

## [Freshworks CRM App Reference](#)

[Overview](#)

[Audience](#)

## [GETTING STARTED](#)

[Quick Start](#)

[Install NVM](#)

[Install Node](#)

[Install the Freshworks CLI](#)

[Create an App](#)

[Test the App](#)

[Validate and Pack](#)

## [REFERENCE](#)

[Freshworks CLI](#)

[Create](#)

[Generate](#)

[Run](#)

[Validate](#)

[Pack](#)

[Version](#)

[External Libraries](#)

[JWT](#)

## [CONFIGURATION](#)

[App Manifest](#)

[Placeholders](#)

[Global Navigation Pane](#)

[Contact Details Page](#)

[Deals Details Page](#)

[Sales Account Details Page](#)

[Installation Parameters](#)

[Configure](#)

[Types](#)

[Retrieve](#)

[Test](#)

[Custom Installation Page](#)

[create](#)

[postConfigs](#)

[getConfigs](#)

[validate](#)

[retrieve](#)

[Test](#)

## [BASIC FEATURES](#)

### [App Lifecycle Methods](#)

[app.initialized\(\)](#)

[app.activated\(\)](#)

[app.deactivated\(\)](#)

### [Data Methods](#)

#### [Global Data Methods](#)

[loggedInUser](#)

[domainName](#)

[Contact Details page](#)

[Sales Account Details page](#)

[Deals Details page](#)

### [Events Methods](#)

[Contact Details page](#)

[Sales Account Details page](#)

[Deals Details page](#)

### [Interface Methods](#)

[Show Modal](#)

[Show Confirm](#)

[Show Notifications](#)

[Open Create Contact](#)

[Open Create Account](#)

[Open Create Deal](#)

[Open Create Call Log](#)

[Show Contact](#)

[Show Account](#)

[Show Deal](#)

## [ADVANCED FEATURES](#)

### [Data Storage](#)

#### [OAuth](#)

### [Request Method](#)

[Usage](#)

[Sample Requests](#)

[OAuth Request](#)

[Testing](#)

[IP Ranges](#)

[Errors](#)

[Instance Method](#)

[Context](#)

[Get](#)

[Send](#)

[Receive](#)

[Use Cases](#)

## [SERVERLESS](#)

[Overview](#)

[Create a Serverless App](#)

[Dependencies](#)

[Test the App](#)

[App Setup Events](#)

[External Events](#)

[Product Events](#)

[Configure Events](#)

[onContactCreate](#)

[onContactUpdate](#)

[onDealCreate](#)

[onDealUpdate](#)

[onSalesAccountCreate](#)

[onSalesAccountUpdate](#)

[onCustomModuleCreate](#)

[onCustomModuleUpdate](#)

[onAppointmentCreate](#)

[onAppointmentUpdate](#)

[onTaskCreate](#)

[onTaskUpdate](#)

[Testing](#)

[Testing an Omni App](#)

[Scheduled Events](#)

[Server Method Invocation](#)

## [ERRORS and VALIDATIONS](#)

[Troubleshooting](#)

[Lint Validations](#)

[GUIDELINES](#)

[UI Style Guide](#)

[App Content Guidelines](#)

[Code Review Guidelines](#)

## [MARKETPLACE LISTING](#)

[Types of Apps](#)

[Freshworks Apps](#)

[Custom Apps](#)

[External Apps](#)

# Freshworks CRM App Reference

## Overview

This document provides an overview of the steps involved in building and submitting a Freshworks CRM app. This is an early access document and contains references to the Freshsales App Reference document for overlapping information between Freshworks CRM and Freshsales. Developers who have existing apps on Freshsales can refer to the [App Manifest](#) and [Placeholders](#) sections of this document for information on how to migrate their apps to Freshworks CRM. Developers can also refer to the [App Manifest](#) and [Placeholders](#) sections of this document to build apps that are compatible with both Freshworks CRM and Freshsales.

## Audience

The intended audience for this document are:

- Developers who want to have an early start with building apps for FCRM.
- Developers who want to migrate their existing Freshsales apps to FCRM.
- Developers who want to have an early start with building omni apps.

# GETTING STARTED

## Quick Start

You can build a Freshworks CRM app by using the following steps.

1. [Install NVM](#)
2. [Install Node](#)
3. [Install the Freshworks CLI](#)
4. [Create an App](#)
5. [Test the App](#)
6. [Validate and Pack](#)

Apps built on the Freshworks platform are compatible with the latest and immediately preceding versions of the following browsers.

- Google Chrome
- Firefox
- Edge
- Safari

## Install NVM

Node Version Manager (NVM) enables you to install and work with multiple versions of Node.js. You can use NVM to install Node.

1. To install NVM:
  - On Mac or Linux, follow the [installation and upgrade steps](#).
  - On Windows, navigate to the [release channel](#) > **Assets**, download the **nvm-setup.zip** file, extract the contents of the file, and use the installer in the extracted files.

2. Run the following command to verify the NVM installation.

```
$ nvm --version
```

## Install Node

1. To install Node using NVM, run the following command.

```
$ nvm install 10.18
```

**Note:** Support for Node 8.10.x has been deprecated.

2. Run the following command to verify the Node installation.

```
$ node --version
```

**Note:** On Windows, if the Node version is not displayed, you should run the `nvm on` command to enable nvm.

3. Run the following command to set the default Node version.

```
$ nvm alias default 10.18
```

4. On Windows, to install the tools and configurations required to start using Node, use the following command.

```
$ npm install --global --production windows-build-tools
```

For more information, see the [write-up on node-gyp](#).

## Install the Freshworks CLI

**Note:** The Developer portal and SDK [Terms of Use](#) apply to the use of the CLI.

1. Run the following command to install the CLI.

```
npm install https://dl.freshdev.io/cli/fdk.tgz -g
```

2. Run the following command to verify the CLI installation.

```
$ fdk version
```

## Create an App

You can use the following steps to create an app that displays a logo and sample text in the **INTEGRATIONS** section of the Contact Details page.

1. From the command line, navigate to the empty directory in which you want to create an app.
2. Run the following command: `$ fdk create`  
A prompt to choose a product is displayed.
3. Select **freshworks\_crm** and press **Enter**. A new app is created based on the **your\_first\_app** template.

**Note:** To create an Omni app, select **omni** and press **Enter**. A new app is created based on the **your\_first\_app** template.

The following directories and files are created as a result of the `fdk create` command.

DIRECTORY/FILE	DESCRIPTION
app/	Contains all the files required for the front-end component of an app. The JS file follows the ES5 standard.
app/app.js	Contains the app logic that renders a logo and sample text in the INTEGRATIONS section of the Contact Details page.
app/icon.svg	Contains the product icon. If you intend to extend the app, you can replace the icon.svg file. The icon file should be of SVG type with a resolution of 64 x 64 pixels.
app/template.html	Contains the HTML code required for the app's UI, which is rendered in an IFrame.
app/modal.html	Contains code to receive and send information between the Freshworks CRM UI and app instance.



config/	Contains the installation parameters and OAuth configuration files.
config/iparams.json	Contains all the installation parameters whose values are set when the app is installed. For more information, see <a href="#">Installation Parameters</a> .
manifest.json	Contains details such as the platform version the app uses, product to which the app belongs, and npm packages that the app uses (dependencies).
README.md	Contains additional instructions, information, and code-related specifications pertaining to the app

**Note:** If you use React, Ember, Vue, or any other front-end framework, include the source files of your app in the **src** directory, to ensure quick app reviews.

## Test the App

### Notes:

- To test your app, use the latest version of Chrome.
  - To request for a trial Freshworks CRM account to test your app, email us at [marketplace@freshworks.com](mailto:marketplace@freshworks.com).
1. From the command line, navigate to the directory that contains the app related files and run the following command: `$ fdk run`
  2. Log in to your Freshworks CRM account. To the Freshworks CRM account URL, append **?dev=true**.  
Example URL: **`https://domain.myfreshworks.com/crm/sales?dev=true`**
  3. To allow the Chrome browser to connect to the test server that runs on HTTP, On Chrome 79 and higher versions:
    1. Navigate to **Settings -> Advanced -> Privacy and security -> Site settings -> Insecure content**.

2. In the **Allow** section, click **Add** and enter the Freshworks CRM account URL.  
Example URL: **https://domain.myfreshworks.com/crm/sales**

On Chrome 78 and lower versions:

1. In the address bar, if a shield icon is displayed, click the icon. A warning message is displayed as the Support portal runs on HTTPS and the test server runs on HTTP.
2. Click **Load Unsafe Scripts** to continue testing.
4. On the Home page, from the left global sidebar, select Contacts. The Contacts page is displayed. Click a contact. The app is rendered in the INTEGRATIONS section of the Contact Details page.

#### Notes:

1. Each component in the coverage summary should be **at least 80%** for apps to be submitted in the Freshworks Marketplace. See [Code coverage](#) for more information.
2. If you have any issues when testing your app, attach [detailed logs](#) of the output in your support ticket for quick resolution from the support team.

## Validate and Pack

To check if the app is error-free and package it for submission, follow the given steps.

1. To validate the code, run the following command.

```
$ fdk validate [--app-dir DIR]
```

Here, **DIR** is the relative or absolute path to the app directory. If there are errors in the code, corresponding violations are displayed.

To auto-fix certain errors and display only those that need manual fixing, use the

```
$fdk validate --fix
```

 command. Ensure to fix all violations before packing the app.

