** API)*
`form_model_id`	**required**	DB id of car_model table for car models *(available in LIST response structure of **Car Models** API)*
`car_model_year_id`	**required**	DB id of car_model_year table for car years *(available in LIST response structure of **Car Years** API)*

Additional Info:

Authorization: REQUIRED

This end-point uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X POST \
  --url http://<domain>/api/web/v1/client-jobs/61 \
  -H 'authorization: Bearer 5bJzwIqQjucbcxHFu' \
  -H 'cache-control: no-cache' \
  -H 'content-type: multipart/form-data; boundary=---011000010111000001101001' \
  -F user_id=1 \
  -F 'job_name=New API job#1 2701 resend' \
```

Response Example:

```
HTTP/1.1 201 Created
Content-Type:application/json;charset=UTF-8
Location:http://<domain>/api/web/v1/client-jobs/62
{
    "id": 61,
    "user_id": 1,
    "service_type_id": 7,
    "job_name": "New API job#1 2701 resend",
    "job_description": "New API job#1 2701 resend, job is created through API",
    "job_datetime": null,
    "job_dt_choice_from": "2016-01-27 16:00:00",
    "job_dt_choice_to": "2016-01-27 18:00:00",
    "zipcode_id": 38897,
    "search_radius": 10,
    "status": 1,
    "created_at": 1453888861,
    "updated_at": 1453888861,
    "status_text": "STATUS_CURRENT_JOBS",
    "service_type": "Oil Change",
```

<table>
<tr>
<td>

```
  -F 'job_dt_choice_from=2016-01-27 16:00' \
  -F 'job_dt_choice_to=2016-01-27 18:00' \
  -F form_service_type=7 \
  -F form_zipcode=38897 \
  -F search_radius=10 \
  -F 'job_description= New API job#1 2701
resend, job is created through API' \
  -F form_make_id=2 \
  -F form_model_id=15 \
  -F car_model_year_id=122
```

Note: The existing job, whose job-id is passed in the URL, will get deleted and a new job will be created.

</td>
<td>

```
    "car": {
        "car_model_year_id": 122,
        "car_model_year": "1993",
        "car_model_id": 15,
        "car_model": "Spider",
        "car_make_id": 2,
        "car_make": "Alfa Romeo"
    },
    "zipcode": "92103",
    "jobcode": 1453888861,
    "bid_win_by": null,
    "bid_win": null,
    "bid_count": "0"
}
```

</td>
</tr>
<tr>
<td>CLOSE JOB LIST : Get all close job list to implement logic to force client to closed a job.</td>
<td>GET /client-jobs /job-closed</td>
</tr>
<tr>
<td>

Request Example:

```
curl -X GET \
  --url http://<domain>/api/web/v1/client-
jobs/job-closed \
  -H 'authorization: Bearer
y8ZSJYBi0DrU2q1IwYLenDvVco7U1ylH' \
  -H 'cache-control: no-cache'
```

Note: You can customize your logic by using the given list to force a client to closed a job. Here `company_job_status` is conditionally using to implement the logic to the client.

Please refer the Client activity and Provider activity API to understand the status.

</td>
<td>

Response Example:

```
HTTP/1.1 200 OK
Content-Type:application/json;charset=UTF-8
[
  {
    "job_id": 39,
    "job_created_by": 79,
    "job_name": "5K or 6 month service-
25Apr2016",
    "client_job_status":3,
    "job_awarded_to": 45,
    "company_job_status": 4,
  }
]
```

</td>
</tr>
</table>

Services Bidding & Acceptance

Ulafix allows REST clients to create bids and manage the entire life-cycle of a bid on a service request.

CREATE BID : A new bid will be created for a particular job.	**POST** /bids

PARAMETERS

`job_id`	**required**	id of Job for which this bid will be created; *(available in LIST response structure of **Provider Activity** API)*
`job_created_by`	**required**	id of User who created Job for which this bid will be created *(available in LIST response of **Provider Activity** API as `client_id` field)*
`job_bid_by`	**required**	id of Provider who is placing the bid; *(available in response of **Login** API after login as `id` field)*
`bid_amount`	**required**	Bid amount in decimal (e.g. 25, 20.00, 10.99 etc)
`suggested_job_start_datetime`	**required**	Suggested time when Provider can start working on this job in a specific format (2016-01-27 16:00)
`duration`	**required**	Estimated duration to complete the job in hours (e.g. 1, 1.5 etc)
`comments`	optional	Additional remarks if any

Additional Info:

Authorization: REQUIRED

This end-point uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X POST \
  --url http://<domain>/api/web/v1/bids \
  -H 'authorization: Bearer X_J8kwflDQH7sTb' \
  -H 'cache-control: no-cache' \
  -H 'content-type: multipart/form-data; boundary=---011000010111000001101001' \
  -F job_id=61 \
  -F job_created_by=1 \
  -F job_bid_by=22 \
  -F bid_amount=70 \
  -F 'suggested_job_start_datetime=2016-01-27 16:30:00' \
  -F duration=1.5 \
  -F 'comments=Bid from API'
```

Note: The parameter-value pairs must be sent to the API server as HTTP multipart/form-data.

In the above request example syntax, this is achieved by setting `content-type` header (`-H`) to `multipart/form-data` and the parameters along with their values are sent as form data (`-F`).
Also, the job_bid_by value must be the same as that of user whose access token is being sent as authorization.

Response Example:

```
HTTP/1.1 201 Created
Content-Type:application/json;charset=UTF-8
Location:http://<domain>/api/web/v1/bids/82
{
    "job_id": "61",
    "job_created_by": "1",
    "job_bid_by": "22",
    "bid_amount": "70",
    "suggested_job_start_datetime": "2016-01-27 16:30:00",
    "duration": "1.5",
    "comments": "Bid from API",
    "is_accepted": false,
    "parent_bid_id": 0,
    "created_at": 1454323475,
    "id": 82,
    "is_accepted_numeric": 0
}
```

REVISE BID : A new bid will be created with a reference to previous bid. | **POST** /bids

PARAMETERS

`job_id`	**required**	id of Job for which this bid will be created; *(available in LIST response structure of **Provider Activity** API)*
`job_created_by`	**required**	id of User who created Job for which this bid will be created *(available in LIST response of **Provider Activity** API as `client_id` field)*
`job_bid_by`	**required**	id of Provider who is placing the bid; *(available in response of **Login** API after login as `id` field)*
`bid_amount`	**required**	Bid amount in decimal (e.g. 25, 20.00, 10.99 etc)
`suggested_job_start_datetime`	**required**	Suggested time when Provider can start working on this job in a specific format (2016-01-27 16:00)
`duration`	**required**	Estimated duration to complete the job in hours (e.g. 1, 1.5 etc)
`comments`	optional	Additional remarks if any
`parent_bid_id`	**required**	DB id of job_bids table for latest bid by the provider on this job *(available in DETAIL response structure of **Client Jobs** API)*

Additional Info:

Authorization: REQUIRED

This end-point uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X POST \
  --url http://<domain>/api/web/v1/bids \
  -H 'authorization: Bearer X_J8kwflDQH7sTb' \
  -H 'cache-control: no-cache' \
  -H 'content-type: multipart/form-data; boundary=---011000010111000001101001' \
  -F job_id=61 \
  -F job_created_by=1 \
  -F job_bid_by=22 \
  -F bid_amount=50.00 \
  -F 'suggested_job_start_datetime=2016-01-27 16:30:00' \
  -F duration=1.5 \
  -F 'comments=Revised Bid' \
  -F parent_bid_id=82
```

Note: The parameter-value pairs must be sent to the API server as HTTP multipart/form-data.

In the above request example syntax, this is achieved by setting `content-type` header (`-H`) to `multipart/form-data` and the parameters along with their values are sent as form data (`-F`).
Also, the job_bid_by value must be the same as that of user whose access token is being sent as authorization.

Response Example:

```
HTTP/1.1 201 Created
Content-Type:application/json;charset=UTF-8
Location:http://<domain>/api/web/v1/bids/83
{
    "job_id": "61",
    "job_created_by": "1",
    "job_bid_by": "22",
    "bid_amount": "50.00",
    "suggested_job_start_datetime": "2016-01-27 16:30:00",
    "duration": "1.5",
    "comments": "Revised Bid",
    "is_accepted": false,
    "parent_bid_id": 82,
    "created_at": 1454326237,
    "id": 83,
    "is_accepted_numeric": 0
}
```

LIST BIDS : All bids placed by particular provider will be listed.

GET /bids

Authorization: REQUIRED
The API uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:
*authorization: Bearer **<token>***

Request Example:

```
curl -X GET \
  --url
http://<domain>/api/web/v1/
bids \
  -H 'authorization: Bearer
5bJzwIqQjucbcxH7Ri2PnzBj' \
  -H 'cache-control: no-
cache'
```

Note: If access_token sent as authorization is for

- User - then bids listed will be those that were placed on jobs this User created
- Provider - then bids listed will be those that were placed by this Provider

Response Example:

```
HTTP/1.1 200 OK
X-Pagination-Total-Count: 48
X-Pagination-Page-Count: 3
X-Pagination-Current-Page: 1
X-Pagination-Per-Page: 20
Link: <http://<domain>/api/web/v1/bids?page=1>; rel=self,
      <http://<domain>/api/web/v1/bids?page=2>; rel=next,
      <http://<domain>/api/web/v1/bids?page=3>; rel=last
Transfer-Encoding: chunked
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 1,
            "parent_bid_id": 0,
            "job_id": 14,
            "job_created_by": 1,
            "job_bid_by": 22,
            "bid_amount": "350.00",
            "suggested_job_start_datetime": "2015-08-24
13:00:00",
            "duration": "1.00",
            "comments": "First bid on Latest Test Job 2108",
            "is_accepted": false,
            "created_at": 1441104825,
            "is_accepted_numeric": 0
        },
        ...
        ...
    ],
    "_links": {
        "self": {
            "href": "http://<domain>/api/web/v1/bids?page=1"
        },
        "next": {
            "href": "http://<domain>/api/web/v1/bids?page=2"
        },
        "last": {
            "href": "http://<domain>/api/web/v1/bids?page=3"
        }
    },
    "_meta": {
        "totalCount": 48,
        "pageCount": 3,
        "currentPage": 1,
        "perPage": 20
    }
}
```

PICK BID : A bid is picked by User and Job is awarded to corresponding Provider who placed that bid & also a job is added to the calendar as per the bid details

PUT /bids/{bid-id}

PARAMETERS

`pay_method`	**required**	Payment Method - either `cash` or `online`

Additional Info:
Authorization: REQUIRED
This end-point uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a*

Response Example:

```
HTTP/1.1 200 OK
Content-
Type:application/json;charset=UTF-8
{
```

successful login request) to be sent in the request as part of the HTTP header in following format:
authorization: Bearer ***<token>***

Request Example:

```
curl -X PUT \
  --url http://<domain>/api/web/v1/bids/83 \
  -H 'authorization: Bearer X_J8kwflDQH7sTb' \
  -H 'cache-control: no-cache'
  -H 'content-type: application/x-www-form-urlencoded' \
  -d pay_method=cash
```

Note: the bid can only be picked if the job, against which this bid was placed, had been created by the user whose access token is being sent as authorization.

Note: For online payment, the response will contain the bid object and an RSA key which will be used to encrypt relevant data in the mobile app, before sending over to ccAvenue server. For more details refer to ccAvenue Documentation.


```
  "bid": {
    "id": 83,
    "parent_bid_id": 82,
    "job_id": 61,
    "job_created_by": 1,
    "job_bid_by": 22,
    "bid_amount": "50.00",
    "suggested_job_start_datetime":
"2016-01-27 16:30:00",
    "duration": "1.50",
    "comments": "Revised bid",
    "is_accepted": true,
    "pay_method": 1,
    "created_at": 1465369896,
    "updated_at": 1465378097,
    "is_accepted_numeric": 1
  },
  "payment": {
    "user_id": 1,
    "provider_id": 22,
    "bid_id": 83,
    "job_id": 61,
    "order_status": "Success",
    "payment_mode": "Cash",
    "currency": "INR",
    "amount": 57.50,
    "tax_amount": 7.5,
    "created_at": 1465378097,
    "updated_at": 1465378097,
    "id": 26
  }
}
```


SEARCH BIDS : Search & list bids based on query string

GET /bids/search

PARAMETERS (QUERY STRING)

This end-point supports LIKE search or partial-match search using following field as parameter

`q`	**required**	query/search parameter to match with Job code

This end-point also supports exact search using following fields as parameter

`job_id`	optional	query/search parameter to match with Job Id
`is_accepted`	optional	query/search parameter to match with is_accepted flag of bid

Authorization: REQUIRED

The API uses HTTP Bearer Authentication and requires `access_token` *(which is available in the* ***Login API*** *response after a successful login request)* to be sent in the request as part of the HTTP header in following format:
authorization: Bearer ***<token>***

Request Example:

1)

```
curl -X GET \
  --url
http://<domain>/api/web/v1/bids/search?
q=1440149145 \
  -H 'authorization: Bearer
5bJzwIqQjucbcxH7Ri2PnzBj' \
  -H 'cache-control: no-cache'
```

2)

```
curl -X GET \
```

Response Example:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
```


```
{
    "items": [
        {
            "id": 18,
            "parent_bid_id": 12,
            "job_id": 14,
            "job_created_by": 1,
            "job_bid_by": 22,
            "bid_amount": "236.00",
            "suggested_job_start_datetime": "2015-
08-24 13:00:00",
            "duration": "1.00",
            "comments": "",
            "is_accepted": true,
            "created_at": 1441349174,
            "is_accepted_numeric": 1
        }
    ]
```

```
  --url
http://<domain>/api/web/v1/bids/search?
q=1440149145&is_accepted=1 \
  -H 'authorization: Bearer
5bJzwIqQjucbcxH7Ri2PnzBj' \
  -H 'cache-control: no-cache'
```

Note: If access_token sent as authorization is for

- User - then bids listed will be those that were placed on jobs this User created
- Provider - then bids listed will be those that were placed by this Provider

Using the `expand` parameters, you may also specify which additional related fields should be added in the result. This end-point supports following expand parameters:

- **bid_by_company**: includes user related info of Provider who placed bid; the structure is same as **User API** response
- **created_by_user**: includes user related info of User who created the job; the structure is same as **User API** response
- **job**: includes job related info on which the bid is placed; the structure is same as **Job API** response

Example usage:
`http://<domain>/api/web/v1/premium-companies/search?q=1440149145&is_accepted=1&`**`expand`**`=job`
will add `job` details in the response structure.

```
}
```

EXPAND Parameter Response Example:


```
{
    "items": [
        {
            "id": 18,
            "parent_bid_id": 12,
            "job_id": 14,
            "job_created_by": 1,
            "job_bid_by": 22,
            "bid_amount": "236.00",
             ...
            "job": {
                "id": 14,
                "user_id": 1,
                "service_type_id": 7,
                "job_name": "Latest Test Job 2108",
                "job_description": "Test job",
                ...
                "service_type": "Oil Change",
                "car": {
                    "car_model_year_id": 1078,
                    "car_model_year": "1995",
                    ...
                },
                "zipcode": "92025",
                "jobcode": 1440149145,
                "bid_win_by": "Greenbird Media
Inc.",
                "bid_win": {
                    "bid_amount": "236.00",
                    "duration": "1.00"
                },
                "bid_count": "8"
            }
        }
    ]
}
```

Messages

Ulafix allows REST clients to create, list & delete private messages between User & Provider related to a job.