2. To pack the app for submission, run the following command.

```
$ fdk pack [--app-dir DIR]
```

The command generates the **dist/<app\_directory>.zip** file.

3. To publish the app to the Freshworks Marketplace, navigate to the Freshworks CRM developer portal and upload the packed file. For more information on the app submission process, see [Types of Apps](#).

# REFERENCE

## Freshworks CLI

The Freshworks Command-Line Interface (CLI) enables you to build, test, and package apps that are published to the Marketplace.

- To view the list of CLI commands, after you [install the CLI](#), at the command prompt type **fdk** and press **Enter**.
- To view the information about a CLI command, run the following command: **\$ fdk COMMAND (-h | --help)**
- To view the detailed logs that pertain to a CLI command, run the following command: **\$ NODE\_DEBUG=fdk fdk COMMAND**

## Create

The command creates an app in the specified directory, based on the default app template. If no directory is specified, the app is created in the current directory.

**Note:** Ensure that the directory in which the app is to be created is empty.

**\$ fdk create [--app-dir DIR] [--products PRODUCT] [--template TEMPLATE]**

OPTIONS	USAGE
--app-dir	To specify the absolute or relative path of the project directory where the app is to be created. If the option is not used with the create command, the app is created in the current directory.

<code>--products</code>	<p>To specify the product for which the app is created. If the option is not used with the create command, you are prompted to select a product.</p> <p>Select <b>freshworks_crm</b>.</p> <p>To create an omni app that is compatible with freshworks_crm and freshsales, select <b>omni</b>.</p>
<code>--template</code>	<p>To specify the template for which the app is created. If the option is not used with the create command, you are prompted to select a template.</p> <p>Select <b>your_first_app</b> to create a front-end app and <b>your_first_serverless_app</b> template to create a serverless app.</p>

### Example 1

To create a Freshworks CRM in the current directory, run the following command. Before you run the command, ensure that the current directory is empty.

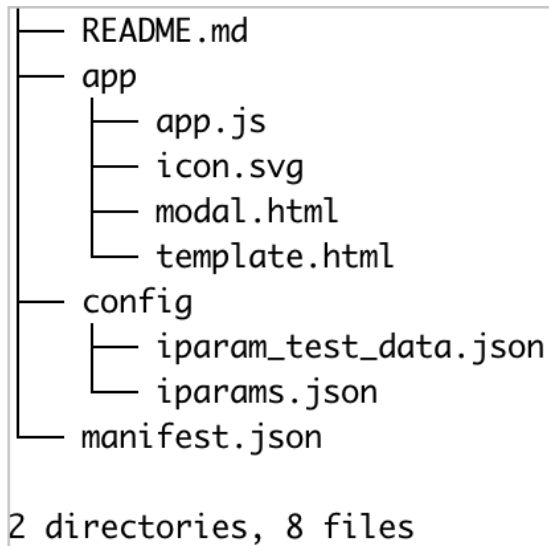
```
$ fdk create --products freshworks_crm --template your_first_app
```

To create an omni app in the current directory, run the following command. Before you run the command, ensure that the current directory is empty.

```
$ fdk create --products omni --template your_first_app
```

### Sample Output

A Freshworks CRM app is created based on the **your\_first\_app** template, with the following files.



## Example 2

To create a Freshworks CRM or omni app in the **myfirstapp** directory, run one of the following commands. Before you run the command, ensure that the **myfirstapp** directory is empty.

- `$ fdk create --app-dir ./myfirstapp`
- `$ fdk create --app-dir /Users/user/myfirstapp`

A prompt is displayed asking you to select the product and template. To create a Freshworks CRM app, enter **freshworks\_crm** and to create an omni app, enter **omni** as the product. To create a front-end app select **your\_first\_app** as the template.

## Output

The [app files](#) are created in the **/Users/user/myfirstapp** directory.

## Generate

This command is used to generate app files with a specific template for an existing app that was created by using the **create** command.

1. After you create an app, run the following command: `$ fdk generate`

A list of the following files is displayed.

- **oauth\_config.json**: Contains the configuration parameters, such as client\_id, client\_secret, authorize\_url, token\_url, and account/agent scope, required for OAuth-based authentication.
- **iparams.json**: Contains the installation parameters, such as iparam name and type of iparam, whose values are set when the app is installed.
- **iparams.html**: Contains the html code to build the Custom Installation page. You can enter the page title, select the product css to link, and choose to add jquery, fresh\_client.js, and iparam.js and iparam.css.
- **server.js**: Contains the event registration and callback methods. You need to enter the names of the events and callback functions.
- **manifest.json**: Contains app information such as platform version, product, location, event registration, and so on. You need to select the product and app location.

2. Select a file and enter the appropriate parameter values. The corresponding file is generated based on the values.

## Run

The run command starts the local server, deploys the app project on the server, and enables you to [test the app](#).

```
$ fdk run [--app-dir DIR]
```

OPTION	USAGE
--app-dir	To specify the absolute or relative directory path to the app project that is to be deployed on the local server.

If you are running a serverless app that contains dependencies, the packages defined in [manifest.json](#) are automatically installed.

After testing, press **Control + C** to stop the server. A code coverage summary that indicates the extent of testing is displayed. A coverage summary that indicates extensive testing is as follows.

```
===== Coverage summary =====  
Statements   : 98.18% ( 54/55 )  
Branches     : 100% ( 4/4 )  
Functions    : 84.62% ( 22/26 )  
Lines        : 98.18% ( 54/55 )  
=====
```

**Note:** When apps are submitted for the App Review process, ensure that in the coverage summary, the coverage of each component is at least 80%.

### Example 1

To deploy the app project located in the current directory, run the following command: `$ fdk run`

### Output

```
Starting local testing server at http://*:10001/  
Append "dev=true" to your Freshworks_crm account URL to start testing  
e.g.  
https://domain.myfreshworks.com/crm/sales/contacts/2007663673?dev=true  
Quit the server with Control-C.
```

### Example 2

To deploy the app project located in the **myfirstapp** directory, run the following command:

```
$ fdk run --app-dir /Users/user/myfirstapp
```

The [output](#) is the same as that for Example 1.

### Example 3

To deploy a serverless app project located in the **myfirstapp** directory, run the following command:

```
$ fdk run --app-dir /Users/user/myfirstapp
```

## Output

```
$ Starting local testing server at http://*:10001/
```

Append "dev=true" to your Freshworks\_crm account URL to start testing

e.g.

```
https://domain.myfreshworks.com/crm/sales/contacts/2007663673?dev=true
```

Quit the server with Control-C.

To simulate product, app setup, and external events, visit -

```
http://localhost:10001/web/test
```

To test the installation page, visit -

```
http://localhost:10001/custom_configs
```

## Validate

This command validates whether the app code is error free. If there are errors in the code, corresponding violations are displayed after the command is run.

```
$ fdk run --app-dir [--fix]
```

OPTION	USAGE
--app-dir	To specify the absolute or relative path to the app directory that contains the files to be validated.
--fix	To fix all the lint errors for which auto-fix support is available.



## Pack

The **pack** command creates an app package file that can be submitted for the app review process.

```
$ fdk pack [--app-dir DIR]
```

OPTION	USAGE
--app-dir	To specify the absolute or relative path to the app project directory that contains the app files to be packed. If the option is omitted from the pack command, the files in the current directory are packed.

After you run the **pack** command, a **dist/<app\_directory>.zip** file is generated. To submit the app and get it published to the Freshworks Marketplace, upload the generated file to the Freshworks CRM developer portal.

### Example 1

To pack the app files located in the current app project directory, run the following command: \$

```
fdk pack
```

The **/dist/<current\_app\_directory>.zip** file is generated.

### Example 2

To pack the app project files located in the **myfirstapp** directory, run the following command:

```
$ fdk pack --app-dir /Users/user/myfirstapp
```

The **/dist/myfirstapp.zip** file is generated.

## Version

The **version** command displays the installed and latest CLI version numbers.

```
$ fdk version
```

## Output

```
Installed: 6.0.0
```

```
Latest: 6.6.0
```

```
To install the latest version type
```

```
"npm install https://dl.freshdev.io/cli/fdk.tgz -g"
```

## [External Libraries](#)

### [JWT](#)

# CONFIGURATION

## App Manifest

The **manifest.json** file contains information such as the platform version used by a product, locations on the product UI where an app can be rendered, dependent files used by the app, and third-party domains with which the app can interact. This section details the various parameters of the **manifest.json** file.

**Note:** To migrate an existing Freshsales app as an FCRM app, use the following **manifest.json** file format.

### Sample manifest.json

```
{
  "platform-version": "2.1",
  "product": {
    "freshworks_crm": {
      "location": {
        "contact_entity_menu": {
          "url": "template.html",
          "icon": "icon.svg"
        }
      }
    }
  }
}
```

For omni apps (apps that are compatible with Freshsales and Freshworks CRM) the default **manifest.json** file format is as follows:

**Note:** To migrate an existing Freshsales app as an omni app, use the following **manifest.json** file format. Also, specify the placeholders supported in [Freshsales](#) and [Freshworks CRM](#) appropriately in the **location** object.

### Sample manifest.json for omni apps

```
{
  "platform-version": "2.1",
  "product": {
    "freshworks_crm": {
      "location": {
        "contact_entity_menu": {
          "url": "template.html",
          "icon": "icon.svg"
        }
      }
    }
    "freshsales": {
      "location": {
        "contact_entity_menu": {
          "url": "template.html",
          "icon": "icon.svg"
        }
      }
    }
  }
}
```

```
}
```

### Sample manifest.json for serverless omni apps

```
{
  "platform-version": "2.1",
  "product": {
    "freshworks_crm": {
      "events": {
        "onDealCreate": {
          "handler": "onEventHandler"
        },
        "onExternalEvent": {
          "handler": "onExternalEvent"
        }
      }
    },
    "freshsales": {
      "events": {
        "onDealCreate": {
          "handler": "onEventHandler"
        },
        "onExternalEvent": {
          "handler": "onExternalEvent"
        }
      }
    }
  }
}
```

## Attributes

ATTRIBUTE NAME	DATA TYPE	DESCRIPTION
platform-version	string	Specifies the platform version used to build the app, auto-generated when the app files are created. The version information is used to ensure backward compatibility when a new platform version is released.
product mandatory	object	Associates a Freshworks product with the information that is necessary to render the app on the specified product. (Child attribute: freshworks_crm) (Child attributes for omni apps: freshworks_crm, freshsales)
freshworks_crm mandatory	object	Specifies information that is necessary to render the app on the Freshworks CRM UI. (Child attribute: location or events (for severless, omni apps))
freshsales mandatory	object	Specifies information that is necessary to render the app on the Freshsales UI. (Child attribute: location or events (for severless, omni apps))
location	object	Specifies the locations on the product(s) UI, where the app is to be rendered. For more information, see <a href="#">Placeholders</a> <b>Attributes:</b> url, icon <b>Note:</b> For omni apps, the placeholders specified in freshworks_crm.location and freshsales.location need not be the same.

url mandatory	string	<p>Specifies the relative path from the app project directory to the HTML file based on which the app is rendered. The contents of the HTML file are used to render the app in an IFrame at the specified location.</p> <p><b>Default value:</b> <b>template.html</b> (file in the <b>app</b> directory)</p>
icon mandatory	string	<p>Specifies the relative path from the app project directory to an image file (in SVG format). The image is used as the app's icon when the app is rendered on the UI. Ensure that the resolution of the image is 64 x 64 pixels.</p> <p><b>Default value:</b> <b>icon.svg</b> (file in the <b>app</b> directory)</p>
events mandatory for serverless, omni apps	object	<p>Contains the various serverless events and the corresponding call-back function names, specified in the following format:</p> <pre>"events": {   "&lt;eventName1&gt;": {     "handler": "&lt;callbackFunctionName1&gt;"   },   "&lt;eventName2&gt;": {     "handler": "&lt;callbackFunctionName2&gt;"   } }</pre> <p>For more information, see the <a href="#">Overview</a> section in Serverless.</p>

## Placeholders

On a page of the Freshworks CRM UI, there are various locations where an app can be rendered. This section specifies the locations and how to configure the **manifest.json** file, for an app to be displayed in the specified location. An app can be rendered in multiple locations.

### Notes:

1. Freshworks CRM apps are supported as web apps.
2. For omni apps, specify valid [Freshsales](#) and Freshworks CRM placeholders in the appropriate **location** object. For information on the manifest.json structure for omni apps, see [Sample manifest.json for omni apps](#).
3. To migrate a Freshsales app as an FCRM app, replace the Freshsales placeholders with the appropriate Freshworks CRM placeholders, in the **location** object.

## Global Navigation Pane

**full\_page\_app** - The app icon is displayed on the left navigation pane and when clicked, the app page occupies the entire viewport.

### manifest.json

```
"product": {
  "freshworks_crm": {
    "location": {
      "full_page_app": {
        "url": "myfirstapp.html",
        "icon": "logo.svg"
      }
    }
  }
}
```



**Notes:**

1. For a Freshworks CRM account, a maximum of 10 full-page apps can be installed. The **Full-Page App** icon is displayed on the left navigation pane. App users can hover over the icon to view the list of all installed full-page apps.
2. The logo.svg file should have a resolution of 64 x 64 pixels and fill value set to FFF.

## Contact Details Page

**contact\_entity\_menu** - The app is displayed in the INTEGRATIONS section of the Contact Details page.

**manifest.json**

```
"product": {  
  "freshworks_crm": {  
    "location": {  
      "contact_entity_menu": {  
        "url": "myfirstapp.html",  
        "icon": "logo.svg"  
      }  
    }  
  }  
}
```

## Deals Details Page

**deal\_entity\_menu** - The app is displayed in the INTEGRATIONS section of the Deal Details page.

## manifest.json

```
"product": {  
  "freshworks_crm": {  
    "location": {  
      "deal_entity_menu": {  
        "url": "myfirstapp.html",  
        "icon": "logo.svg"  
      }  
    }  
  }  
}
```

## Sales Account Details Page

**sales\_account\_entity\_menu** - The app is displayed in the INTEGRATIONS section on the Sales Account Details page.

## manifest.json

```
"product": {  
  "freshworks_crm": {  
    "location": {  
      "sales_account_entity_menu": {  
        "url": "myfirstapp.html",  
        "icon": "logo.svg"  
      }  
    }  
  }  
}
```

# Installation Parameters

The FDK enables you to define and use parameters whose values app users can set when they install an app. These parameters are termed installation parameters or iparams. To define and use iparams:

1. [Configure the iparams](#). During app installation, the configured iparams are displayed on an installation page. App users can enter appropriate values for the iparams and click INSTALL. The entered values are validated and saved.
2. In the app logic, use the `client.iparams.get()`, `client.iparams.get(iparam_key)`, or the [Request method](#) (for secure iparams), to retrieve the iparam values.

**Note:** To prevent exposure of sensitive iparam information, secure iparams can only be retrieved by the Request method.

## Configure

1. From your app's root directory, navigate to the **iparam.json** file.
2. Configure iparams by using the following sample format.

```
{
  "contact": {
    "display_name": "Contact Details",
    "description": "Please enter the contact details",
    "type": "text",
    "required": true
  },
  "age": {
    "display_name": "Age",
    "description": "Please enter your age in years",
    "type": "number"
  }
}
```

```

    },
    "contactType": {
      "display_name": "Contact Type",
      "description": "Please select the contact type",
      "type": "dropdown",
      "options": [
        "Phone",
        "Email"
      ],
      "default_value": "Email"
    },
    "domainName": {
      "display_name": "Domain Name",
      "description": "Please enter your domain name",
      "type": "domain",
      "type_attributes": {
        "product": "freshworks_crm"
      },
      "required": true
    },
    "apiKey": {
      "display_name": "API Key",
      "description": "Please enter your api_key",
      "type": "api_key",
      "secure": true,
      "required": true,
      "type_attributes": {
        "product": "freshworks_crm"
      }
    }
  }
}

```

Configure each iparam as a JSON object with the following attributes.

**Note:** If you are migrating an existing Freshsales app as an FCRM app or omni app, for **domain** and **api\_key** type iparams, ensure to specify the appropriate **type\_attributes.product** value.

ATTRIBUTE	DATA TYPE	DESCRIPTION
display_name mandatory	string	Identifier of a parameter on the installation page.
description	string	Helper text that is displayed along with the parameter, on the installation page. The description can include examples.
type mandatory	string	Type of input field displayed, for the iparam, on the installation page.  <b>Possible values:</b> text, paragraph, dropdown, email, number, phone_number, date, url, radio, checkbox, multiselect, domain, api_key  For information on the available types of input fields, see Types.
options	array of strings	List of values displayed on the installation page, if the iparam's input field type is <b>radio</b> , <b>multiselect</b> , or <b>dropdown</b> .
default_value	string	Preselected value(s) (from options) displayed on the installation page, if the iparam's input field type is <b>radio</b> , <b>multiselect</b> , or <b>dropdown</b> .
required	boolean	Specifies whether the iparam is displayed as a mandatory parameter. An asterisk is displayed next to the parameter on the installation page.  <b>Possible values:</b> true, false
visible	boolean	Specifies whether the iparam is displayed on the installation page.  <b>Possible values:</b> true, false  <b>Example</b>

		<pre> {   "contact": {     "display_name": "Contact Details",     "description": "Please enter the contact details",     "type": "text",     "required": true,     "visible": true   } } </pre>
secure	boolean	<p>Specifies whether the iparam is used to collect sensitive data, which has to be protected from exposure through the app's front-end components.</p> <p><b>Possible values:</b> true, false</p> <p><b>Example</b></p> <pre> {   "apiKey": {     "display_name": "Api Key",     "type": "text",     "required": true,     "secure": true   } } </pre> <p><b>Note:</b> Sensitive iparams, such as key, auth, token, secret, are detected when running the fdk validate and fdk pack commands and a warning is displayed to mark them as secure.</p> <p>For comprehensive information on secure iparams, read the <a href="#">securing sensitive installation parameters</a> blog post.</p>
regex	object	<p>Expression(s) used to validate a parameter value entered in the installation page. The regex validation is performed in addition to the default field-type validation. If the validation fails, an error message is displayed. The regex expression and associated error message formats are as follows:</p>