CREATE MESSAGE : A new message will be created for a particular job. | **POST** /messages

PARAMETERS

job_id	**required**	id of Job to which this message will be associated;
from_user_id	**required**	user_id of User or Provider who is sending the message
to_user_id	**required**	user_id of User or Provider for whom the message is intended
message	**required**	Content of the message

Additional Info:

Authorization: REQUIRED

This end-point uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X POST \
  --url http://<domain>/api/web/v1/messages \
  -H 'authorization: Bearer X_J8kwflDQH7sTb' \
  -H 'cache-control: no-cache' \
  -H 'content-type: multipart/form-data;
boundary=---011000010111000001101001' \
  -F job_id=61 \
  -F from_user_id=1 \
  -F to_user_id=22 \
  -F 'message=Message added through API'
```

The from_user_id value must be the same as that of user whose access token is being sent as authorization.

Response Example:

```
HTTP/1.1 201 Created
Content-Type:application/json;charset=UTF-8
Location:http://<domain>/api/web/v1/messages/85
{
    "from_user_id": "1",
    "to_user_id": "22",
    "job_id": "61",
    "message": "Message added through API",
    "created_at": 1454398978,
    "id": 85
}
```

DELETE MESSAGE : A message will be deleted | **DEL** /messages/{message-id}

Additional Info:

Authorization: REQUIRED

This end-point uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X DELETE \
  --url
http://<domain>/api/web/v1/messages/85 \
  -H 'authorization: Bearer X8kwflDQH7sTb' \
  -H 'cache-control: no-cache'
```

Response Example:

```
HTTP/1.1 204 No Content
Content-Type:application/json;charset=UTF-8
```

Please note that on successful delete No response will be received in the response body - but the HTTP Status Code will be set to 204 No Content.

Also, the message will only be deleted if the message was created by (i.e. sent from) the user whose access token is being sent as authorization.

LIST MESSAGES : List messages of user whose access token is sent | **GET** /messages

PARAMETERS (QUERY STRING)

This end-point also supports exact search using following fields as parameter

job_id	optional	Query parameter to filter messages by job_id

`user_id`	optional	Query parameter to filter messages by user_id of the other party (sender or receiver) in the message.

Authorization: REQUIRED

The API uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

1)
```
curl -X GET \
  --url
http://<domain>/api/web/v1/messages \
  -H 'authorization: Bearer
5bJzwIqQjucbcxH7Ri2PnzBj' \
  -H 'cache-control: no-cache'
```

2)
```
curl -X GET \
  --url
http://<domain>/api/web/v1/messages?job
_id=14&user_id=22 \
  -H 'authorization: Bearer
5bJzwIqQjucbcxH7Ri2PnzBj' \
  -H 'cache-control: no-cache'
```

Note: The messages listed will be those which were either sent by or received by the user whose access_token is being sent as authorization.

Response Example:

```
HTTP/1.1 200 OK
X-Pagination-Total-Count: 4
X-Pagination-Page-Count: 1
X-Pagination-Current-Page: 1
X-Pagination-Per-Page: 20
Link:
<http://<domain>/api/web/v1/messages?job_id=14&user_id=22&
page=1>; rel=self
Transfer-Encoding: chunked
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 54,
            "from_user_id": 22,
            "to_user_id": 1,
            "job_id": 14,
            "message": "Giving updates in 1 hr.",
            "created_at": 1442820487
        },
       ...
       ...
    ],
    "_links": {
        "self": {
            "href":
"http://<domain>/api/web/v1/messages?job_id=14&user_id=22&page=1"
        }
    },
    "_meta": {
        "totalCount": 4,
        "pageCount": 1,
        "currentPage": 1,
        "perPage": 20
    }
}
```

Provider Activity

Ulafix allows REST clients to get a list of all jobs related to the particular provider which stored in the system.

LIST : All jobs will be listed for particular provider	**GET** / company-jobs

Authorization: REQUIRED
The API uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:
*authorization: Bearer **<token>***

Request Example:

```
curl -X GET \
       --url
http://<domain>/api/web/v1/
company-jobs \
-H 'authorization: Bearer
X_J8kwflDQH7sTbRBX-
UxvAU5R0Jq4qC' \
       -H 'cache-control:
no-cache'
```

Response Example:

```
HTTP/1.1 200 OK
X-Pagination-Total-Count: 21
X-Pagination-Page-Count: 2
X-Pagination-Current-Page: 1
X-Pagination-Per-Page: 20
Link: <http://<domain>/api/web/v1/company-jobs?page=1>; rel=self,
Transfer-Encoding: chunked
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 90,
            "user_id": 22,
            "job_id": 37,
            "distance": "6.588179",
            "status": 3,
            "created_at": 1451976585,
            "status_text": "Awarded",
            "service_type": "Transmission Repair",
            "service_type_id": 12,
            "job_name": "fjob1",
            "job_description": "In by an appetite no humoured
      returned informed. Possession so comparison inquietude he
      he conviction no decisively. Marianne jointure attended
      she hastened surprise but she. Ever lady son yet you very
      paid form away. He advantage of exquisite resolving if on
      tolerably. Become sister on in garden it barton waited
      on. ",
            "job_datetime": "2016-01-23 13:30:00",
            "job_dt_choice_from": "2016-01-23 13:00:00",
            "job_dt_choice_to": "2016-01-23 22:00:00",
            "zipcodeid": 38905,
            "zipcode": "92111",
            "jobcode": 1451976585,
            "client_id": 1,
            "clientname": "GBIT Souvik",
            "clientemail": "souvik.gbit@gmail.com",
            "clientcontact": "8790549672",
            "bid_amount": "30.00",
            "bid_start": "2016-01-23 13:30:00",
            "bid_duration": "12.00",
            "bid_is_accepted": "1"
        },
        ...
                ...
        {
            "id": 9,
            "user_id": 22,
            "job_id": 14,
            "distance": "0.000000",
            "status": 3,
            "created_at": 1440149145,
            "status_text": "Awarded",
            "service_type": "Oil Change",
            "service_type_id": 7,
            "job_name": "Latest Test Job 2108",
```

```
            "job_description": "This will test if job alert
          ...
             ...

            "zipcode": "92025",
            "jobcode": 1440149145,
            "client_id": 1,
            "clientname": "GBIT Souvik",
            "clientemail": "souvik.gbit@gmail.com",
            "clientcontact": "8790549672",
            "bid_amount": "236.00",
            "bid_start": "2015-08-24 13:00:00",
            "bid_duration": "1.00",
            "bid_is_accepted": "1"
        }
    ],
    "_links": {
        self": {
            "href": "http://profiles.dev/api/web/v1/company-
    jobs?page=1"
        },
        "next": {
            "href": "http://profiles.dev/api/web/v1/company-
    jobs?page=2"
        },
        "last": {
            "href": "http://profiles.dev/api/web/v1/company-
    jobs?page=2"
        }
    },
    "_meta": {
        "totalCount": 21,
        "pageCount": 2,
        "currentPage": 1,
        "perPage": 20
    }
}
```


SEARCH : Search based on following query string	**GET** /company-jobs/search?

PARAMETERS

`jobname`		The query/search by `jobname` parameter used to get the list of jobs as per `jobname` search. It can also be used to search by job name as well as jobcode. `e.g:` /company-jobs/search?jobname= XWT-MAIN OR /company-jobs/search?jobname= 1452237455
`service`		The query/search by `service` parameter used to get the list of jobs as per `service` search. `e.g:` /company-jobs/search?service=9
`zipcode -OR- zipcodeid`		The query/search by `zipcode -OR- zipcodeid` parameter used to get the list of jobs as per `Zipcode -OR- zipcodeid` search. `e.g:` /company-jobs/search? zipcode=90101 OR /company-jobs/search?zipcodeid=38333

jobfromdate AND jobtodate		The query/search by jobfromdate -AND- jobtodate parameter used to get the list of jobs as per jobfromdate -AND- jobtodate search. It can be used to get the lists of job between jobfromdate to jobtodate date range. e.g: /company-jobs/search? jobfromdate=2016-03-01&jobtodate=2016-03-30
cjobstatus		The query/search by cjobstatus parameter used to get the list of jobs as per job status search. It can be used to get the lists of job as per job status by passing status. The following job status are used for Service Request: 0 => " Ignored " 1 => "Received" 2 => " Bid " 3 => " Awarded " 4 => "Finished" e.g: /company-jobs/search?cjobstatus=0

Request Example:

```
curl -X GET \
      --url
http://<domain>/api/web/v1/company-jobs/search? \ -H
'authorization: Bearer X_J8kwflDQH7sTbRBX-UxvAU5R0Jq4qC'
      -H 'cache-control: no-cache'
```

e.g :
http://<domain>/api/web/v1/company-jobs/search?jobname=1451976585&jobfromdate=2016-01-01&jobtodate=2016-03-30&expand=bid

Using the fields parameters, you may also specify which fields should be included in the result.
For example, the URL
http://<domain>/api/web/v1/company-jobs/search?q=au&fields=jobname,service_type will only return the jobname and searvice fields in the response. You can also use others option here.

Using the expand parameters, you may also specify which additional related fields should be added in the result. This end-point supports

Response Example:

```
HTTP/1.1 200 Created
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 90,
            "user_id": 22,
            "job_id": 37,
            "distance": "6.588179",
            "status": 3,
            "created_at": 1451976585,
            "status_text": "Awarded",
            "service_type": "Transmission Repair",
            "service_type_id": 12,
            "job_name": "fjob1",
            "job_description": "In by an appetite
      no humoured returned informed. Possession so
      comparison inquietude he he conviction no
      decisively. Marianne jointure attended she
      hastened surprise but she. Ever lady son yet
      you very paid form away. He advantage of
      exquisite resolving if on tolerably. Become
      sister on in garden it barton waited on. ",
            "job_datetime": "2016-01-23 13:30:00",
            "job_dt_choice_from": "2016-01-23
      13:00:00",
            "job_dt_choice_to": "2016-01-23
      22:00:00",
            "zipcodeid": 38905,
            "zipcode": "92111",
            "jobcode": 1451976585,
            "client_id": 1,
            "clientname": "GBIT Souvik",
            "clientemail": "souvik.gbit@gmail.com",
            "clientcontact": "8790549672",
            "bid_amount": "30.00",
```

following expand parameter only :

- **bid**: includes bid related info of job within the same structure response for particular provider.

Example usage:
`http://<domain>/api/web/v1/company-jobs?`**`expand`**`=bid`
will add `bid` details in the response structure.


```
                "bid_start": "2016-01-23 13:30:00",
                "bid_duration": "12.00",
                "bid_is_accepted": "1",
                "bid": [
                  {
                    "id": 60,
                    "parent_bid_id": 0,
                    "job_id": 37,
                    "job_created_by": 1,
                    "job_bid_by": 22,
                    "bid_amount": "30.00",
                    "suggested_job_start_datetime":
          "2016-01-23 13:30:00",
                    "duration": "12.00",
                    "comments": "",
                    "is_accepted": 1,
                    "created_at": 1451976817
                  }
                },

                ...
            ]
}
```


IGNORE: Ignore the job by using this action | **PUT** /company-obs/<company_job:id>

PARAMETERS

Company_job:id reference to the id field in response list.

`action`	**required**	ignore *(default value : ignore)*

Request Example:

```
curl -X PUT \
  --url http:
://<domain>/api/web/v1/company-jobs/90 \
  -H 'authorization: Bearer
X_J8kwflDQH7sTbRBX-UxvAU5R0Jq4qC' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/x-www-form-
urlencoded' \
  -d action=ignore

curl -X PATCH \
  --url http:
://<domain>/api/web/v1/company-jobs/90 \
  -H 'authorization: Bearer
X_J8kwflDQH7sTbRBX-UxvAU5R0Jq4qC' \
  -H 'cache-control: no-cache' \
  -H 'content-type: application/x-www-form-
urlencoded' \
  -d action=ignore
```

Response Example:

```
HTTP/1.1 200 OK
Content-Type:application/json;charset=UTF-8
{

  "isSuccess": true,
  "message": "This job is ignored by you."
}
```

Client Activity

Ulafix allows REST clients to get a list of all jobs.