		<pre>"&lt;regex_name&gt;": "&lt;valid_regex_expression&gt;" "&lt;regex_name_error&gt;": "&lt;error_message&gt;"</pre> <p><b>Example regex expression to verify that the age entered is between 10 and 99</b></p> <pre>{   "Age": {     "display_name": "Age",     "description": "Please enter your age in years",     "type": "text",     "regex": {       "age-limit": "[1-9][0-9]",       "age-limit-error": "The age must be between 10-99"     }   } }</pre>
events	array of objects	<p>HTML events and associated callbacks, specified as an array of JSON objects. Each JSON object is an "&lt;event name&gt;": "&lt;callback function name&gt;" pair. If an event is associated with an iparam, when the event occurs in the installation page, the associated callback function is invoked.</p> <p><b>Note:</b> Currently, only change events are supported. Therefore, ensure that the &lt;event name&gt; is <b>change</b>. For information on how to define callback functions, see the <a href="#">make your installation page dynamic section</a>.</p>
data-bind	string	<p>Product values (domain name or API key associated with an account) used to pre-populate an iparam field.</p> <p><b>Example</b></p> <pre>{   "apikey": {     "display_name": "API key",     "description": "Please enter your account's API key",     "type": "text",     "data-bind": "product.api_key",     "required": true   },   "domain": {     "display_name": "Product Domain",     "description": "Please enter your account's Domain",     "type": "text",     "data-bind": "product.domain",     "required": true   } }</pre>

		<p>If the iparam is of <b>text</b> type, <b>product.domain</b> populates the field with <b>&lt;sub-domain&gt;.myfreshworks.com/crm</b>.</p> <p>If the iparam is of <b>text</b> type and if the app is installed on an FCRM account that is migrated from Freshsales, <b>product.domain</b> populates the field with <b>&lt;sub-domain&gt;.freshworks.com/crm</b>.</p> <p>If the iparam is of <b>text</b> type and if the app is an omni app that is installed on a Freshsales account, <b>product.domain</b> populates the field with <b>&lt;sub-domain&gt;.freshsales.io</b>.</p> <p>If the iparam is of <b>domain</b> type, <b>product.domain</b> populates the field with the <b>&lt;sub-domain&gt;</b> part of the domain URL. <b>&lt;sub-domain&gt;</b> is based on the business account on which the app is installed.</p>
<p>type_attributes</p> <p><b>Mandatory</b> if type is <b>domain</b> or <b>api_key</b></p>	object	<p>Attributes that help to populate certain sections of the input field (for an iparam of the <b>domain</b> type) and validate the sub-domain and api key obtained from an app user.</p> <p><b>Attribute of the object:</b></p> <p><b>product:</b> Identifier of the product for which the sub-domain or API key is obtained from the app user, when the app is installed. If this attribute value is <b>current</b>, the product is identified based on the product account on which the app is installed.</p> <p>For more information, see <a href="#">Types</a> &gt; domain.</p>

**Note:** Alternatively, you can use the [generate](#) command to interactively configure iparams.



## Types

**domain** - An input box to enter the sub-domain part of a domain URL is displayed. By default, the protocol and domain name part of the URL are displayed in the input box and are not editable. The domain name is generated based on the **type\_attributes.product** value that is specified as part of the JSON object used to define the iparam.



```
{
  "domainName": {
    "display_name": "Domain_FCRM",
    "description": "Enter the domain of your FCRM account",
    "type": "domain",
    "type_attributes": {
      "product": "freshworks_crm"
    },
    "required": true
  }
}
```

If the **type\_attributes.product** value is **freshworks\_crm**, in the input box that is displayed on the Installation page, the domain name part is populated as **myfreshworks.com**.

If the value is **freshworks\_crm** and if the app is installed on an FCRM account that has been migrated from Freshsales, the domain name part is populated as **freshworks.com**.

If the **type\_attributes.product** value is **current**, the domain name is populated based on the product on which the app is installed. If the app is installed on a Freshsales account, the domain name part is populated as **freshsales.io**. If the app is installed on an FCRM account, the domain name part is populated as **myfreshworks.com** or **freshworks.com** (if the account is migrated from Freshsales).

For omni apps, specify **type\_attributes.product** as **current**, to enable the app to retrieve and populate the domain name based on the product context.

For information on the other types of iparams, see [Configure](#) > Types.

## Make your installation page dynamic

You can build a product specific installation page by using the **onFormLoad()** and **onFormUnload()** functions. The **onFormLoad()** is a callback function that is triggered when an Installation page is loaded, during app installation. The **onFormUnload()** is a callback function that is triggered when the Settings page, where the app user enters values for the installation

parameters, is closed. On app initialization, the `client.context.productContext` is set based on the product account on which the app is being installed. The callback functions use `productContext` to render an Installation page with iparams whose attributes are set based on `productContext`. To do this:

1. From the app's root directory, navigate to the `config` directory and create the `assets/iparams.js` file.
2. Define the `onFormLoad()` and `onFormUnload()` callback functions by using the following sample format.

```
function onFormLoad() {
  app.initialized().then(
    function(client)
    {
      let product = client.context.productContext.name;
      if(product == 'freshsales')
      {
        //iparams definition for Freshsales

      }else if(product == 'freshworks_crm')
      {
        //iparams definition for FCRM

      }else{
        console.log('ERROR:Missing Expected product from client');
      }
    },
    function(error)
    {
      //If unsuccessful
      console.log('ERROR:Problem in fetching product from client');
    }
  );
}

function onFormUnload() {
  //logic to perform when the installation page is closed
}
```

3. To set or modify the attribute value of iparams, based on `productContext`, use `utils.set()`; in the callback function definition. For information on how to use `utils.set()`, see [iparam utility methods](#).

### Sample iparams.js to render product specific installation page

```
function onFormLoad() {
  app.initialized().then(
    function(client)
```

```

{
  let product = client.context.productContext.name;
  if(product == 'freshsales')
  {
    //Change display names based on product
    utils.set('Domain', {label: 'Freshsales Domain'});
    utils.set('ApiKey', {label: 'Freshsales API Key'});

    //Change hint based on product
    utils.set('Domain', {hint: 'Enter the sub-domain of your Freshsales
account'});
    utils.set('ApiKey', {hint: 'Enter API key of your Freshsales
account'});

    //Hide FCRM specific fields
    utils.set('FieldToDisplayInFCRM', {visible: false});

  }else if(product == 'freshworks_crm')
  {
    //Change display names based on product
    utils.set('Domain', {label: 'FCRM Domain'});
    utils.set('ApiKey', {label: 'FCRM API Key'});

    //Change hint based on product
    utils.set('Domain', {hint: 'Enter the sub-domain of your FCRM
account'});
    utils.set('ApiKey', {hint: 'Enter API key of your FCRM account'});

    //Hide Freshsales specific fields
    utils.set('FieldToDisplayInFreshsales', {visible: false});

  }else{
    console.log('ERROR:Missing expected product from client');
  }
},
function(error)
{
  //If unsuccessful
  console.log('ERROR:Problem in fetching product from client');
}
);
}

```

When you configure iparams, you can use the **events** attribute to build a dynamic installation page to collect values for the installation parameters. For more information, see [dynamic installation page](#).

## Retrieve

For information on how to retrieve the configured iparams and use them in app components, see [Retrieve the iparams](#).

## Test

To test the configured installation parameters:

1. From the command prompt, navigate to the app project folder, and run the following command: `fdk run`
2. If the app contains an Installation page, the following message is displayed:  
**To test the installation page, visit -**  
[https://localhost:10001/custom\\_configs](https://localhost:10001/custom_configs)
3. Navigate to the specified location. The Installation page is displayed.
4. For omni apps, a **Select Product** drop-down box is displayed. Select the product for which you want to test the installation parameters. The Installation page is displayed with product specific iparam fields.

**Note:** For an FCRM app, if a **domain** type iparam is configured, when testing the installation page, the domain part of the URL is updated as **myfreshworks.com**. For an omni app, if a **domain** type iparam is configured, when testing the installation page, the domain part of the URL is updated as **freshsales.io** or **myfreshworks.com**, based on the product selected from the **Select Product** drop-down list.

5. Enter appropriate values in the fields and click **INSTALL**.

## Custom Installation Page

To collect values for installation parameters, you can build and use a Custom Installation page with certain UI elements and workflows that are unavailable in a standard installation page. For example, a Custom Installation page can:

- Contain elements such as nested fields, toggles, sliders, date and time pickers, mappers, and so on.
- Make installation forms dynamic; the value of a parameter determines the other parameters to be displayed.
- Help perform client-side validation of the installation parameter values entered.

To create and use a Custom Installation page:

1. [Create](#) the **iparams.html** file, specifying all required HTML, CSS, and JavaScript (JS) information.
2. In the **iparams.html** file, add the [postConfigs](#) method to retrieve the iparams as form fields with values and store them.
3. In the **iparams.html** file, add the [getConfigs](#) method to retrieve the stored iparams' values alone and populate the Edit Settings page.
4. To perform a client-side validation of the iparams entered, in the iparams.html file, add the [validate](#) method.
5. [Retrieve](#) the stored iparams and use them in apps.

**Note:** To handle the secure iparam values entered in the Custom Installation page, the Request method is used. In SMI, when a serverless component is invoked, the iparams defined in a Custom Installation page are passed as a JSON payload.

## create

To create a custom installation page:

1. Navigate to the **config** directory, create an **iparams.html** file.

**Note:** Ensure that the **config** directory contains only one of the two files - **iparams.json** or **iparams.html**.