LIST : All jobs will be listed	**GET** / client-jobs

Authorization: REQUIRED

The API uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X GET \
       --url
http://<domain>/api/web/v1/
client-jobs \
-H 'authorization: Bearer
5bJzwIqQjucbcxHFu-
7Ri2PnzBjimiue' \
       -H 'cache-control:
no-cache'
```

Response Example:

```
HTTP/1.1 200 OK
X-Pagination-Total-Count: 11
X-Pagination-Page-Count: 1
X-Pagination-Current-Page: 1
X-Pagination-Per-Page: 20
Link: <http://<domain>/api/web/v1/client-jobs?page=1>; rel=self,
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 40,
             "user_id": 1,
             "service_type_id": 7,
             "job_name": "MakeModelYear",
             "job_description": "testing oil change",
             "job_datetime": null,
             "job_dt_choice_from": "2016-02-24 08:00:00",
             "job_dt_choice_to": "2016-02-24 21:00:00",
             "zipcode_id": 38898,
             "search_radius": 100,
             "status": 1,
             "created_at": 1454406103,
             "updated_at": 1454406103,
             "status_text": "STATUS_CURRENT_JOBS",
             "service_type": "Oil Change",
             "car": {
               "car_model_year_id": 183,
               "car_model_year": "1986",
               "car_model_id": 33,
               "car_model": "5000CS Quattro",
               "car_make_id": 5,
               "car_make": "Audi"
             },
              "zipcode": "92104",
                    "jobcode": 1454406103,
                    "bid_win_by": "MediaInc",
                    "bid_win": {
                      "bid_amount": "2500.00",
                      "duration": "9.00"
                    },
             "bid_count": "1"       },
        ...
     }
    ],
    "_links": {
       self": {
       "href": "http://profiles.dev/api/web/v1/client-jobs?page=1"
           },
    },
    "_meta": {
        "totalCount": 11,
        "pageCount": 1,
        "currentPage": 1,
        "perPage": 20
    }
}
```

SEARCH : Search based on following query string		**GET** /client-jobs/search?
PARAMETERS		
`jobname`		The query/search by `jobname` parameter used to get the list of jobs as per `jobname` search. It can also be used to search by job name as well as jobcode. `e.g:` /client-jobs/search?jobname= fjob1 OR / client -jobs/search?jobname= 1451976585
`service`		The query/search by `service` parameter used to get the list of jobs as per `service` search. `e.g:` / client -jobs/search?service=9
`zipcode -OR- zipcodeid`		The query/search by `zipcode -OR- zipcodeid` parameter used to get the list of jobs as per `Zipcode -OR- zipcodeid` search. `e.g:` / client -jobs/search? zipcode=90101 OR / client -jobs/search?zipcodeid=38333
`jobfromdate AND jobtodate`		The query/search by `jobfromdate -AND- jobtodate` parameter used to get the list of jobs as per `jobfromdate -AND- jobtodate` search. It can be used to get the lists of job between `jobfromdate` to `jobtodate` date range. `e.g:` / client -jobs/search? jobfromdate=2016-03-01&jobtodate=2016-03-30
`jobstatus`		The query/search by `cjobstatus` parameter used to get the list of jobs as per `job status` search. It can be used to get the lists of job as per job status by passing status. The following job status are used for Service Request: `1` => "*Current*" `2` => " *Past* " `3` => " *Awarded* " `e.g:` / client -jobs/search?jobstatus=1 / client -jobs/search?jobstatus=2 / client -jobs/search?jobstatus=3

Request Example:

```
curl -X GET \
      --url
http://<domain>/api/web/v1/client-jobs/search? \ -H
'authorization: Bearer 5bJzwIqQjucbcxHFu-7Ri2PnzBjimiue'
      -H 'cache-control: no-cache'
```

`e.g :`
http://<domain>/api/web/v1/company-jobs/search?`jobname`=make&`jobfromdate`=2

Response Example:

```
HTTP/1.1 200 Created
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 40,
            "user_id": 1,
            "service_type_id": 7,
            "job_name": "MakeModelYear",
            "job_description": "testing oil
      change",
            "job_datetime": "2016-02-24 13:00:00",
            "job_dt_choice_from": "2016-02-24
      08:00:00",
```

016-01-01&jobtodate=2016-03-30&expand=bid

Using the fields parameters, you may also specify which fields should be included in the result.
For example, the URL http://<domain>/api/web/v1/ client-jobs /search?q=au&fields=jobname,service_type will only return the jobname and searvice fields in the response. You can also use others option here.

Using the expand parameters, you may also specify which additional related fields should be added in the result. This end-point supports following expand parameter only :

- **bid**: includes bid related info of job within the same structure response

Example usage:
http://<domain>/api/web/v1/client-jobs?**expand**=bid
will add bid details in the response structure.


```
            "job_dt_choice_to": "2016-02-24
        21:00:00",
            "zipcode_id": 38898,
            "search_radius": 100,
            "status": 3,
            "created_at": 1454406103,
            "updated_at": 1454406703,
            "status_text": "STATUS_AWARDED",
            "service_type": "Oil Change",
            "car": {
              "car_model_year_id": 183,
              "car_model_year": "1986",
              "car_model_id": 33,
              "car_model": "5000CS Quattro",
              "car_make_id": 5,
              "car_make": "Audi"
            },
            "zipcode": "92104",
            "jobcode": 1454406103,
            "bid_win_by": "MediaInc",
            "bid_win": {
              "bid_amount": "2500.00",
              "duration": "9.00"
            },
            "bid_count": "1",
            "bid": [
              {
                "id": 63,
                "parent_bid_id": 0,
                "job_id": 40,
                "job_created_by": 1,
                "job_bid_by": 53,
                "bid_amount": "2500.00",
                "suggested_job_start_datetime":
        "2016-02-24 13:00:00",
                "duration": "9.00",
                "comments": "test tester",
                "is_accepted": 1,
                "created_at": 1454406651
              }
            ]
          }

        },
        ...
    ]
}
```


DELETE: Delete the job by using this action

PUT /client-jobs/<job-id>

PARAMETERS
Job-id reference to the id field in response list.

Request Example:

```
curl -X DELETE \
--url http://<domain>//api/web/v1/client-jobs/40 \
  -H 'authorization: Bearer 5bJzwIqQjucbcxHFu-7Ri2PnzBjimiue' \
  -H 'cache-control: no-cache' \
```

Response Example:


```
HTTP/1.1 200 OK
Content-Type:application/json;charset=UTF-8
{

  "isSuccess": true,
  "message": "Your Job was deleted successfully."
}
```

Provider Analytics

Ulafix allows REST clients to get all the analytics of a particular provider.

LIST : Provider analytics will be listed here | **GET** / company-stats

Authorization: REQUIRED

The API uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X GET \
       --url http://<domain>/api/web/v1/
company-stats \
-H 'authorization: Bearer
X_J8kwflDQH7sTbRBX-UxvAU5R0Jq4qC' \
       -H 'cache-control: no-cache'
```

Using the `fields` parameters, you may also filter the provider analytics response by specify which analytics should be included in the result. Available fields parameters options are:

- `searchStats`,
- `bidConversionStats`,
- `jobCompletionStats`,
- `revanueStats`

`e.g:`

For example, the URL `http://<domain>/api/web/v1/`/ company -stats?**`fields`**=`searchStats,bidConversionStats, jobCompletionStats,revanueStats`

will only return as per filter by given fields parameters in the response.

Response Example:

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Content-Type: application/json; charset=UTF-8
{
  "searchStats": {
    "totalImpressionsCount": 27,
    "totalClickCount": 4
  },
  "bidConversionStats": {
    "totalBidJobs": "17",
    "totalSuccessJobs": "12",
    "totalFailedJobs": 5
  },
  "jobCompletionStats": {
    "totalBidJobs": "17",
    "totalCompletedJobs": "8",
    "totalPendingJobs": 4
  },
  "revanueStats": [
    {
      "job_id": "29",
      "amount": "375.00"
    },
    {
      "job_id": "28",
      "amount": "400.00"
    },
    {
      "job_id": "27",
      "amount": "450.00"
    },
  ...
     ...

    {
      "job_id": "TOTAL",
      "amount": "2600.00"
    }
  ]
}
```

Provider Calendar

Ulafix allows REST clients to get all the schedule events of a particular provider.

LIST : All the schedule events will be listed here | **GET** /company-job-schedules

Authorization: REQUIRED

The API uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X GET \
       --url http://<domain>/api/web/v1/ company-job-schedules \
-H 'authorization: Bearer  X_J8kwflDQH7sTbRBX-UxvAU5R0Jq4qC' \
       -H 'cache-control: no-cache'
```

Using the `fields` parameters, you may also specify which fields should be included in the result.
For example, the URL `http://<domain>/api/web/v1/company-job-schedules?fields=start,duration ...` will only return the `start` and `duration` fields in the response.

Using the `expand` parameters, you may also specify which additional related fields should be added in the result. This end-point supports following three expand parameters:

- **company**: includes company related info of a provider with same structure as **Users API** response
- **client**: includes client related info of client user to the related job with structure as **Users API** response
- **job**: includes job related info submitted by client user with same structure as **Client Job API** response

Example usage: `http://<domain>/api/web/v1/company-job-schedules?`**`expand`**`=company,client,job` will add `company , client` and `job` details in the response structure.

Response Example:

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Content-Type: application/json; charset=UTF-8
{

  "items": [
    {
      "id": 47,
      "job_id": 34,
      "client_id": 2,
      "company_id": 22,
      "start": "2016-01-15 13:30:00",
      "duration": "13.25",
      "duration_unit": "hours",
      "repeat_flag": 0,
      "repeat_type": null
    },
    {
      "id": 46,
      "job_id": 37,
      "client_id": 1,
      "company_id": 22,
      "start": "2016-01-23 13:30:00",
      "duration": "12.00",
      "duration_unit": "hours",
      "repeat_flag": 0,
      "repeat_type": null
    },
  ...
      ...

    {
      "id": 12,
      "job_id": 9,
      "client_id": 2,
      "company_id": 22,
      "start": "2015-08-14 10:30:00",
      "duration": "1.00",
      "duration_unit": "hours",
      "repeat_flag": 0,
      "repeat_type": null
    }
  ]
}
```

SEARCH : Search scheduled events based on start and end date query string | **GET** /company-job-schedules/search?<query parameters>

PARAMETERS

`start`	**required**	The query/search date parameter `e.g` (`2016-01-01`)
`end`	**required**	The query/search date parameter `e.g` (`2016-03-30`)

Request Example:

Response Example:

```
HTTP/1.1 200 Created
```

```
curl -X GET \
      --url
http://<domain>/api/web/v1/car-
models/search?start=2016-01-
01&end=2016-03-30 \
-H 'authorization: Bearer
X_J8kwflDQH7sTbRBX-UxvAU5R0Jq4qC' \
      -H 'cache-control: no-cache'
```

```
Content-Type: application/json; charset=UTF-8
[
  {
    "id": "47",
    "job_name": "Special test",
    "job_description": "Test job ",
    "service_type": "Oil Change",
    "zipcode": "92801",
    "search_radius": 100,
    "radius_unit": "mi",
    "start": "2016-01-15 13:30:00",
    "jobCreator": "New User",
    "winning_bid_amount": "20.00",
    "bid_duration": "13.25",
    "end": "2016-01-16 02:45:00"
  },
  {
    "id": "46",
    "job_name": "fjob1",
    "job_description": "In by an appetite no
humoured returned informed. ",
    "service_type": "Transmission Repair",
    "zipcode": "92111",
    "search_radius": 100,
    "radius_unit": "mi",
    "start": "2016-01-23 13:30:00",
    "jobCreator": "GBIT Souvik",
    "winning_bid_amount": "30.00",
    "bid_duration": "12.00",
    "end": "2016-01-24 01:30:00"
  }
]
```

CREATE SCHEDULE : A new schedule will be created for external job. | **POST** /company-job-schedules

PARAMETERS

At least one set of schedule start and end value must be passed. The value of <n> below starts from 0;

`start<n>`	**required**	The schedule start `e.g '2016-06-10 12:00'`
`end<n>`	**required**	The schedule end `e.g '2016-06-10 13:30'`
`events_count`	**required**	Total no. of schedules to be added
`company_id`	**required**	User id whose calendar will be updated

Additional Info:

Authorization: REQUIRED

This end-point uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format: *authorization: Bearer **<token>***

Request Example:

```
curl -X POST \
  --url
http://<domain>/api/web/v1/messages/company-job-
schedules \
  -H 'authorization: Bearer X_J8kwflDQH7sTb' \
  -H 'cache-control: no-cache' \
  -H 'content-type: multipart/form-data;
boundary=---011000010111000001101001' \
  -F 'start0='\''2016-06-10 12:00'\''' \
```

Response Example:

```
HTTP/1.1 200 Created
Content-Type:application/json;charset=UTF-8
{
  "success": true,
  "message": "Calendar updated
successfully!"
}
```

```
  -F 'end0='\''2016-06-10 13:30'\''' \
  -F 'start1='\''2016-06-10 11:00'\''' \
  -F 'end1='\''2016-06-10 11:30'\''' \
  -F events_count=2 \
  -F company_id=22
```

The `company_id` value must be the same as that of user whose access token is being sent as authorization.

DELETE SCHEDULE : An external job schedule will be deleted | **DEL** /company-job-schedules /{ id}

Additional Info:

Authorization: REQUIRED

This end-point uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X DELETE \
  --url http://<domain>/api/web/v1/company-job-schedules/85 \
  -H 'authorization: Bearer X8kwflDQH7sTb' \
  -H 'cache-control: no-cache'
```

Response Example:

```
HTTP/1.1 204 No Content
Content-Type:application/json;charset=UTF-8
```

Please note that on successful delete No response will be received in the response body - but the HTTP Status Code will be set to 204 No Content.

Also, the schedule will only be deleted if:

a) it is an **external** job schedule

b) the job schedule was created by the user whose access token is being sent as authorization.