2. In the **iparams.html** file, ensure to include:
  - The required HTML, CSS, and JS dependencies.
  - The **fresh\_client.js** file, to enable access to certain platform features within the Custom Installation page.
  - A reference to the product style sheet, to maintain design consistency.
3. For omni apps, to render a product-specific Custom Installation page during run-time, in the **iparams.html** file, use the **client.context.productContext.name** attribute. The attribute specifies the name of the product (**freshsales** or **freshworks\_crm**) in which the app is being installed.

## Example

```
<script>
    var currentProduct = client.context.productContext.name;
    if(currentProduct == "freshsales"){
        // Manipulate DOM elements relevant to freshsales
    }
```

```

    }
    else {
        // Manipulate DOM elements relevant to freshworks_crm
    }
</script>

```

4. To retrieve product account specific information such as the domain name and api key associated with an account, in **iparams.html**, use a **text** type **input** element with the **data-bind** attribute.

### Example

```

<input type="text" name="api_key" data-bind="product.api_key">
<input type="text" name="domain" data-bind="product.domain">

```

**Note:** You can add external assets, such as .css and .js, in the **config/assets** folder to render the Custom Installation page.

### Sample iparams.html

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <link rel="stylesheet" type="text/css"
href="https://static.freshdev.io/fdk/2.0/assets/freshworks_crm.cs
s">
    <script
src="https://static.freshdev.io/fdk/2.0/assets/fresh_client.js"><
/script>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.0/jquery.mi
n.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/select2/4.0.6-rc.0/js
/select2.min.js"></script>
    <link
href="https://cdnjs.cloudflare.com/ajax/libs/select2/4.0.6-rc.0/c
ss/select2.min.css" rel="stylesheet" />
    <link rel="stylesheet" type="text/css"
href="./assets/iparams.css">
    <script src="./assets/iparams.js"></script>
    <style>

```

```

.dropdown {
    width: 10%;
    color: black;
    border: 5;
}
#error_div {
    color: red;
}
.select2-container {
    width: 70%;
}
.select2-container--default {
    width: 80% !important;
}
.select2-locked {
    padding: 0px !important;
}
</style>
<script type= "text/javascript">
    jQuery("#error_div").hide();
    jQuery("#department").select2({
        width: "resolve"
    });
</script>
<script type= "text/javascript">
    function getConfigs(configs) {
        jQuery("#error_div").hide();
        jQuery("input[name=api_key]").val(configs["api_key"]);

jQuery("input[name=first_name]").val(configs.requester["first_name"]);

jQuery("input[name=last_name]").val(configs.requester["last_name"]);

        for(var i= 0; i < configs.department.length; i++ ) {
            jQuery("#department option[value=" +
configs.department[i] + "']").attr("selected",true);
        }
        jQuery("#department").select2({
            width: "resolve"
        });
        if(configs.conditions) {

```

```

jQuery("input[name=\"condition\"]").attr("checked",false);
    for(var a= 0; a < configs.conditions.length; a++ ) {
        jQuery("input[name=\"condition\"][value="+
configs.conditions[a]+"").attr("checked",true);
    }
}
}
</script>
<script type= "text/javascript">
    function validate() {
        let isValid = true;
        var input = jQuery("input[name=api_key]").val();
        if(!input.match(/^[A-z]+$\/)) {
            jQuery("#error_div").show();
            isValid = false;
        }
        else {
            jQuery("#error_div").hide();
        }
        return isValid;
    }
</script>
<script type= "text/javascript">
    function postConfigs() {
        var requester = {};
        var department = [];
        var conditions = [];
        var api_key = jQuery("input[name=api_key]").val();
        var first_name =
jQuery("input[name=first_name]").val();
        var last_name = jQuery("input[name=last_name]").val();
        requester["first_name"] = first_name;
        requester["last_name"] = last_name;
        jQuery("#department option:selected").each(function(){
            company.push(jQuery(this).val());
        });

jQuery("input[name=\"condition\"]:checked").each(function(){
    conditions.push(jQuery(this).val());
});
return {
    __meta: {

```



```

        secure: ["api_key"]
    },
    api_key,
    requester,
    department: company,
    conditions: conditions
}
}
</script>
</head>
<body>
    <div class="requester-fields">
        <h3>Requester Fields</h3>
        <label for="api_key">API key</label>
        <input type="text" name="api_key">
        <span id="error_div" class="error" style="display:
none;">Please enter a valid input. Please enter only
alphabets.</span>
        <label for="first_name">First Name</label>
        <input type="text" name="first_name">
        <label for="last_name">Last Name</label>
        <input type="text" name="last_name">
    </div>
    <div class="account-fields">
        <h3>Department Fields</h3>
        <select class="select2-fields int-select
select2-offscreen" data-disable-field="Department"
data-placeholder=" " id="department" multiple="multiple"
name="company[]" rel="select-choice" tabindex="-1">
            <option value="Department">Department</option>
            <option value="City">City</option>
            <option value="Country">Country</option>
            <option value="Email">Email</option>
            <option value="Phone">Phone</option>
            <option value="PostalCode">Postal Code</option>
            <option value="State">State</option>
        </select>
    </div>
    <label class="checkbox-inline"><input name="condition"
type="checkbox" value="order"> Display orders from sample
app</label>

```

```

        <label class="checkbox-inline"><input name="condition"
type="checkbox" value="type"> Display type from sample
app</label>
    </body>
</html>

```

## postConfigs

To store the installation parameters/form fields configured through **iparams.html** and the corresponding values entered in the Custom Installation page, include this method in the **iparams.html** file. The method is triggered when, on the installation page, **INSTALL** is clicked. The method stores the iparams and corresponding values entered in the custom installation page, as JSON key-value pairs. Secure iparam values are passed using the meta tag in this method.

### Sample iparams.html

```

<script type= "text/javascript">
    function postConfigs() {
        var requester={};
        var department = [];
        var conditions = [];
        var first_name = jQuery("input[name=first_name]").val();
        var last_name = jQuery("input[name=last_name]").val();
        requester["first_name"] = first_name;
        requester["last_name"] = last_name;
        jQuery("#department option:selected").each(function() {
            department.push(jQuery(this).val());
        });
        jQuery("input[name='condition']:checked").each(function() {
            conditions.push(jQuery(this).val());
        });
        return {
            requester,
            department: department,
            conditions: conditions
        }
    }
</script>

```

## getConfigs

To retrieve the stored iparams' values and populate them on the Edit Settings page, include this method in the **iparams.html** file. The method is triggered when, on the Installed Apps Listing page, the Settings icon is clicked.

### Sample iparams.html

```
<script type= "text/javascript">
function getConfigs(configs) {
    jQuery("input[name=first_name]").val(configs["first_name"]);
    jQuery("input[name=last_name]").val(configs["last_name"]);
    for(var i= 0; i < configs.department.length; i++ ) {
        jQuery("#department option[value=" + configs.department[i] +
"]").attr("selected",true);
    }
    jQuery("#department").select2();
    if(configs.conditions) {
        jQuery("input[name="condition"]").attr("checked",false);
        for(var a= 0; a < configs.conditions.length; a++ ) {
            jQuery("input[name="condition"][value="+
configs.conditions[a]+""]).attr("checked",true);
        }
    }
}
</script>
```

## validate

To validate the values entered on the installation page or the Edit settings page, include this method in the iparams.html file. The method is triggered when:

- During installation, users enter iparam values and click **INSTALL** on the installation page.
- After installation, users edit iparam values and click **SAVE** on the Edit Settings page.

If the method returns **false**, the installation is stopped or the iparam values are not saved.

### Sample iparams.html

```

function validate() {
    let isValid = true;
    var input = jQuery("input[name=last_name]").val();
    if(!input.match(/^[A-z]+$/)) {
        jQuery("#error_div").show();
        isValid = false;
    }
    else {
        jQuery("#error_div").hide();
    }
    return isValid;
}

```

## retrieve

For information on how to retrieve the configured iparams and to use them in the app components, see the [Retrieve](#) section.

## Test

**Note:** To test the custom installation page on local settings, use the latest version of the FDK.

1. From the command prompt, navigate to the app project folder, and run the following command: **fdk run**
2. If the app contains a custom installation page, the following message is displayed:  
**To test the custom installation page, visit -**  
[https://localhost:10001/custom\\_configs](https://localhost:10001/custom_configs)
3. Navigate to the specified location. The Custom Installation page is displayed.
4. For omni apps, a **Select Product** drop-down list is displayed. Select the product for which you want to test the installation parameters. The Custom Installation page is displayed with product specific iparam fields.
5. Enter appropriate values in the fields and click **INSTALL** to test the Installation page.

# BASIC FEATURES

## App Lifecycle Methods

The Freshworks\_crm web interface is built as a single-page application. Unlike traditional web applications, a single-page app does not reload the entire page when the context is changed; it modifies only the relevant sections.

To enable an app to refresh its data at the appropriate time, the parent application emits an [app.activated\(\)](#) method. The method timing differs based on the app's location. When the page that contains the app is loaded for the first time (identified by the [app.initialized\(\)](#) method), the app needs to register to listen for the [app.activated\(\)](#) method.

### app.initialized()

The method is triggered when the page that contains the app is loaded for the first time. On success, it returns the client object that is used to register for the [app.activated\(\)](#) and [app.deactivated\(\)](#) methods. As the app renders within an IFrame, all communication (through [Data methods](#), [Interface methods](#), and [Events methods](#)) between the app and parent page occurs through the client object.

The client ([client.context](#)) object returned on app initialization, contains the [productContext](#) object that specifies the name of the product and URL of the business account on which the app is being installed. You can use the [client.context.productContext.name](#) and [client.context.productContext.url](#) attributes, to perform product specific actions such as rendering a product specific Installation page.

### Sample productContext