BULK DELETE : External job schedule bulk deletion | **GET** / company-job-schedules /bulk-delete

PARAMETERS (QUERY STRING)

`schedule_id[]`	**required**	Id of schedule to be deleted
`events_count`	**required**	Total no. of schedules to be deleted
`company_id`	**required**	User id whose calendar will be updated

Authorization: REQUIRED

The API uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X POST \
  --url http://<domain>/api/web/v1/company-job-schedules/bulk-delete \
  -H 'authorization: Bearer X_J8kwflDQH7sTb' \
  -H 'cache-control: no-cache' \
  -H 'content-type: multipart/form-data; boundary=---011000010111000001101001' \
  -F 'schedule_id[]=102' \
  -F 'schedule_id[]=103'  \
  -F del_events_count=2 \
  -F company_id=22
```

Response Example:

```
HTTP/1.1 200 Created
Content-Type:application/json;charset=UTF-8
{
  "success": true,
  "message": "Calendar updated successfully!"
}
```

Provider Review & Rating & Closed Job

Ulafix allows REST clients to get list of all review and ratings particular log user.

LIST : All the review & rating will be listed here	**GET** /company-reviews

Authorization: REQUIRED

The API uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X GET \
       --url http://<domain>/api/web/v1/
company-reviews \
-H 'authorization: Bearer
X_J8kwflDQH7sTbRBX-UxvAU5R0Jq4qC' \
       -H 'cache-control: no-cache'
```

Response Example:

```
HTTP/1.1 200 OK
X-Pagination-Total-Count: 6
X-Pagination-Page-Count: 1
X-Pagination-Current-Page: 1
X-Pagination-Per-Page: 20
Link: <http://<domain>/api/web/v1/company-
reviews?page=1>; rel=self,
Transfer-Encoding: chunked
Content-Type: application/json; charset=UTF-8
{

  "items": [
    {
      "id": 20,
      "review_for": 22,
      "review_by": 1,
      "job_id": 27,
      "review": "The efficiency could have
been much better...need to improve on the
communications side",
      "rating": "3.50",
      "created_at": 1443613307
    },
    {
      "id": 19,
      "review_for": 22,
      "review_by": 1,
      "job_id": 28,
      "review": "Good job guys..keep it up!",
      "rating": "4.00",
      "created_at": 1443613261
    },
  ...
      ...

     {
      "id": 11,
      "review_for": 22,
      "review_by": 2,
      "job_id": 9,
      "review": "Average services - just about
managed to get the job done - that's all",
      "rating": "3.00",
      "created_at": 1442322366
    }
  ]
       "_links": {
           "self": {
             "href":
       "http://profiles.dev/api/web/v1/company
       -reviews?page=1"
           }
         },
         "_meta": {
```

<table>
<tr><td></td><td><pre>
            "totalCount": 6,
            "pageCount": 1,
            "currentPage": 1,
            "perPage": 20
        }
}
</pre></td></tr>
<tr><td>SEARCH : Search review & rating for a job based on jobid query string</td><td>GET / company-reviews/search?<query parameters></td></tr>
<tr><td colspan="2">PARAMETERS
<table>
<tr><td>jobid</td><td>required</td><td>The query/search jobid parameter</td></tr>
</table></td></tr>
<tr><td>Request Example:

<pre>
curl -X GET \
      --url
http://<domain>/api/web/v1/company-reviews/search?jobid=29
&expand=review_by,review_for,job \
-H 'authorization: Bearer
X_J8kwflDQH7sTbRBX-UxvAU5R0Jq4qC' \
      -H 'cache-control: no-cache'
</pre>

Using the <code>fields</code> parameters, you may also specify which fields should be included in the result.
For example, the URL <code>http://<domain>/api/web/v1/company-reviews?fields=review,rating ...</code> will only return the <code>review</code> and <code>rating</code> fields in the response.

Using the <code>expand</code> parameters, you may also specify which additional related fields should be added in the result. This end-point supports following three expand parameters:
• review_for: includes user related sufficient info within structure response.
• review_by: includes user related sufficient info within structure response
• job: includes job related info submitted by client user with same structure as Client Job API response

Example usage:</td><td>Response Example:
<pre>
HTTP/1.1 200 Created
Content-Type: application/json; charset=UTF-8
{
  "items": [
    {
      "id": 16,
      "review_for": {
        "id": 22,
        "username": "souvik.provider",
        "email": "souvik@greenbirdit.com",
        "role": 2,
        "status": 1,
        "policy_agreed": 1,
        "created_at": 1439278924,
        "updated_at": 1440056530
      },
      "review_by": {
        "id": 46,
        "username": "integrated.user",
        "email": "tester2.gbit@gmail.com",
        "role": 1,
        "status": 1,
        "policy_agreed": 1,
        "created_at": 1443176218,
        "updated_at": 1443176384
      },
      "job_id": 29,
      "review": "Excellent services, very
professional. Highly recommended!",
      "rating": "5.00",
      "created_at": 1443179948,
      "job": {
        "id": 29,
        "user_id": 46,
        "service_type_id": 12,
        "job_name": "Integrated Job 2509",
        "job_description": "Enter at your peril,
past the vaulted door. Impossible things will happen
that the world's never seen before. In Dexter's
laboratory lives the smartest boy you've ever seen,
but Dee Dee blows his experiments to Smithereens!
There's gloom and doom when things go boom in
Dexter's lab!",
        "job_datetime": "2015-09-30 14:30:00",
        "job_dt_choice_from": "2015-09-30 12:30:00",
        "job_dt_choice_to": "2015-09-30 16:30:00",
        "zipcode_id": 38896,
        "search_radius": null,
</pre></td></tr>
</table>

http://<domain>/api/web/v1/company-reviews?**expand**=review_by,review_for,job
will add `review_by` , `review_for` and `job` details in the response structure.


```
        "car_model_year_id": null,
        "status": 2,
        "created_at": 1443176616,
        "updated_at": 1443179909
      }
    }
  ],
  "_links": {
    "self": {
      "href":
"http://profiles.dev/api/web/v1/company-reviews/search?jobid=29&expand=review_by,review_for,job&page=1&per-page=10"
    }
  },
  "_meta": {
    "totalCount": 1,
    "pageCount": 1,
    "currentPage": 1,
    "perPage": 10
  }
}
```


CREATE REVIEW, RATING & CLOSED JOB : A review , rating and closed a job can be done by using Create Method.

POST /company-reviews

PARAMETERS

`review`	**required**	It is simple Text type .
`review_by`	**required**	id of the User who want to submit review & rating; *(available in logged user Id)*
`review_for`	**required**	id of the User for which `review_by` want to give review & rating; *(available in response structure of **User** API)*
`rating`	**optional**	Specify in numeric (1, 1.5, 2, 2.5 to 5)
`job_id`	**required**	id of Job under whose review will be created; *(available in response structure of **Client Activity** API)*
`extra_amount_paid`	**optional**	It is a number with two decimal place (`eg: 102.25`)

Additional Info:

Authorization: REQUIRED

This end-point uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X POST \
  --url http://<domain>/api/web/v1/company-reviews \
  -H 'authorization: Bearer 5bJzwIqQjucbcxHFu-7Ri2PnzBjimiue' \
  -H 'cache-control: no-cache' \
  -H 'content-type: multipart/form-data; boundary=---011000010111000001101001' \
  -F 'review =testing the review & rating' \
  -F review_by = 1 \
```

Response Example:

```
HTTP/1.1 201 Created
Content-Type:application/json;charset=UTF-8
Location:http://<domain>/api/web/v1/company-reviews
{
  "isSuccess": 1,
  "message": "Your review was submitted and the Job was closed successfully."
}
```

`-F review_for = 22 \`
`-F rating= 3.5 \`
`-F job_id= 23 \`
`-F extra_amount_paid= 105.50`
 	
Note: The parameter-value pairs must be sent to the API server as HTTP multipart/form-data.
 	
In the above request example syntax, this is achieved by setting `content-type` header (`-H`) to `multipart/form-data` and the parameters along with their values are sent as form data (`-F`).
Also, the `review_by` value must be the same as that of the user whose access token is being sent as authorization.	

DELETE: Delete the Review & rating by using this action	**DELETE** /company-reviews/<review-id>
PARAMETERS *review-id reference to the id field in response list.*	

Request Example:
`curl -X DELETE \`
`--url http://<domain>//api/web/v1/company-review/40 \`
`-H 'authorization: Bearer 5bJzwIqQjucbcxHFu-7Ri2PnzBjimiue' \`
`-H 'cache-control: no-cache' \`
	Response Example:
	`HTTP/1.1 200 OK`
	`Content-Type:application/json;charset=UTF-8`
	`{`
	`"isSuccess": 1,`
	`"message": "Your review was deleted successfully."`
	`}`

Premium Companies upgrade (for companies)

Ulafix allows REST clients to get upgrade their companies with premium services.

Right now we are using paypal payment gateway to get upgrade the companies account. Mainly there are two steps to complete this process.

When a company process to upgrade the account by selecting any one premium slot (which ever available that time) will redirect to paypal gateway and then after make a successful payment in paypal will redirect to the system.

So first `STEP I` will call on before paypal api call.

If the payment made successful through paypal then `STEP II` only.

If the payment is fail due to any reason then skip `STEP II` and call `STEP III` only.

`STEP I :`

CREATE PREMIUM MEMBERSHIP : Before to redirect to payment gateway	**POST** /premium-memberships/create-premium-membership

PARAMETERS

`premium_pricing_id`	**required**	Id of the premium pricing slot *(available in response structure of **Premium Prices** API)*
`zipcode_id`	**required**	id of the Zipcode for which `user(company)` want to upgrade with premium; *(available in response structure of **Zipcode** API)*

Authorization: REQUIRED

The API uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X POST \
       --url
http://<domain>/api/web/v1/premium-
memberships/create-premium-membership \
-H 'authorization: Bearer
X_J8kwflDQH7sTbRBX-UxvAU5R0Jq4qC' \
-H 'cache-control: no-cache' \
-H 'content-type: multipart/form-data;
boundary=---011000010111000001101001' \
-H 'postman-token: 5b8203bf-ef3f-0819-9290-
5c1ad77e257a' \
-F premium_pricing_id=116717  \
-F zipcode_id=38895
```

Response Example:

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Content-Type: application/json; charset=UTF-8
{
  "isSuccess": 1,
  "message": "Premium membership data was
submitted successfully.",
  "premium_membership_data":
       {
          "premium_pricing_id": "116717",
          "user_id": "53",
           "zipcode_id": "38895",
           "created_at": 1454582931,
           "updated_at": 1454582931,
           "id": 44,
           "zipcode": "92101"
         }
}
```

STEP II

UPDATE PREMIUM MEMBERSHIP : AFter redirect from payment gateway if payment successfully DONE	**POST** /premium-memberships/<premium_id>/update-premium-membership

PARAMETERS

paypal_token	**required**	Get it from paypal response.
paypal_payment_id	**required**	Get it from paypal response *(After successful payment id done)*
paypal_payer_id	**required**	Get it from paypal response *(After successful payment id done)*
paypal_status	**required**	Get it from paypal response *(It will be true on false. True : Successful payment done)*

Authorization: REQUIRED

The API uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X POST \
      --url
http://<domain>/api/web/v1/premium-
memberships/44/update-premium-membership \
-H 'authorization: Bearer
X_J8kwflDQH7sTbRBX-UxvAU5R0Jq4qC' \
-H 'cache-control: no-cache' \
-H 'content-type: multipart/form-data;
boundary=---011000010111000001101001' \
-F paypal_token=EC-4K83345815488845K \
-F paypal_payment_id=PAY-
8SH87143DW5551739KYGQC7Y \
-F paypal_payer_id=S9W267BBPR7RC \
-F paypal_status=true
```

Response Example:

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Content-Type: application/json; charset=UTF-8
{
  "isSuccess": 1,
  "message": "anas.provider account has been
upgraded to a premium provider account.",
  "premium_membership_data":
        {
          "id": 44,
          "premium_pricing_id": 116717,
          "zipcode_id": 38895,
          "user_id": 53,
          "paypal_token": "EC-
      4K83345815488845K",
          "paypal_payment_id": "PAY-
      8SH87143DW5551739KYGQC7Y",
          "paypal_payer_id": "S9W267BBPR7RC",
          "paypal_status": "true",
          "status": 1,
          "created_at": 1454582931,
          "updated_at": 1454583149,
          "zipcode": "92101"
        }
 }
```

STEP III

DELETE: this method will call when payment is fail	**DELETE** / premium-memberships /< premium -id>

PARAMETERS

Premium_id reference to the id field in response which get it from `STEP I.`

Request Example:

```
curl -X DELETE \
--url http://<domain>//api/web/v1/premium-
memberships/40 \
  -H 'authorization: Bearer
5bJzwIqQjucbcxHFu-7Ri2PnzBjimiue' \
  -H 'cache-control: no-cache' \
```

Response Example:

```
HTTP/1.1 200 OK
Content-Type:application/json;charset=UTF-8
{
  "isSuccess": 1,
  "message": "Your failure preimium entry was
deleted successfully."
}
```

Premium Prices

Ulafix allows REST clients to get a list of all premium price slots stored in the system.

LIST : All premium price slots will be listed	**GET** /premium-prices

Authorization: REQUIRED
The API uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:
*authorization: Bearer **<token>***

Request Example:

```
curl -X GET \
       --url
http://<domain>/api/web/v1/
premium-prices \
-H 'authorization: Bearer
5bJzwIqQjucbcxH7Ri2PnzBj' \
       -H 'cache-control:
no-cache'
```

Response Example:

```
HTTP/1.1 200 OK
X-Pagination-Total-Count: 128229
X-Pagination-Page-Count: 6412
X-Pagination-Current-Page: 1
X-Pagination-Per-Page: 20
Link: <http://<domain>/api/web/v1/premium-prices?page=1>; rel=self,
      <http://<domain>/api/web/v1/premium-prices?page=2>; rel=next,
      <http://<domain>/api/web/v1/premium-prices?page=2138>; rel=last
Transfer-Encoding: chunked
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 1,
            "zipcode_id": 38896,
            "pricing_name": "Slot 1 - 92102",
            "price": "100.00",
            "pricing_order": 1,
            "status": 1,
            "created_at": 1440485227,
            "updated_at": 1440485227,
            "zipcode": "92102"
        },
        ...
              ...
        {
            "id": 21,
            "zipcode_id": 39775,
            "pricing_name": "Test Slot 3 : 94045",
            "price": "50.00",
            "pricing_order": 3,
            "status": 1,
            "created_at": 1440485227,
            "updated_at": 1443078653,
            "zipcode": "94045"
        }
    ],
    "_links": {
        "self": {
   "href": "http://<domain>/api/web/v1/premium-prices?page=1"
        },
        "next": {
   "href": "http://<domain>/api/web/v1/premium-prices?page=2"
        },
        "last": {
   "href": "http://<domain>/api/web/v1/premium-prices?page=6412"
        }
    },
    "_meta": {
        "totalCount": 128229,
        "pageCount": 6412,
        "currentPage": 1,
        "perPage": 20
    }
}
```

SEARCH : Search premium price slots based on query string	**GET** /premium-prices/search?q=*nnn*

PARAMETERS

`q`	**required**	The query/search parameter to match with zipcode of the slot

Authorization: REQUIRED

The API uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X GET \
      --url
http://<domain>/api/web/v1/premium-
prices/search?q=921 \
      -H 'authorization: Bearer
      5bJzwIqQjucbcxH7Ri2PnzBj' \
      -H 'cache-control: no-cache'
```

Using the `fields` parameters, you may also specify which fields should be included in the result.