```
productContext: {  
  url: "https://sample.myfreshworks.com/crm",  
  name: "freshworks_crm"  
}
```

## app.js

```
app.initialized().then(  
    function(client)  
    {  
        //If successful, register the app activated and deactivated  
        event callback.  
        client.events.on("app.activated", onAppActivated);  
        client.events.on("app.deactivated", onAppDeactivated);  
    },  
    function(error)  
    {  
        //If unsuccessful  
        console.log();  
    }  
);
```

Unless you are building an app that is completely isolated (independent of the data on the page), ensure that the core logic of the app is not within the [app.initialized\(\)](#) callback. Place the logic within the [app.activated\(\)](#) callback, to ensure that the app can react when the parent application communicates. For example, on the Contact Details page, when you navigate from one contact to another, [app.activated\(\)](#) is triggered and the corresponding callback is run.

## app.activated()

The method is triggered when the app comes into scope and the method timing differs based on the location of the app.

LOCATION	WHEN IS THE METHOD TRIGGERED?
Contact Details page, Deal Details page, Sales Account Details page	When you click the app icon for the first time to display the app (apps in these locations are in a minimized state when the page is loaded).

**Note:** The `app.activated()` method is also triggered and the corresponding callback is executed when you navigate from one module to another on the respective module page.

### app.js

```
client.events.on("app.activated", onAppActivated);
function onAppActivated() {
    console.log("App Activated");
}
```

### app.deactivated()

The method is triggered when the app goes out of scope and can be used to clean stale data. In addition, this method is triggered when an agent moves from one module entity to the next and from one module page to the next.

### app.js

```
client.events.on("app.deactivated", onAppDeactivated);
function onAppDeactivated() {
    console.log("App Deactivated");
}
```

## Data Methods

You can use the data methods to retrieve information about the different objects on a page.

## Global Data Methods

The following objects can be used in all locations and retrieved using data methods

### **loggedInUser**

Returns details of the user who is logged into the Freshworks CRM UI.

```
client.data.get("loggedInUser").then (
  function(data) {
    // success operation
  },
  function(error) {
    // failure operation
  }
);
```

### **Sample Output Payload**

```
"loggedInUser":{
  "id":42528,
  "display_name":"Joe Doe",
  "email":"joe.doe@freshworks.com",
  "is_active":true,
  "work_number":null,
  "mobile_number":null,
  "confirmed":true,
  "privileges":"13221119178788",
  "job_title":"Engineer",
  "language":"en",
  "last_login_at":"2020-08-17T18:38:38+05:30",
  "time_zone":"Chennai",
  "avatar":null,
```



```
"signature":null,
"access_scope":"global",
"abilities":[
    "manage_contacts",
    "view_contacts",
    "create_contacts",
    "edit_all_contacts",
    "delete_all_contacts",
    "view_sales_accounts",
    "create_sales_accounts",
    "edit_all_sales_accounts",
    "delete_all_sales_accounts",
    "view_deals",
    "create_deals",
    "edit_all_deals",
    "delete_all_deals",
    "common_user_actions"
],
"reply_to":"name@mkp.myfreshworks.com",
"from":"joe.doe@freshworks.com",
"plan_name":"sales360_estate",
"imap_configured_email":"",
"deal_pipeline_id":29368,
"created_at":"2020-07-10T12:07:34+05:30",
"is_forgotten":false,
"role_id":70841,
"territory_ids":[],
"account_id":1594363054,
"sales_activity_type_ids":[
    78175,
    78176,
    78177,
    78178,
```

```

78179,
78180,
78181
]
}

```

### Attributes of the loggedInUser object

ATTRIBUTE	DATA TYPE	DESCRIPTION
id	number	Identifier of the user.
display_name	string	Display name of the user.
email	string	Email address of the user.
is_active	boolean	Specifies whether the user is active to accept new leads, contacts, tasks, appointments, and so on.
work_number	string	Official number of the user.
mobile_number	string	Mobile number of the user.
confirmed	boolean	Specifies whether the user has accepted the account activation mail.
privileges	string	Access privileges assigned to the user, specified through a role id.
job_title	string	Designation of the user.
language	string	Language of the user; by default, language is <b>en</b> .
last_login_at	string	Last logged timestamp of the user, specified in the UTC format.

time_zone	string	Time zone of the user.
avatar	string	Avatar URL.
signature	string	Email signature of the user in HTML format.
access_scope	string	Access scope of the user based on the role assigned to the user. The access can be global, restricted, or territory.
abilities	array	Actions the user can perform, based on the role assigned to the user.
reply_to	string	Email address in which the user can receive responses.
from	string	Email address using which the user can send emails from the Freshworks CRM account.
plan_name	string	FCRM plan under which the user's account falls.
imap_configured_email	string	User's email address that is connected to Freshworks CRM so that all prospect details are received directly in the product.
deal_pipeline_id	string	ID of preferred deal pipeline that is associated with the user.
created_at	string	Timestamp of when the logged in user's details are saved in FCRM, specified in the UTC format.
is_forgotten	boolean	Specifies whether the user details are permanently deleted from the FCRM system.
role_id	number	Identifier of the role associated with the user.

		When a role is configured in Freshworks CRM, an auto-generated identifier is associated with the role.
territory_ids	array of strings	Identifiers of the territories to which the user is associated for auto-assignment of prospects.
account_id	number	Identifier of the FCRM business account to which the user is associated.
sales_activity_type_ids	array of numbers	Identifiers of all sales activity types configured for a user.

## domainName

Returns the domain name of the business account that uses Freshworks CRM (or Freshsales) and product context of the product for which the app is installed. When building an omni app (app that is compatible with Freshworks CRM and Freshsales), you can use the product context to implement product-specific tasks.

```
client.data.get("domainName").then (
  function(data) {
    // success operation
  },
  function(error) {
    // failure operation
  }
);
```

## Sample Payload

```
{
  domainName: "sample.myfreshworks.com",
  productContext: {
    url: "https://sample.myfreshworks.com/crm",
    name: "freshworks_crm"
  }
}
```

## Attributes of the **payload**

ATTRIBUTE NAME	DATE TYPE	DESCRIPTION
domainName	string	<p>Domain name assigned by Freshworks, for a business account.</p> <p>If the app is deployed on an FCRM account that is migrated from Freshsales, the domainName is <code>&lt;sub-domain&gt;.freshworks.com</code>.</p> <p>For an omni app that is deployed on a Freshsales account, the domainName is <code>&lt;sub-domain&gt;.freshsales.io</code>.</p>
productContext	object	<p>Product context of the product for which the app is installed</p> <p><b>Attributes:</b></p> <p><b>url (string):</b> Complete web-address (protocol, sub-domain name, domain name, and relative path) of the product instance. <b>Possible values:</b></p> <p><b><code>https://&lt;sub-domain&gt;.myfreshworks.com/crm</code>,</b></p> <p><b><code>https://&lt;sub-domain&gt;.freshworks.com/crm</code>,</b></p> <p><b><code>https://&lt;sub-domain&gt;.freshsales.io</code></b></p> <p><b>name (string):</b> Name of the product. <b>Possible values:</b></p> <p><b><code>freshworks_crm</code>, <code>freshsales</code></b></p>

[Contact Details page](#)

[Sales Account Details page](#)

[Deals Details page](#)

Events Methods

[Contact Details page](#)

[Sales Account Details page](#)

[Deals Details page](#)

## Interface Methods

[Show Modal](#)

[Show Confirm](#)

[Show Notifications](#)

[Open Create Contact](#)

[Open Create Account](#)

[Open Create Deal](#)

[Open Create Call Log](#)

[Show Contact](#)

[Show Account](#)

[Show Deal](#)

# ADVANCED FEATURES

## [Data Storage](#)

## [OAuth](#)

## Request Method

You should use the Request method to avoid exposing sensitive information, such as API keys or user credentials, when you make HTTP requests to third-party domains as users will be able to inspect these values in a browser and access them. The other benefit is Cross-Origin Resource Sharing (CORS) support as requests are routed through a proxy.

### Notes:

1. All apps that make requests to third-party domains must use the Request method or they will not be published in the Freshworks Marketplace.
2. Requests made using the Request method should return String, JSON, or XML data type.
3. The timeout for requests is six seconds.
4. There is a rate limit of 50 requests per minute per app per account.

## Usage

The Request method should be defined in the **app.js** file by using the following syntax.

```
client.request.get("URL", options)
    .then(
        function(data) {
            //handle "data"
```



```

        // "data" is a json string with status, headers, and
        response.
    },
    function(error) {
        // handle failure
    }
};

```

The Request method should be defined in the **server.js** file by using the following syntax.