For example, the URL

`http://<domain>/api/web/v1/premium-prices/search?q=921&`**`fields`**`=id,zipcode`

will only return the `id` and `zipcode` fields in the response.

Response Example:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 37314,
            "zipcode_id": 12426,
            "pricing_name": "Slot 1 - 29210",
            "price": "100.00",
            "pricing_order": 1,
            "status": 1,
            "created_at": 1440485227,
            "updated_at": 1443078633,
            "zipcode": "29210"
        },
        ...
         ...
        {
            "id": 127259,
            "zipcode_id": 42413,
            "pricing_name": "Slot 2 - 99210",
            "price": "75.00",
            "pricing_order": 2,
            "status": 1,
            "created_at": 1440485227,
            "updated_at": 1443078644,
            "zipcode": "99210"
        }
    ]
}
```

Premium Companies

Ulafix allows REST clients to get a list of all providers who have upgraded their membership to premium status.

LIST : All premium companies will be listed	**GET** / premium-companies

Authorization: REQUIRED

The API uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X GET \
       --url
http://<domain>/api/web/v1/
premium-companies \
-H 'authorization: Bearer
5bJzwIqQjucbcxH7Ri2PnzBj' \
       -H 'cache-control:
no-cache'
```

Response Example:

```
HTTP/1.1 200 OK
X-Pagination-Total-Count: 6
X-Pagination-Page-Count: 1
X-Pagination-Current-Page: 1
X-Pagination-Per-Page: 20
Link: <http://<domain>/api/web/v1/premium-prices?page=1>; rel=self,
Transfer-Encoding: chunked
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 5,
            "premium_pricing_id": 1,
            "zipcode_id": 38896,
            "user_id": 22,
            "paypal_token": "EC-7048423543430774P",
            "paypal_payment_id": "PAY-42249494AK837384LKXUDTHQ",
            "paypal_payer_id": "S9W267BBPR7RC",
            "paypal_status": "true",
            "status": 1,
            "created_at": 1441282461,
            "updated_at": 1441282485,
            "zipcode": "92102"
        },
        ...
                ...
        {
            "id": 12,
            "premium_pricing_id": 13,
            "zipcode_id": 39267,
            "user_id": 47,
            "paypal_token": "EC-7DE88202BP1048346",
            "paypal_payment_id": "PAY-7KV55439JA317940FKYGPZDY",
            "paypal_payer_id": "S9W267BBPR7RC",
            "paypal_status": "true",
            "status": 1,
            "created_at": 1443691613,
            "updated_at": 1443691676,
            "zipcode": "92801"
        }
    ],
    "_links": {
        "self": {
   "href": "http://<domain>/api/web/v1/premium-companies?page=1"
        }
    },
    "_meta": {
        "totalCount": 6,
        "pageCount": 1,
        "currentPage": 1,
        "perPage": 20
    }
}
```

SEARCH : Search premium companies based on query string	**GET** /premium- companies/search?q=*nnn*

PARAMETERS

`q`	**required**	The query/search parameter to match with zipcode OR username OR email of the premium provider

Authorization: REQUIRED

This end-point uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X GET \
      --url
http://<domain>/api/web/v1/premium-companies/search?q=921 \
      -H 'authorization: Bearer 5bJzwIqQjucbcxH7Ri2PnzBj' \
      -H 'cache-control: no-cache'
```

Using the `fields` parameters, you may also specify which fields should be included in the result.

For example, the URL `http://<domain>/api/web/v1/premium-companies/search?q=921&`**`fields`**`=id,paypal_token` will only return the `id` and `paypal_token` fields in the response.

Using the `expand` parameters, you may also specify which additional related fields should be added in the result. This end-point supports following two expand parameters:

- **user**: includes user related info of premium provider with same structure as **User API** response
- **pricing**: includes premium pricing related info of premium slot subscribed to by provider with same structure as **Premium Pricing API** response

Example usage: `http://<domain>/api/web/v1/premium-companies/search?q=921&`**`expand`**`=user,pricing` will add `user` and `pricing` details in the response structure.

Response Example:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 11,
            "premium_pricing_id": 12,
            "zipcode_id": 39267,
            "user_id": 26,
            "paypal_token": "EC-8K295667HG627721M",
            "paypal_payment_id": "PAY-38R37756MD468405RKYGPWRQ",
            "paypal_payer_id": "S9W267BBPR7RC",
            "paypal_status": "true",
            "status": 1,
            "created_at": 1443691284,
            "updated_at": 1443691308,
            "zipcode": "92801"
        },
        ...
         ...
    ]
}
```

EXPAND Parameter Response Example:

```
{
    "items": [
        {
            "id": 11,
            "premium_pricing_id": 12,
            "zipcode_id": 39267,
            "user_id": 26,
            "paypal_token": "EC-8K295667HG627721M",
            ...
            "pricing": {
                "id": 12,
                "zipcode_id": 39267,
                "pricing_name": "Slot 2 - 92801",
                ...
            },
            "user": {
                "id": 26,
                "username": "anaheim.provider",
                ...
                "profile": {
                    ...
                },
                "contacts": {
                    ...
                }
            }
        },
       ...
       ...
    ]
}
```

Zipcodes

Ulafix allows REST clients to get a list of all Zipcodes (USA) stored in the system.

LIST : All zipcodes will be listed	**GET** /zipcodes

Request Example:

```
curl -X GET \
      --url
http://<domain>/api/web/
v1/zipcodes \
      -H 'cache-control:
no-cache'
```

Response Example:

```
HTTP/1.1 200 OK
X-Pagination-Total-Count: 42741
X-Pagination-Page-Count: 2138
X-Pagination-Current-Page: 1
X-Pagination-Per-Page: 20
Link: <http://<domain>/api/web/v1/zipcodes?page=1>; rel=self,
      <http://<domain>/api/web/v1/zipcodes?page=2>; rel=next,
      <http://<domain>/api/web/v1/zipcodes?page=2138>; rel=last
Transfer-Encoding: chunked
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 1,
            "country_id": 233,
            "zipcode": "00501",
            "latitude": "40.922326",
            "longitude": "-72.637078",
            "city": "Holtsville",
            "state": "NY",
            "status": 1
        },
        ....
         ....
        {
            "id": 20,
            "country_id": 233,
            "zipcode": "00631",
            "latitude": "18.269187",
            "longitude": "-66.864993",
            "city": "Castaner",
            "state": "PR",
            "status": 1
        }
    ],
    "_links": {
        "self": {
            "href": "http://<domain>/api/web/v1/zipcodes?page=1"
        },
        "next": {
            "href": "http://<domain>/api/web/v1/zipcodes?page=2"
        },
        "last": {
            "href":
"http://<domain>/api/web/v1/zipcodes?page=2138"
        }
    },
    "_meta": {
        "totalCount": 42741,
        "pageCount": 2138,
        "currentPage": 1,
        "perPage": 20
    }
}
```

SEARCH : Search zipcodes based on query string	**GET** /zipcodes/search?q=*nnn*

PARAMETERS

`q`	**required**	The query/search parameter to match with zipcode

Request Example:

```
curl -X GET \
      --url
http://<domain>/api/web/v1/zipcodes
/search?q=921 \
      -H 'cache-control: no-cache'
```

Using the `fields` parameters, you may also specify which fields should be included in the result.
For example, the URL `http://<domain>/api/web/v1/zipcodes/search?q=921&fields=id,zipcode` will only return the `id` and `zipcode` fields in the response.

Response Example:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 143,
            "country_id": 233,
            "zipcode": "00921",
            "latitude": "18.390429",
            "longitude": "-66.066124",
            "city": "San Juan",
            "state": "PR",
            "status": 1
        },
        {
            "id": 548,
            "country_id": 233,
            "zipcode": "01921",
            "latitude": "42.683256",
            "longitude": "-71.017403",
            "city": "Boxford",
            "state": "MA",
            "status": 1
        },
              ...
              ...
        {
            "id": 42733,
            "country_id": 233,
            "zipcode": "99921",
            "latitude": "55.513301",
            "longitude": "-133.117022",
            "city": "Craig",
            "state": "AK",
            "status": 1
        }
    ]
}
```

Countries

Ulafix allows REST clients to get a list of all Countries stored in the system.

LIST : All countries will be listed	**GET** /countries

Request Example:

```
curl -X GET \
      --url
http://<domain>/api/web/
v1/countries \
      -H 'cache-control:
no-cache'
```

Response Example:

```
HTTP/1.1 200 OK
X-Pagination-Total-Count: 250
X-Pagination-Page-Count: 13
X-Pagination-Current-Page: 1
X-Pagination-Per-Page: 20
Link: <http://<domain>/api/web/v1/countries?page=1>; rel=self,
      <http://<domain>/api/web/v1/countries?page=2>; rel=next,
      <http://<domain>/api/web/v1/countries?page=13>; rel=last
Transfer-Encoding: chunked
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 1,
            "country_code": "AD",
            "phone_code": 376,
            "name": "Andorra"
        },
        ...
         ...
        {
            "id": 20,
            "country_code": "BE",
            "phone_code": 32,
            "name": "Belgium"
        }
    ],
    "_links": {
        "self": {
            "href":
"http://<domain>/api/web/v1/countries?page=1"
        },
        "next": {
            "href":
"http://<domain>/api/web/v1/countries?page=2"
        },
        "last": {
            "href":
"http://<domain>/api/web/v1/countries?page=13"
        }
    },
    "_meta": {
        "totalCount": 250,
        "pageCount": 13,
        "currentPage": 1,
        "perPage": 20
    }
}
```

SEARCH : Search countries based on query string	**GET** /countries/search?q=*nnn*

PARAMETERS

q	**required**	The query/search parameter to match with country name

Request Example:

```
curl -X GET \
      --url
http://<domain>/api/web/v1/countrie
s/search?q=in \
      -H 'cache-control: no-cache'
```

Using the `fields` parameters, you may also specify which fields should be included in the result.
For example, the URL
`http://<domain>/api/web/v1/countrie s/search?q=in&fields=id,name`
will only return the `id` and `name` fields in the response.

Response Example:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 10,
            "country_code": "AR",
            "phone_code": 54,
            "name": "Argentina"
        },
        {
            "id": 17,
            "country_code": "BA",
            "phone_code": 387,
            "name": "Bosnia and Herzegovina"
        },
        ...
         ...
        {
            "id": 240,
            "country_code": "VI",
            "phone_code": 1340,
            "name": "U.S. Virgin Islands"
        }
    ]
}
```

Services

Ulafix allows REST clients to get a list of all Service Types stored in the system.

LIST : All service types will be listed	**GET** /services

Request Example:

```
curl -X GET \
      --url
http://<domain>/api/web/
v1/services \
      -H 'cache-control:
no-cache'
```

Response Example:

```
HTTP/1.1 200 OK
X-Pagination-Total-Count: 250
X-Pagination-Page-Count: 13
X-Pagination-Current-Page: 1
X-Pagination-Per-Page: 20
Link: <http://<domain>/api/web/v1/services?page=1>; rel=self
Transfer-Encoding: chunked
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 9,
            "service_description": "",
            "status": 1,
            "name": "Brake Repair"
        },
        {
            "id": 17,
            "service_description": "",
            "status": 1,
            "name": "Cooling Systems"
        },
```

```
            ...
                ...
            {
                "id": 8,
                "service_description": "",
                "status": 1,
                "name": "Wheel Alignment"
            }
        ],
        "_links": {
            "self": {
                "href": "http://<domain>/api/web/v1/services?page=1"
            }
        },
        "_meta": {
            "totalCount": 15,
            "pageCount": 1,
            "currentPage": 1,
            "perPage": 20
        }
}
```


SEARCH : Search services based on query string | **GET** /services/search?

*PARAMETERS (**at least one of the below parameters need to be passed in API call**)*

`q`	optional	The query/search parameter to match with service name `e.g.:` /services/search?q=*e*
`listby`	optional	The query/search by `listby` parameter used to get the list of services as per 2 service type flags - `emergency_services` and `auto_parts`. The following parameters can be passed: `FETCH_AUTOPARTSONLY` - only Parts (has `auto_parts` =1) `FETCH_REGULARONLY` - only Regular Services (has `emergency_services`=0) `FETCH_EMERGENCYONLY` - only Emergency Services (has `emergency_services`=1) `FETCH_ALL` - all active records in Services table `e.g:` /services/search?listby=FETCH_AUTOPARTSONLY /services/search?listby=FETCH_REGULARONLY /services/search?listby=FETCH_EMERGENCYONLY /services/search?listby=FETCH_ALL

Request Example:

```
curl -X GET \
      --url
http://<domain>/api/web/v1/services
/search?q=re \
      -H 'cache-control: no-cache'
```

Using the `fields` parameters, you may also specify which fields should be included in the result.
For example, the URL
`http://<domain>/api/web/v1/services/search?q=re&fields=id,name`
will only return the `id` and `name` fields in the response.

Response Example:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 21,
            "service_description": "",
            "emergency_service": 0,
            "auto_parts": 1,
            "status": 1,
            "name": "Parts"
        },
        ...
    ]
}
```

Car Make

Ulafix allows REST clients to get a list of all Car Makes stored in the system.

LIST : All car makes will be listed | **GET** /car-makes

Request Example:

```
curl -X GET \
      --url
http://<domain>/api/web/
v1/car-makes \
      -H 'cache-control:
no-cache'
```

Response Example:

```
HTTP/1.1 200 Created
X-Pagination-Total-Count: 79
X-Pagination-Page-Count: 4
X-Pagination-Current-Page: 1
X-Pagination-Per-Page: 20
Link: <http://<domain>/api/web/v1/car-makes?page=1>; rel=self,
      <http://<domain>/api/web/v1/car-makes?page=2>; rel=next,
      <http://<domain>/api/web/v1/car-makes?page=4>; rel=last
Transfer-Encoding: chunked
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 1,
            "status": 1,
            "name": "Acura"
        },
        {
            "id": 2,
            "status": 1,
            "name": "Alfa Romeo"
        },
        ...
            ...
        {
            "id": 20,
            "status": 1,
            "name": "Alfa Romeo"
        }
    ],
    "_links": {
            "self": {
                "href": "http://<domain>/api/web/v1/car-
      makes?page=1"
                },
            "next": {
                "href": "http://<domain>/api/web/v1/car-
      makes?page=2"
                },
            "last": {
                "href": "http://<domain>/api/web/v1/car-
      makes?page=4"
                }
        },
    "_meta": {
        "totalCount": 79,
        "pageCount": 4,
        "currentPage": 1,
        "perPage": 20
    }
}
```

SEARCH : Search makes based on query string | **GET** /car-makes/search?q=*au*

PARAMETERS

q	**required**	The query/search parameter

Request Example: | **Response Example:**

```
curl -X GET \
      --url
http://<domain>/api/web/v1/car-
makes/search?q=au \
      -H 'cache-control: no-cache'
```

Using the `fields` parameters, you may also specify which fields should be included in the result.
For example, the URL `http://<domain>/api/web/v1/car-makes/search?q=au&fields=id,name` will only return the `id` and `name` fields in the response.

```
HTTP/1.1 200 Created
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 5,
            "status": 1,
            "name": "Audi"
        },
        {
            "id": 6,
            "status": 1,
            "name": "Austin"
        },
        {
            "id": 64,
            "status": 1,
            "name": "Renault"
        }
    ]
}
```

Car Model

Ulafix allows REST clients to get a list of all Car Models stored in the system.

LIST : All car models will be listed | **GET** /car-models

Request Example:

```
curl -X GET \
      --url
http://<domain>/api/web/
v1/car-models \
      -H 'cache-control:
no-cache'
```

Response Example:

```
HTTP/1.1 200 Created
X-Pagination-Total-Count: 10469
X-Pagination-Page-Count: 53
X-Pagination-Current-Page: 1
X-Pagination-Per-Page: 20
Link: <http://<domain>/api/web/v1/car-models?page=1>; rel=self,
      <http://<domain>/api/web/v1/car-models?page=2>; rel=next,
      <http://<domain>/api/web/v1/car-models?page=53>; rel=last
Transfer-Encoding: chunked
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 108,
            "status": 1,
            "name": "1 Series",
            "make_id": 8,
            "make_name": "BMW"

        },
        {
            "id": 41,
            "status": 1,
            "name": "100",
            "make_id": 5,
            "make_name": "Audi"
        },
        ...
            ...
        {
            "id": 406,
            "status": 1,
```

```
                "name": "2500 Club Coupe",
                "make_id": 27,
                "make_name": "GMC"
            }
        ],
        "_links": {
                "self": {
                        "href": "http://<domain>/api/web/v1/car-
        models?page=1"
                        },
                "next": {
                        "href": "http://<domain>/api/web/v1/car-
        models?page=2"
                        },
                "last": {
                        "href": "http://<domain>/api/web/v1/car-
        models?page=53"
                        }
                },
        "_meta": {
            "totalCount": 1046,
            "pageCount": 53,
            "currentPage": 1,
            "perPage": 20
        }
}
```


SEARCH : Search models based on query string | **GET** /car-models/search?q=*au*

PARAMETERS

`q`	**required**	The query/search parameter

OR

`car_make_id`	**required**	Id of the "**car-make**" whose models needs to be fetched *(available in **id** field of response structure of **Car Make** API)*

Request Example:

```
curl -X GET \
      --url
http://<domain>/api/web/v1/car-
models/search?q=au \
      -H 'cache-control: no-cache'
```

OR

```
curl -X GET \
      --url
http://<domain>/api/web/v1/car-
models/search?car_make_id=8 \
      -H 'cache-control: no-cache'
```

Using the `fields` parameters, you may also specify which fields should be included in the result.
For example, the URL
`http://<domain>/api/web/v1/car-models/search?q=au&fields=id,name,make-name`
will only return the `id`,`name` and `make_name` fields in the response.

Response Example:

```
HTTP/1.1 200 Created
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 903,
            "status": 1,
            "name": "Aura"
            "make_id": 67,
            "make_name": "Saturn"
        },
        {
            "id": 812,
            "status": 1,
            "name": "Aurora"
            "make_id": 56,
            "make_name": "Oldsmobile"

        },
        {
            "id": 725,
            "status": 1,
            "name": "Marauder"
            "make_id": 49,
            "make_name": "Mercury"
        }
    ]
}
```

Car Year

Ulafix allows REST clients to get a list of all Car Years stored in the system.

LIST : All car years will be listed | **GET** /car-years

Request Example:

```
curl -X GET \
      --url
http://<domain>/api/web/
v1/car-years\
      -H 'cache-control:
no-cache'
```

Response Example:

```
HTTP/1.1 200 Created
X-Pagination-Total-Count: 7268
X-Pagination-Page-Count: 364
X-Pagination-Current-Page: 1
X-Pagination-Per-Page: 20
Link: <http://<domain>/api/web/v1/car-years?page=1>; rel=self,
      <http://<domain>/api/web/v1/car-years?page=2>; rel=next,
      <http://<domain>/api/web/v1/car-years?page=364>; rel=last
Transfer-Encoding: chunked
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 2538,
            "car_model_id": 329,
            "status": 1,
            "year": "1909",
        },
        {
            "id": 1576,
            "car_model_id": 230,
            "status": 1,
            "year": "1926",
        },
        ...
              ...
        {
            "id": 3905,
            "car_model_id": 575,
            "status": 1,
            "year": "2010",
        }
    ],
    "_links": {
           "self": {
        "href": "http://<domain>/api/web/v1/car-years?page=1"
            },
            "next": {
        "href": "http://<domain>/api/web/v1/car-years?page=2"
            },
            "last": {
        "href": "http://<domain>/api/web/v1/car-years?page=324"
            }
        },
    "_meta": {
        "totalCount": 7268,
        "pageCount": 364,
        "currentPage": 1,
        "perPage": 20
    }
}
```

SEARCH : Search years based on query string | **GET** /car-years/search?q=*1980*

PARAMETERS

q	**required**	The query/search parameter

OR

`car_model_id`	**required**	Id of the "**car model**" whose years needs to be fetched *(available in **id** field of response structure of **Car Model** API)*

Request Example:

```
curl -X GET \
      --url
http://<domain>/api/web/v1/car-
years/search?q=1980 \
      -H 'cache-control: no-cache'
```

OR

```
curl -X GET \
      --url
http://<domain>/api/web/v1/car-
years/search?car_model_id=108 \
      -H 'cache-control: no-cache'
```

Using the `fields` parameters, you may also specify which fields should be included in the result.
For example, the URL `http://<domain>/api/web/v1/car-years/search?q=au&fields=id,year` will only return the `id` and `year` fields in the response.

Response Example:

```
HTTP/1.1 200 Created
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 1051,
            "car_model_id": 158,
            "status": 1,
            "year": "1980",
        },
        {
            "id": 1051,
            "car_model_id": 165,
            "status": 1,
            "year": "1980",

        },
        {
            "id": 2551,
            "car_model_id": 332,
            "status": 1,
            "year": "1980",
        },
        ...
    ]
}
```

EXPAND : Expand years based on make and model query string — **GET** /car-years?expand=make,model

PARAMETERS

`expand`	**required**	The query/expand parameter

Request Example:

```
curl -X GET \
      --url
http://<domain>/api/web/v1/car-
years?expand=make,model \
      -H 'cache-control: no-cache'
```

Using the `expand` parameters, you may also expand results which give expanded result with make and model details.
For example, the URL `http://<domain>/api/web/v1/car-years?expand=make,model` will return the expandable results with `make` and `models` details in the response.

Response Example:

```
HTTP/1.1 200 Created
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 2538,
            "car_model_id": 329,
            "status": 1,
            "year": "1909",
            "model":
                    {
                        "id": 329,
                        "car_make_id": 25,
                        "model_name": "Model T",
                        "status":1,
                    }
            "make":
                    {
                        "id": 25,
                        "make_name": "Ford",
                        "status":1,
                    }
        },
        ...
    ]
}
```

Provider Closed Job List

Ulafix allows REST clients to get a list of all closed jobs to implement logic to forcefully close the jobs by provider.

CLOSE JOB : Get all listing of jobs as per provider, which need to be closed as per logic by the provider.	**GET** /company-jobs/job-closed

Authorization: REQUIRED

The API uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X GET \
      --url
http://<domain>/api/web/v1/company-
jobs/job-closed  \
      -H 'authorization: Bearer
      jnUK6JW9JUNB4eUp5YPvQkQh4jSwR9h5'
      \
      -H 'cache-control: no-cache'
```

Note: You can customize your logic by using the given list to force a provider to closed jobs. Here `client_job_status_flag` and `olderJobflag` are flag which using to set logic to mandatory close the job. if this flag are true.

Please refer the provider activity and Client activity API to understand the status.

Response Example:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
[
        {
            "job_id": 28,
            "job_name":"Regular Service-28Mar2016",
            "company_job_status": 3,
            "job_awarded_to": 24,
            "start": "2016-04-05 10:30:00",
            "end_datetime": "2016-04-05 11:00:00",
            "client_job_status": 3,
            "job_created_by":1,
            "client_job_status_flag": 0,
            "olderJobflag": 1,

        },
        ...
         ...
        {
            "job_id": 29,
            "job_name":"Towing-29Mar2016",
            "company_job_status": 3,
            "job_awarded_to": 24,
            "start": "2016-03-31 10:00:00",
            "end_datetime": "2016-03-31 10:30:00",
            "client_job_status": 2,
            "job_created_by":1,
            "client_job_status_flag": 1,
            "olderJobflag": 1,
        }
]
```

Auto Part Job (Request) Listing

Ulafix allows REST provider user to get a list of all auto parts (request) jobs.

LIST : All jobs will be listed	**GET** / auto-parts

Authorization: REQUIRED
The API uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:
*authorization: Bearer **<token>***

Request Example:

```
curl -X GET \
       --url
http://<domain>/api/web/v1/
auto-parts \
-H 'authorization: Bearer
p-
QOlgXcQw7dPb_pVxRl_KxQDVj81
kC-' \
       -H 'cache-control:
no-cache'
```

Response Example:

```
HTTP/1.1 200 OK
X-Pagination-Total-Count: 4
X-Pagination-Page-Count: 1
X-Pagination-Current-Page: 1
X-Pagination-Per-Page: 20
Link: <http://<domain>/api/web/v1/ auto-parts?page=1>; rel=self,
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 52,
             "user_id": 78,
             "service_type_id": 22,
             "job_name": "Tyres-27May2016",
             "job_description": "asdf",
             "job_datetime": null,
             "job_dt_choice_from": null,
             "job_dt_choice_to": null,
             "zipcode_id": null,
             "search_radius": null,
             "extra_amount_paid": null,
             "parts_image": "http://<domain>
      /frontend/assets/uploads/company/parts/78/sqr_09BUj7BKEk1
      5jwHPOc1NM1EK24YsRgD1.jpg",
             "job_type": 2,
             "status": 3,
             "created_at": 1464354420,
             "updated_at": 1464675730,
             "status_text": "STATUS_AWARDED",
              "service_type": "Tyres",
              "car": {
                      "car_model_year_id": 280,
                      "car_model_year": "2015",
                      "car_model_id": 28,
                      "car_model": "Amaze",
                      "car_make_id": 2,
                      "car_make": "Honda"
                    },
                    "zipcode": null,
                    "jobcode": 1464354420,
                    "bid_win_by": "Excel Auto Provider",
                    "bid_win": {
                      "bid_amount": "852.00",
                      "duration": "1.00",
                      "pay_method": 1
                    },
                    "bid_count": "1"
            },
        ...
    ],
    "_links": {
       self": {
       "href": "http://ulafix-hyd.dev/api/web/v1/auto-parts?page=1"
           },
    },
    "_meta": {
```

<table>
<tr><td></td><td><pre>
        "totalCount": 4,
        "pageCount": 1,
        "currentPage": 1,
        "perPage": 20
    }
}
</pre></td></tr>
<tr><td colspan="2">SEARCH : Search based on following query string</td><td>GET /auto-parts/search?</td></tr>
<tr><td colspan="3">PARAMETERS
<table>
<tr><td>jobname</td><td></td><td>The query/search by jobname parameter used to get the list of jobs as per jobname search. It can also be used to search by job name as well as jobcode.
e.g:
/auto-parts/search?jobname= clutch
OR
/ auto-parts/search?jobname= 1463638454</td></tr>
<tr><td>service</td><td></td><td>The query/search by service parameter used to get the list of jobs as per service search.
e.g: / auto-parts/search?service=9</td></tr>
<tr><td>jobstatus</td><td></td><td>The query/search by jobstatus parameter used to get the list of jobs as per job status search. It can be used to get the lists of job as per job status by passing status.

The following job status are used for Service Request:
1 => "Current"
2 => " Past "
3 => " Awarded "

e.g: / auto-parts/search?jobstatus=1
/ auto-parts/search?jobstatus=2
/ auto-parts/search?jobstatus=3</td></tr>
</table>
</td></tr>
<tr><td colspan="2">Request Example:

curl -X GET \
--url
http://<domain>/api/web/v1/auto-parts/search? \ -H
'authorization: Bearer p-QOlgXcQw7dPb_pVxRl_KxQDVj8lkC-'
-H 'cache-control: no-cache'

e.g :
http://<domain>/api/web/v1/auto-parts/search?jobname=clutch&jobstatus=1&expand=bid

Using the fields parameters, you may also specify which fields should be included in the result.
For example, the URL
http://<domain>/api/web/v1/auto-</td><td>Response Example:
HTTP/1.1 200 Created
Content-Type: application/json; charset=UTF-8
<pre>
{
    "items": [
        {
            "id": 46,
            "user_id": 78,
            "service_type_id": 21,
            "job_name": "Clutch Plate-19May2016",
            "job_description": "",
            "job_datetime": "2016-05-30 17:00:00",
            "job_dt_choice_from": "2016-05-30 08:00:00",
            "job_dt_choice_to": "2016-05-30 21:00:00",
            "zipcode_id": 558,
            "search_radius": null,
            "extra_amount_paid": "300.00",
            "parts_image": "http://<domain>/frontend/assets/uploads/company/parts/defaultLogo.png",
            "job_type": 2,
            "status": 2,
</pre></td></tr>
</table>

`parts /search?jobname=clutch &fields=job_name,service_type` will only return the `job_name` and `parts_image` fields in the response. You can also use others option here.

Using the `expand` parameters, you may also specify which additional related fields should be added in the result. This end-point supports following expand parameter only :

- **bid**: includes bid related info of job within the same structure response

Example usage:
`http://<domain>/api/web/v1/auto-parts?`**`expand`**`=bid`
will add `bid` details in the response structure.


```
            "created_at": 1463638454,
            "updated_at": 1464590200,
            "status_text": "STATUS_PAST_JOBS",
            "service_type": "Clutch Plate",
            "car": {
              "car_model_year_id": 899,
              "car_model_year": "2015",
              "car_model_id": 112,
              "car_model": "Indica Vista",
              "car_make_id": 8,
              "car_make": "Tata"
            },
            "zipcode": null,
            "jobcode": 1463638454,
            "bid_win_by": "Excel Auto Provider",
            "bid_win": {
              "bid_amount": "860.00",
              "duration": "1.00",
              "pay_method": 1
            },
            "bid_count": "4",
            "bid": [
              {
                "id": 52,
                "parent_bid_id": 0,
                "job_id": 46,
                "job_created_by": 78,
                "job_bid_by": 89,
                "bid_amount": "350.00",
                "suggested_job_start_datetime":
      "2016-05-30 12:00:00",
                "duration": "5.00",
                "comments": "",
                "is_accepted": 0,
                "pay_method": null,
                "created_at": 1463638596,
                "updated_at": 0
                },
              ...
               ]
              }
          },
          ...
    ]
}
```

Provider Parts Requests

Ulafix allows REST clients to create a service request from a User.

CREATE PARTS REQUEST : A new parts request (job) will be created.	**POST** /auto-parts

PARAMETERS

`user_id`	**required**	id of User under whose account service request will be created; *(available in response structure of **Login** API on successful login)*
`job_name`	**required**	Job name for easy identification
`form_service_type`	**required**	DB id of service_type table for services having auto_parts = 1 *(available in LIST response structure of **Services** API)*
`job_description`	**required**	Job description
`auto_parts_image`	optional	Provider logo file
`form_make_id`	**required**	DB id of car_make table for car makes *(available in LIST response structure of **Car Makes** API)*
`form_model_id`	**required**	DB id of car_model table for car models *(available in LIST response structure of **Car Models** API)*
`car_model_year_id`	**required**	DB id of car_model_year table for car years *(available in LIST response structure of **Car Years** API)*

Additional Info:

Authorization: REQUIRED

This end-point uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X POST \
  --url http://<domain>/api/web/v1/auto-parts \
  -H 'authorization: Bearer 5bJzwIqQjucbcxHFu' \
  -H 'cache-control: no-cache' \
  -H 'content-type: multipart/form-data; boundary=---011000010111000001101001' \
  -F user_id=22 \
  -F job_name=Parts-02June2016 \
  -F form_service_type=21 \
  -F 'job_description=API Parts Request..' \
  -F auto_parts_image=@abcd.png \
  -F form_make_id=1 \
  -F form_model_id=18 \
  -F car_model_year_id=188
```

Note: The parameter-value pairs must be sent to the API server as HTTP multipart/form-data.

In the above request example syntax, this is achieved by setting `content-type` header (`-H`) to `multipart/form-data` and the parameters along with their values are sent as form data (`-F`).

Response Example:

```
HTTP/1.1 201 Created
Content-Type:application/json;charset=UTF-8
Location:http://<domain>/api/web/v1/auto-parts/61
```


```
{
    "id": 61,
    "user_id": 22,
    "service_type_id": 21,
    "job_name": "Parts-02June2016",
    "job_description": " API Parts Request..",
    "job_datetime": null,
    "job_dt_choice_from": null,
    "job_dt_choice_to": null,
    "zipcode_id": null,
    "search_radius": null,
    "parts_image":
"http://<domain>/frontend/assets/uploads/company/parts/78/sqr_ghgh665a.jpg",
    "status": 1,
    "created_at": 1453888861,
    "updated_at": 1453888861,
    "status_text": "STATUS_CURRENT_JOBS",
    "service_type": "Parts",
    "car": {
        "car_model_year_id": 188,
        "car_model_year": "2014",
        "car_model_id": 18,
        "car_model": "Swift Dzire",
        "car_make_id": 1,
        "car_make": "Maruti Suzuki"
    },
    "zipcode": null,
    "jobcode": 1453888861,
    "bid_win_by": null,
    "bid_win": null,
    "bid_count": "0"
}
```

Also, the user_id value must be the same as that of the user whose access token is being sent as authorization.

RESEND PARTS REQUEST : A new parts request (job) will be created while deleting the existing job whose id is passed in the URL.

POST /auto-parts /*{job-id}*

PARAMETERS

`user_id`	**required**	id of User under whose account service request will be created; *(available in response structure of **Login** API on successful login)*
`job_name`	**required**	Job name for easy identification
`form_service_type`	**required**	DB id of service_type table for services having auto_parts = 1 *(available in LIST response structure of **Services** API)*
`job_description`	**required**	Job description
`auto_parts_image`	optional	Provider logo file
`form_make_id`	**required**	DB id of car_make table for car makes *(available in LIST response structure of **Car Makes** API)*
`form_model_id`	**required**	DB id of car_model table for car models *(available in LIST response structure of **Car Models** API)*
`car_model_year_id`	**required**	DB id of car_model_year table for car years *(available in LIST response structure of **Car Years** API)*

Additional Info:

Authorization: REQUIRED

This end-point uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X POST \
  --url http://<domain>/api/web/v1/auto-parts/61 \
  -H 'authorization: Bearer 5bJzwIqQjucbcxHFu' \
  -H 'cache-control: no-cache' \
  -H 'content-type: multipart/form-data; boundary=---011000010111000001101001' \
  -F user_id=22 \
  -F job_name=Parts-02June2016 \
  -F form_service_type=21 \
  -F 'job_description=API Parts Request 2' \
  -F auto_parts_image=@abcr.png \
  -F form_make_id=1 \
  -F form_model_id=18 \
  -F car_model_year_id=188
```

Note: The existing job, whose job-id is passed in the URL, will get deleted and a new job will be created.

Response Example:

```
HTTP/1.1 201 Created
Content-Type:application/json;charset=UTF-8
Location:http://<domain>/api/web/v1/auto-parts/62
{
    "id": 62,
    "user_id": 22,
    "service_type_id": 21,
    "job_name": "Parts-02June2016",
    "job_description": " API Parts Request 2",
    "job_datetime": null,
    "job_dt_choice_from": null,
    "job_dt_choice_to": null,
    "zipcode_id": null,
    "search_radius": null,
    "parts_image":
"http://<domain>/frontend/assets/uploads/company/parts/78/sqr_y65iik1a.jpg",
    "status": 1,
    "created_at": 1453888861,
    "updated_at": 1453888861,
    "status_text": "STATUS_CURRENT_JOBS",
    "service_type": "Parts",
    "car": {
        "car_model_year_id": 188,
        "car_model_year": "2014",
        "car_model_id": 18,
        "car_model": "Swift Dzire",
        "car_make_id": 1,
        "car_make": "Maruti Suzuki"
    },
    "zipcode": null,
    "jobcode": 1453888861,
    "bid_win_by": null,
    "bid_win": null,
    "bid_count": "0"
}
```

DELETE PARTS REQUEST : Parts request (job) will be deleted.	**DELETE** /auto-parts/*{job-id}*

Additional Info:
Authorization: REQUIRED
This end-point uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:
*authorization: Bearer **<token>***

Request Example:

```
curl -X DELETE \
  --url http://<domain>/api/web/v1/auto-parts/62 \
  -H 'authorization: Bearer 5bJzwIqQjucbcxHFu' \
  -H 'cache-control: no-cache'
```

Response Example:

```
HTTP/1.1 204 No Content
Content-Type:application/json;charset=UTF-8
```

Please note that on successful delete No response will be received in the response body - but the HTTP Status Code will be set to 204 No Content.

Also, the parts request will only be deleted if the message was created by (i.e. sent from) the user whose access token is being sent as authorization.

Auto Parts Bidding & Acceptance

Ulafix allows REST clients to create bids and manage the entire life-cycle of a bid on a parts request.

CREATE PARTS BID : A new bid will be created for a particular parts job.	**POST** /part-bids

PARAMETERS

`job_id`	**required**	id of Job for which this bid will be created; *(available in LIST response structure of **Provider Activity** API)*
`job_created_by`	**required**	id of User who created Job for which this bid will be created *(available in LIST response of **Provider Activity** API as `client_id` field)*
`job_bid_by`	**required**	id of Provider who is placing the bid; *(available in response of **Login** API after login as `id` field)*
`bid_amount`	**required**	Bid amount in decimal (e.g. 25, 20.00, 10.99 etc)
`comments`	optional	Additional remarks if any

Additional Info:
Authorization: REQUIRED
This end-point uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:
*authorization: Bearer **<token>***

Request Example:

```
curl -X POST \
 --url http://<domain>/api/web/v1/part-bids \
 -H 'authorization: Bearer X_J8kwflDQH7sTb' \
 -H 'cache-control: no-cache' \
 -H 'content-type: multipart/form-data; boundary=---011000010111000001101001' \
  -F job_id=51 \
  -F job_created_by=22 \
```

Response Example:

```
HTTP/1.1 201 Created
Content-Type:application/json;charset=UTF-8
Location:http://<domain>/api/web/v1/part-bids/53
{
    "job_id": "51",
    "job_created_by": "22",
    "job_bid_by": "85",
    "bid_amount": "2700.00",
    "comments": "Parts Bid from API",
    "is_accepted": false,
    "parent_bid_id": 0,
    "created_at": 1454323475,
    "id": 53,
    "is_accepted_numeric": 0
}
```

```
  -F job_bid_by=85 \
  -F bid_amount=2700.00 \
  -F 'comments=Parts Bid from API'
```

Note: The parameter-value pairs must be sent to the API server as HTTP multipart/form-data.

In the above request example syntax, this is achieved by setting `content-type` header (`-H`) to `multipart/form-data` and the parameters along with their values are sent as form data (`-F`).
Also, the job_bid_by value must be the same as that of user whose access token is being sent as authorization.

REVISE BID : A new bid will be created with a reference to previous bid. | **POST** /part-bids

PARAMETERS

`job_id`	**required**	id of Job for which this bid will be created; *(available in LIST response structure of **Provider Activity** API)*
`job_created_by`	**required**	id of User who created Job for which this bid will be created *(available in LIST response of **Provider Activity** API as `client_id` field)*
`job_bid_by`	**required**	id of Provider who is placing the bid; *(available in response of **Login** API after login as `id` field)*
`bid_amount`	**required**	Bid amount in decimal (e.g. 25, 20.00, 10.99 etc)
`comments`	optional	Additional remarks if any
`parent_bid_id`	**required**	DB id of job_bids table for latest bid by the provider on this job *(available in DETAIL response structure of **Client Jobs** API)*

Additional Info:
Authorization: REQUIRED
This end-point uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:
*authorization: Bearer **<token>***

Request Example:

```
curl -X POST \
--url http://<domain>/api/web/v1/part-bids \
-H 'authorization: Bearer X_J8kwflDQH7sTb' \
  -H 'cache-control: no-cache' \
  -H 'content-type: multipart/form-data;
boundary=---011000010111000001101001' \
  -F job_id=51 \
  -F job_created_by=22 \
  -F job_bid_by=85 \
  -F bid_amount=2400.00 \
  -F 'comments=Parts Bid from API - Revised'
  -F parent_bid_id=53
```

Note: The parameter-value pairs must be sent to the API server as HTTP multipart/form-data.

In the above request example syntax, this is achieved

Response Example:

```
HTTP/1.1 201 Created
Content-Type:application/json;charset=UTF-8
Location:http://<domain>/api/web/v1/part-
bids/54
{
    "job_id": "51",
    "job_created_by": "22",
    "job_bid_by": "85",
    "bid_amount": "2700.00",
    "comments": "Parts Bid from API - Revised",
    "is_accepted": false,
    "parent_bid_id": 53,
    "created_at": 1454323475,
    "id": 54,
    "is_accepted_numeric": 0
}
```

by setting `content-type` header (`-H`) to `multipart/form-data` and the parameters along with their values are sent as form data (`-F`).
Also, the job_bid_by value must be the same as that of user whose access token is being sent as authorization.

LIST BIDS : All bids placed by particular provider will be listed. | **GET** /part-bids

Authorization: REQUIRED
The API uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:
*authorization: Bearer **<token>***

Request Example:

```
curl -X GET \
  --url
http://<domain>/api/web/v1/
part-bids \
  -H 'authorization: Bearer
5bJzwIqQjucbcxH7Ri2PnzBj' \
  -H 'cache-control: no-
cache'
```

Note: If access_token sent as authorization is for

- Mechanics - then bids listed will be those that were placed on jobs this Provider created
- Parts Provider - then bids listed will be those that were placed by this Provider

Response Example:

```
HTTP/1.1 200 OK
X-Pagination-Total-Count: 48
X-Pagination-Page-Count: 3
X-Pagination-Current-Page: 1
X-Pagination-Per-Page: 20
Link: <http://<domain>/api/web/v1/part-bids?page=1>; rel=self,
      <http://<domain>/api/web/v1/part-bids?page=2>; rel=next,
      <http://<domain>/api/web/v1/part-bids?page=3>; rel=last
Transfer-Encoding: chunked
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 1,
            "parent_bid_id": 0,
            "job_id": 14,
            "job_created_by": 1,
            "job_bid_by": 22,
            "bid_amount": "350.00",
            "suggested_job_start_datetime": null,
            "duration": null,
            "comments": "First bid on Latest Test Job 2108",
            "is_accepted": false,
            "created_at": 1441104825,
            "is_accepted_numeric": 0
        },
        ...
        ...
    ],
    "_links": {
        "self": {
            "href": "http://<domain>/api/web/v1/part-bids?page=1"
        },
        "next": {
            "href": "http://<domain>/api/web/v1/part-bids?page=2"
        },
        "last": {
            "href": "http://<domain>/api/web/v1/part-bids?page=3"
        }
    },
    "_meta": {
        "totalCount": 48,
        "pageCount": 3,
        "currentPage": 1,
        "perPage": 20
    }
}
```

PICK BID : A bid is picked by Provider (mechanic) and Job is awarded to corresponding Parts Provider who placed that bid | **PUT** /part-bids/{bid-id}

PARAMETERS

`pay_method`	**required**	Payment Method - either `cash` or `online`

Additional Info:

Authorization: REQUIRED

This end-point uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X PUT \
  --url http://<domain>/api/web/v1/part-bids/54 \
  -H 'authorization: Bearer X_J8kwflDQH7sTb' \
  -H 'cache-control: no-cache'
  -H 'content-type: application/x-www-form-urlencoded' \
  -d pay_method=cash
```

Note: the bid can only be picked if the job, against which this bid was placed, had been created by the user whose access token is being sent as authorization.

Note: For online payment, the response will contain the bid object and an RSA key which will be used to encrypt relevant data in the mobile app, before sending over to ccAvenue server. For more details refer to ccAvenue Documentation.

Response Example:

```
HTTP/1.1 200 OK
Content-Type:application/json;charset=UTF-8
{
  "bid": {
    "id": 53,
    "parent_bid_id": 0,
    "job_id": 51,
    "job_created_by": 22,
    "job_bid_by": 85,
    "bid_amount": "2700.00",
  "suggested_job_start_datetime": null,
    "duration": null,
    "comments": "New Parts Bid API for axle housing",
    "is_accepted": true,
    "pay_method": 1,
    "created_at": 1465369896,
    "updated_at": 1465378097,
    "is_accepted_numeric": 1
  },
  "payment": {
    "user_id": 22,
    "provider_id": 85,
    "bid_id": 53,
    "job_id": 51,
    "order_status": "Success",
    "payment_mode": "Cash",
    "currency": "INR",
    "amount": 3105,
    "tax_amount": 405,
    "created_at": 1465378097,
    "updated_at": 1465378097,
    "id": 26
  }
}
```

SEARCH BIDS : Search & list parts bids based on query string

GET /part-bids/search

PARAMETERS (QUERY STRING)

This end-point supports LIKE search or partial-match search using following field as parameter

`q`	**required**	query/search parameter to match with Job code

This end-point also supports exact search using following fields as parameter

`job_id`	optional	query/search parameter to match with Job Id
`is_accepted`	optional	query/search parameter to match with is_accepted flag of bid

Authorization: REQUIRED

The API uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

1)

```
curl -X GET \
  --url
```

Response Example:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
{
    "items": [
        {
            "id": 18,
            "parent_bid_id": 12,
            "job_id": 14,
            "job_created_by": 1,
            "job_bid_by": 22,
            "bid_amount": "236.00",
            "suggested_job_start_datetime": "2015-08-24 13:00:00",
```

```
http://<domain>/api/web/v1/bids/search?q=1440149145 \
  -H 'authorization: Bearer 5bJzwIqQjucbcxH7Ri2PnzBj' \
  -H 'cache-control: no-cache'
```

2)
```
curl -X GET \
  --url http://<domain>/api/web/v1/bids/search?q=1440149145&is_accepted=1 \
  -H 'authorization: Bearer 5bJzwIqQjucbcxH7Ri2PnzBj' \
  -H 'cache-control: no-cache'
```

Note: If access_token sent as authorization is for

- Mechanics - then bids listed will be those that were placed on jobs this User created
- Parts Provider - then bids listed will be those that were placed by this Provider

Using the `expand` parameters, you may also specify which additional related fields should be added in the result. This end-point supports following expand parameters:

- **bid_by_company**: includes user related info of Provider who placed bid; the structure is same as **User API** response
- **created_by_user**: includes user related info of User who created the job; the structure is same as **User API** response
- **job**: includes job related info on which the bid is placed; the structure is same as **Job API** response

Example usage:
`http://<domain>/api/web/v1/premium-companies/search?q=1440149145&is_accepted=1&`**`expand`**`=job`
will add `job` details in the response structure.


```
            "duration": "1.00",
            "comments": "",
            "is_accepted": true,
            "created_at": 1441349174,
            "is_accepted_numeric": 1
        }
    ]
}
```


EXPAND Parameter Response Example:


```
{
    "items": [
        {
            "id": 18,
            "parent_bid_id": 12,
            "job_id": 14,
            "job_created_by": 1,
            "job_bid_by": 22,
            "bid_amount": "236.00",
             ...
            "job": {
                "id": 14,
                "user_id": 1,
                "service_type_id": 7,
                "job_name": "Latest Test Job 2108",
                "job_description": "Test job",
                ...
                "service_type": "Oil Change",
                "car": {
                    "car_model_year_id": 1078,
                    "car_model_year": "1995",
                    ...
                },
                "zipcode": "92025",
                "jobcode": 1440149145,
                "bid_win_by": "Greenbird Media Inc.",
                "bid_win": {
                    "bid_amount": "236.00",
                    "duration": "1.00"
                },
                "bid_count": "8"
            }
        }
    ]
}
```

Auto Part Closed Job List

Ulafix allows REST user providers to get a list of all closed jobs to implement logic to forcefully close the jobs by auto provider.

CLOSE JOB : Get all listing of auto part jobs as per auto provider, which need to be closed as per logic by the auto provider.	**GET** / `auto-parts/closed-job-list`

Authorization: REQUIRED

The API uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP header in following format:

*authorization: Bearer **<token>***

Request Example:

```
curl -X GET \
      --url
http://<domain>/api/web/v1/ auto-
parts/closed-job-list  \
      -H 'authorization: Bearer
      jnUK6JW9JUNB4eUp5YPvQkQh4jSwR9h5'
      \
      -H 'cache-control: no-cache'
```

Note: You can customize your logic by using the given list to force a auto provider to closed jobs. Here `client_job_status_flag` and `olderJobflag` are flag which using to set logic to mandatory close the job. if this flag are true.

Please refer the provider activity and Client activity API to understand the status.

Response Example:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
[
        {
                "job_name": "Tyres-27May2016",
                "job_id": "52",
                "status": "3",
                "user_id": "89",
                "assign_datetime": "2016-06-01
            11:52:10",
                "client_job_status": "0",
                "olderJobflag": "1"
        },
        ...
         ...
        {
                "job_name": "Clutch Plate-
27May2016",
                "job_id": "55",
                "status": "3",
                "user_id": "89",
                "assign_datetime": "2016-06-04
            15:20:24",
                "client_job_status": "1",
                "olderJobflag": "0"
        }
]
```

Auto Part Closed Job List for provider clients.

Ulafix allows REST user providers to get a list of all closed jobs to implement logic to forcefully close the jobs by auto provider.

CLOSE JOB : Get list of auto part jobs as per provider user, which need to be closed as per logic by the provider user.	**GET** / `auto-parts/client-auto-part-closed-job`

Authorization: REQUIRED

The API uses HTTP Bearer Authentication and requires `access_token` *(which is available in the **Login API** response after a successful login request)* to be sent in the request as part of the HTTP

Response Example:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
{
  "items":
    [
        {
```

header in following format:
authorization: Bearer ***<token>***

Request Example:

```
curl -X GET \
      --url
http://<domain>/api/web/v1/ auto-
parts/ client-auto-part-closed-job
\
      -H 'authorization: Bearer
      X_J8kwflDQH7sTbRBX-UxvAU5R0Jq4qC'
      \
      -H 'cache-control: no-cache'
```

Note: You can customize your logic by using the given list to force a client to closed a job. Here `company_job_status` is conditionally using to implement the logic to the client.

Please refer the Client activity and Provider activity API to understand the status.


```
                    "jobid": "59",
                    "job_created_by": "22",
                    "job_name": "Tyres-06Jun2016",
                    "client_status": "3",
                    "awarded_company_to": "89",
                    "company_status": "4"
            },
    ]
}
```


Emergency Search

Ulafix allows REST clients to perform Emergency Search based on criteria like zipcodes, service type & radius etc.

SEARCH : Search emergency service provider based on search criteria | **GET** /emergency-services/search

PARAMETERS (QUERY STRING)

`service_type_id`	**required**	DB id of service_type table for Emergency services *(available in LIST response structure of **Services** API)*
`search_radius`	optional	Integer value of radius of search from given zipcode
`zipcode`	**required**	DB id of zipcode table *(available in LIST/SEARCH response structure of **Zipcode** API)*
`from_schedule`	**required**	Schedule from date-time in specific format (e.g. 2016-01-27 15:00) By default set current data/time.
`to_schedule`	**required**	Schedule to date-time in specific format (e.g. 2016-01-27 16:00) By default set one hour later of `from_schedule`

Request Example:

```
curl -X GET \
      --url
http://<domain>/api/web/v1/
emergency-
services/search?service_type_id=8&sea
rch_radius=100&zipcode=38896&from_s
chedule=2016-06-30
15:00&to_schedule=2016-06-30 16:00 \
      -H 'authorization: Bearer
      jnUK6JW9JUNB4eUp5YPvQkQh4jSwR9h5'
      \
      -H 'cache-control: no-cache'
```

Response Example:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
```


```
{
    "items": [
        {
                "user_id": 24,
                "account": {
                "id": 24,
                "username": "new.provider",
                "account_activation_token": null,
                "email": "new@nptest.com",
                "role": "Provider",
                "status": "Active",
                "policy_agreed": 1,
```

Using the `fields` parameters, you may also specify which fields should be included in the result.
For example, the URL
`http://<domain>/api/web/v1/emergency-services/search?service_type_id=7&search_radius=100&zipcode=92102&from_schedule=2016-01-27 15:00&to_schedule=2016-01-27 16:00&fields=user_id,account, contacts`
will only return the `user_id` and `account` , `contacts` fields in the response.


```
                "created_at": 1439363594,
                "updated_at": 1440133120,
                "role_numeric": 2,
                "status_numeric": 1
            },
            "contacts": {
                "id": 24,
                "user_id": 24,
                "address_line1": "5-4-14/3, JN
        Roads",
                "address_line2": "Opposite Rama
        Krishna Theatre",
                "city": "Hyderabad",
                "state": "Telangana",
                "country_id": 105,
                "zipcode_id": 557,
                "latitude": "17.385733",
                "longitude": "78.475530",
                "web_url": null,
                "alt_email": null,
                "contact_number": "9186573874",
                "alt_contact_number": null,
                "country": "India",
                "zip": "500001"
            }
        ...
        ...
        {
            "user_id": 24,
            ...
        }
    ]
}
```


SENT SMS NOTIFIATION : Sent sms to available all emergency providers. | **POST** /sent-sms-notifiaction

PARAMETERS (QUERY STRING)

`provider_ids`	**required**	DB id of providers *(available in SEARCH response structure of **Emergency Search** API)*
`contact_number`	**required**	Ten digit phone number (eg:1234567899)
`zipcode_id`	**required**	DB id of zipcode table *(available in LIST/SEARCH response structure of **Zipcode** API)*
`service_id`	**required**	DB id of service_type table for Emergency services *(available in LIST response structure of **Services** API).*
`model_year_id`	**required**	DB id of car_model_year table *(available in LIST response structure of **Car Year** API)*
`current_location`	optional	Address string of user current location. *(Max. **70** characters)*

Request Example:

Request Example:

```
curl -X POST \
--url http://<domain>/api/web/v1/emergency-services/sent-sms-notifiaction \
-H 'authorization: Bearer X_J8kwflDQH7sTb' \
  -H 'cache-control: no-cache' \
  -H 'content-type: multipart/form-data; boundary=---011000010111000001101001' \
```

Response Example:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
{
      "isSuccess": 1,
       "message": "Sent notifications to all listed providers."
}
```

`-F provider_ids=45,78 \`
`-F contact_number=8521479639 \`
`-F zipcode_id=558 \`
`-F service_id=8 \`
`-F model_year_id=669 \`
`-F current_location='plot 7,Madhapur, piller 9 opp IOC petrol pump' \`
 	
Note that provider_ids can take multiple values as comma separated string as `provider_ids=45,78` in the request.	