```

$request.get("URL", options)
    .then(
        function(data) {
            // handle "data"
            // "data" is a json string with status, headers, and
            response.
        },
        function(error) {
            // handle failure
        }
    );

```

## Parameters

PARAMETER	DESCRIPTION
URL	Is mandatory to make the Request method call.
options	Is a JSON object and can include the following properties: <ul style="list-style-type: none"> <li>● <b>headers</b>: HTTP Headers</li> <li>● <b>body</b>: Entity body, applicable for PATCH, POST, and PUT requests. Should be a string.</li> <li>● <b>staticIP</b>: A boolean value used to make requests with a static outgoing <a href="#">IP</a>.</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>maxAttempts:</b> The maximum number of times a request will try to retrieve a response if a <b>network</b> or <b>429/5xx HTTP</b> error occurs. The maximum permissible value for this argument is 5.</li> <li>• <b>retryDelay:</b> The delay between retry requests, specified in milliseconds. The maximum permissible value for this argument is 1500 ms.</li> </ul>
data	Data received from the Request method if the request is successful.
error	Message received from the request if an error is encountered.

## Using Iparams

Iparams can be used in requests as part of the following:

- [Header](#)
- **URL** - `https://passport-office.com/user?id=<%= iparam.passport %>`
- **Body**

```
client.request.post("https://helloworld.myfreshworks.com/api/contacts", {
  headers: {
    Authorization: "Basic <%= encode(iparam.username + ':' +
iparam.password) %>"
  },
  body: JSON.stringify({
    status: 2,
    priority: 1,
    description: "Test",
    subject: "<%= iparam.subjectPrefix %> Hello there",
    email: "tom@outerspace.com"
  })
});
```

## Sample Requests

For API requests that require authentication, you may need to encode the auth header in base64 format.

- GET request with authentication

```
var headers = {"Authorization": "Token token=<%=  
iparam.api_key %>"};  
var options = { headers: headers };  
var url =  
"https://sample.myfreshworks.com/itil/requesters/5.json";  
client.request.get(url, options)  
.then (  
function(data) {  
    console.log(data);  
},  
function(error) {  
    console.log(error);  
});
```

If the request is successful, **data** is returned in the following format. In case a **network** or **429/5xx HTTP** error occurs, the app will retry the request a number of times to obtain the response. Here, **attempts** is the number of times after which the request has returned a response.

```
{  
    "status" : 200,  
    "headers":  
    {  
        "Content-Length": 20,  
        "Content-Type": "application/json;charset=utf-8"  
    },  
    "response": "{ \"Name\": \"Rachel\" }",  
    "attempts": 1  
}
```

If the request fails due to an error from the platform, an error response is returned in the following format.

```
{
  "status" : 401,
  "headers":
  {
    "Content-Type": "application/json;charset=utf-8"
  },
  "response": [{"message": "Session expired or invalid",
  "attempts": 1,
  "errorCode": "INVALID_SESSION_ID"}]
}
```

If the request fails due to an error from the app, the **error** response is returned in the following format.

```
{
  "status" : 400,
  "headers":
  {
    "Content-Type": "application/json;charset=utf-8"
  },
  "response": "This domain has not been whitelisted.",
  "attempts": 2,
  "errorSource": "APP"
}
```

- GET request without authentication

```
var options = {};
var url = "https://httpbin.org/get?arg1=hello_world";
client.request.get(url, options)
.then (
function(data) {
  console.log(data);
},

```

```
function(error) {  
    console.log(error);  
});
```

- POST request with authentication

```
var headers = {"Authorization": "Token token=<%=  
iparam.api_key %>"};  
  
var options = { headers: headers, body: "Hello world"};  
var url = "https://sample.myfreshworks.com/api/contacts";  
client.request.post(url, options)  
    .then (  
    function(data) {  
        console.log(data);  
    },  
    function(error) {  
        console.log(error);  
    })
```

## OAuth Request

For information on how to make an OAuth request, see [Usage](#) in the OAuth section.

## Testing

For information on how to test an app that uses the Request API feature, see [Test the App](#).

## IP Ranges

Here is the list of IPs you must whitelist when using Request APIs if [IP whitelisting](#) is enabled on the Freshworks CRM support portal or if you wish to whitelist requests from your app.

REGION	IPS
United States	<b>18.233.117.211</b> and <b>35.168.222.30</b>
Europe-Central	<b>18.197.138.225</b> and <b>52.57.69.21</b>
India	<b>13.232.159.149</b> and <b>13.233.170.242</b>
Australia	<b>13.211.182.225</b> and <b>52.63.187.64</b>

## Errors

In case of request failure, a status code is displayed along with a message to help troubleshoot the issue.

An example of a status code and message is as follows.

```
{
  "status": 401,
  "headers": {
    "Content-Type": "text/plain"
  },
  "response": "Session expired or invalid",
  "errorSource": "APP/PLATFORM"
}
```

The following table lists all the supported status code.

STATUS CODE	DESCRIPTION
400	Bad request. For example, if you are making a request to a domain that has not been whitelisted, you will receive this status.
401	Unauthorized request.
429	Too many requests.

500	Internal server request. If this error persists, contact us at marketplace@freshworks.com.
502	Network error. If this error persists, contact us at marketplace@freshworks.com.
503	Service is temporarily unavailable.
504	Timeout error while processing the request.

## Instance Method

A single app can be present in multiple locations on the same page. In addition, a location may open up one or more modals. The locations and modals can be thought of as separate instances of the app and communicate with each other. The app framework provides instance methods to enable these use cases.

### Communication Between Instances

- [Send data from a parent location to a modal](#) - When a modal is opened, the location can send data to pre-populate form fields.
- [Send data from a modal to a parent location](#) - When a user fills a form in a modal window, some of this data may need to be returned to the parent.
- [Send data from one location to another location](#) - In the case of an app that is present in two locations on the same page, a button click in one location can be used to trigger an action in the second location.

The above use cases can be achieved with the following methods.

- [Context](#)
- [Get](#)
- [Send](#)
- [Receive](#)

## Context

This method is used to fetch information about the current instance. For example, modals contain information about the parent location which triggered them and any data that was passed to them.

### parent\_location\_template.html

```
client.instance.context().then(  
  function(context) {  
    console.log(context);  
  }  
);
```

### Output

```
{  
  instanceId: "4",  
  location: "contact_entity_menu"  
}
```

ATTRIBUTE	DESCRIPTION
instanceId	Each instance is auto-assigned with an ID.
location	The location of the current instance

### modal.html

```
client.instance.context().then(  
  function(context) {  
    console.log(context);  
  }  
);
```

### Output



```

{
  instanceId: "6",
  location: "overlay",
  parentId: "4"
}

```

ATTRIBUTE	DESCRIPTION
instanceId	Each instance is auto-assigned with an ID.
location	The location of the current instance.
parentId	ID of the parent instance that triggered the modal.

To know the usage of the context method with an example, see [Send data from a parent location to a modal](#).

## Get

This method is used to fetch context information of all instances that are active at the time of the `get()` call. It can be used in conjunction with the [send\(\)](#) method to send data to a specific instance.

### modal.html

```

client.instance.get().then(
  function(data) {
    console.log(data);
  }
);

```

### Output

```
[
```

```

        {instanceId: "4", location: "contact_entity_menu"},

        {instanceId: "3", location: "deal_entity_menu"},

        {instanceId: "5", location: "overlay"}

    ]

```

ATTRIBUTE	DESCRIPTION
instanceId	Each instance is auto-assigned with an ID.
location	The location of the current instance.

## Send

This method can be used to send data to one or many instances. The [receive\(\)](#) method should be used in the destination instance(s) to receive data. The syntax varies according to the following scenario.

- From one location/modal to one or more locations.

### modal.html

```

client.instance.send({

    message: {name: "James",

               email: "James@freshworks.com"},

    receiver: ["instanceID1", "instanceID2"]

});

```

ATTRIBUTE	DESCRIPTION
message	Data that is sent from location/modal. <b>message</b> can be a string, object, or array.

receiver	InstanceId(s) of the receiver locations.
----------	--

- From a modal to its parent location.

#### modal.html

```
client.instance.send({
    message: {name: "James",
              email: "James@freshworks.com"}
});
```

ATTRIBUTE	DESCRIPTION
message	Data that is sent from location/modal. <b>message</b> can be a string, object, or array.

The destination instances must be active when data is sent. You can retrieve all active instances using the [get\(\)](#) method.

## Receive

This method is used to receive data that is sent by another instance of the same app using the [send\(\)](#) method. Anytime data is sent to the instance, the callback method is executed.

#### modal.html

```
client.instance.receive(
    function(event) {
        var data = event.helper.getData();
        console.log(data);
        /* data contains {senderId: "1", message: { message: {name:
"James", email: "James@freshworks.com"}}} */
    }
);
```

) ;

**data** returns the following attributes.

ATTRIBUTE	DESCRIPTION
senderId	InstanceId(s) of the sender location/modal.
message	Data that is sent from a location/modal. <b>message</b> can be a string, object, or array.

## [Use Cases](#)

# SERVERLESS

## Overview

The Freshworks app development platform includes a serverless environment that enables you to build serverless (back-end) apps. Front-end apps run in response to events such as button clicks or page loads that happen on the Freshworks CRM product's UI (client). You can also create serverless apps that run in response to events such as app installation and uninstallation, events such as call establishment that occur in the Freshworks CRM product, or events that occur in an external product or service. To do this, you can configure event listeners that invoke callback methods. Serverless computing does involve servers, but they are abstracted away from developers. When a configured event occurs, the app logic in the callback method is run on a server, with the help of the event-specific payload passed to the callback method.

You can use the Server Method Invocation (SMI) feature, to build front-end apps that invoke a serverless component.

### Notes:

- The serverless apps are run in the sandbox mode. Therefore, usage of methods such as `process`, `buffer`, `clearImmediate`, `clearInterval`, `clearTimeout`, `setImmediate`, `setInterval`, and `setTimeout` in the app logic is not supported.
- The app execution timeout is 20 seconds.

## Create a Serverless App

After you install [NVM](#), [Node](#), and the [Freshworks CLI](#), to create a serverless app:

1. From the command line, navigate to the empty directory in which you want to create the app.
2. Run the following command: `$ fdk create --products freshworks_crm --template your_first_serverless_app`

The following app related files are created, based on the **your\_first\_serverless\_app** template:

```
|— README.md
|— config
|   |— iparam_test_data.json
|   |— iparams.json
|— manifest.json
|— server
|   |— lib
|   |   |— handle-response.js
|   |— server.js
|   |— test_data
|       |— onAppInstall.json
|       |— onAppUninstall.json
|       |— onAppointmentCreate.json
|       |— onAppointmentUpdate.json
|       |— onContactCreate.json
|       |— onContactUpdate.json
|       |— onCustomModuleCreate.json
|       |— onCustomModuleUpdate.json
|       |— onDealCreate.json
|       |— onDealUpdate.json
|       |— onExternalEvent.json
|       |— onSalesAccountCreate.json
```

- |— `onSalesAccountUpdate.json`
- |— `onScheduledEvent.json`
- |— `onTaskCreate.json`
- |— `onTaskUpdate.json`

4 directories, 22 files

DIRECTORY/FILE	DESCRIPTION
README.md	Contains all information, code-related specifications, and additional instructions pertaining to the app.
server/	Contains all the directories and files that define the serverless app and are used to test it. The JS file follows the ES5 standard.
server/lib	Contains the libraries, with helper methods, that can be used in the <b>server.js</b> file.
server/server.js	Contains the events registered and their corresponding callback methods. The app logic resides in the callback methods' definitions. The event registration and callback methods are placed in the exports code block, to ensure that the code is imported and run by the server.
server/test_data	Contains sample payloads for all the events configured in the server.js file. The sample payloads are used in the local testing of apps. You can modify the content of the JSON files in this directory. Ensure not to modify the names of the JSON files.
config/	Contains the installation parameters related file.
config/iparams_test_data.json	This file is deprecated. Before testing an app, navigate to <a href="https://localhost:10001/custom_configs">https://localhost:10001/custom_configs</a> and enter appropriate values for the configured installation parameters.
config/iparams.json	Contains all the installation parameters whose values are set when

	the app is installed. For more information, see <a href="#">Installation Parameters</a> .
manifest.json	<p>Contains details such as the platform version the app uses, product to which the app belongs, and npm packages that the app uses (dependencies).</p> <pre> {   "platform-version": "2.1",   "product": {     "freshworks_crm": {       "events": {         "&lt;eventName&gt;": {           "handler": "&lt;eventCallbackMethod&gt;"         },         "onExternalEvent": {           "handler": "onExternalEvent"         }       }     },     "freshsales": {       "events": {         "&lt;eventName&gt;": {           "handler": "&lt;eventCallbackMethod&gt;"         },         "onExternalEvent": {           "handler": "onExternalEvent"         }       }     }   } } </pre> <p>For product events, &lt;eventName&gt; is the appropriate product event name. For app setup events, &lt;eventName&gt; is onAppInstall or onAppUninstall. For external events, &lt;eventName&gt; is</p>



	onExternalEvent. For scheduled events, <eventName> is onScheduledEvent.
--	---

## Dependencies

## Test the App

### Notes:

- To test a serverless app, use the latest version of Chrome.
- As testing is only a simulation of events, actual data or record associated with the event is not created in the back-end. To create actual data and test the events, publish your app as a custom app and test the events manually.
- Ensure that the JSON files that contain the sample payloads to test the events are available at **<app's root directory>/server/test\_data**.

To simulate an event and test your app,

1. From the command line, navigate to the directory that contains the app related files and run the following command: **\$ fdk run**
2. In the address bar of the browser, enter **https://localhost:10001/web/test**. A dialog box is displayed.
3. For omni apps, a **Select Product** drop-down box is displayed. Select the product for which you want to test the serverless events.
4. Click **Select an event**. The list of all events configured in the **server.js** file is displayed.  
**Note:** To test app setup events, select **onAppInstall** or **onAppUninstall**. To test external events, select **onExternalEvent**. To test scheduled events, select **onScheduledEvent**. To test a product event, select the corresponding product event's name. Select an event. The corresponding payload is displayed. To test a different scenario other than the default, edit the payload.
5. Click **Simulate**. If the event is successfully simulated, the **Simulate** button changes to a **Success** button. If the event simulation fails because of invalid payload data, the

**Simulate** button changes to a **Failed** button. Modify the payload appropriately and click **Simulate**.

## [App Setup Events](#)

## [External Events](#)

## Product Events

Events that occur in the Freshworks CRM product, such as creating a contact, updating a deal, creating a task, and so on are termed as product events. You can enable product events to trigger apps. To do this, configure event listeners in the **server.js** file. When a product event occurs, the corresponding event listener invokes a callback method. The app logic in the callback method runs with the help of the event-specific payload passed to the callback method. For omni apps, the event listeners must be configured in the **manifest.json** file.

## Configure Events

To register a product event and the corresponding callback:

1. From your app's root directory, navigate to the **server.js** file.
2. In the **exports** block, enter the event, corresponding callback, and callback function definition as follows.

```
exports = {  
  events: [  
    //This is the list of all product events and the  
    corresponding callback methods  
    { event: "eventName", callback:"eventCallbackMethod"}  
  ],  
  eventCallbackMethod: function(payload) {  
    //This is the callback function definition
```

```

        console.log("Logging arguments from the event:" +
JSON.stringify(payload));
    }
}

```

**Note:** Ensure to enter only one callback method for an event. Multiple events can access the same callback method.

For **omni** apps, to register a product event and the corresponding callback:

1. From your app's root directory, navigate to the manifest.json file.
2. In the **events** object, enter the appropriate product event name as follows:

```

"events": {
    "<eventName>": {
        "handler": "<eventCallbackMethod>"
    }
}

```

**Note:** Events of the different products can access the same callback method. Ensure to register the product event name in both **freshsales** and **freshworks\_crm**.

3. Navigate to the **server.js** file.
4. In the **exports** block, enter the callback function definition as follows:

```

exports = {
    // args is a JSON block containing the payload information.
    // args['iparam'] will contain the installation parameter
    values.
    //eventCallbackMethod is the call-back function name specified in
    manifest.json
    eventCallbackMethod: function(args) {
        console.log('Hello ' + args['data']['lead']['email']);
    }
};

```

## Payload Attributes

When a product event occurs, an event-specific payload is passed to the callback method.

```

{
    "account_id" : "value",
    "domain" : "value",

```

```

"event" : "value",
"region" : "value",
"timestamp" : "value",
"productContext": {
  "url": "https://sample.myfreshworks.com/crm",
  "name": "freshworks_crm"
},
"data" : {
  //Contains the list of objects related to the event.
},
"iparams" : {
  "Param1" : "value",
  "Param2" : "value"
}
}

```

The payload is a JSON object with the following attributes.

ATTRIBUTE	DATA TYPE	DESCRIPTION
account_id	number	Identifier of the Freshworks CRM account, auto-generated when the account is configured for a business.
domain	string	Domain name for the Freshworks CRM account.
event	string	Identifier of the product event.
region	string	Region where the app is installed. Possible values: US, EU, EUC, AUS, and IND.
timestamp	number	Timestamp of when the product event occurs, specified in the epoch format.

productContext	object	<p>Product context of the product for which the app is installed.</p> <p><b>Attributes:</b></p> <p><b>url (string):</b> Complete web-address (protocol, sub-domain name, domain name, and relative path) of the product instance. <b>Possible values:</b></p> <p><b>https://&lt;sub-domain&gt;.myfreshworks.com/crm,</b>  <b>https://&lt;sub-domain&gt;.freshworks.com/crm ,</b>  <b>https://&lt;sub-domain&gt;.freshsales.io</b></p> <p><b>name (string):</b> Name of the product. <b>Possible values: freshworks_crm, freshsales</b></p>
iparams	object	<p>Installation parameters specified as a JSON object of &lt;parameter name&gt;: &lt;parameter value&gt; pairs.</p>
data	object	<p>Event-specific Freshworks CRM data, specified as a JSON object of key:value pairs.</p>

## onContactCreate

When a contact is created, the **onContactCreate** event is triggered. To register the **onContactCreate** event and the corresponding callback, use the following sample **server.js** content.

```
exports = {
  events: [
    { event: "onContactCreate", callback:
      "onContactCreateCallback" }
  ]
}
```

```

],
onContactCreateCallback: function(payload) {
    console.log("Logging arguments from onContactCreate
event: " + JSON.stringify(payload));
}
}

```

### Sample Payload

```

{
  "timestamp": 1593015490.2012472,
  "account_id": "1967183444532229946",
  "domain": "sample.myfreshworks.com",
  "event": "onContactCreate",
  "region": "US",
  "productContext": {
    "url": "https://sample.myfreshworks.com/crm",
    "name": "freshworks_crm"
  },
  "data": {
    "actor": {
      "id": 1437,
      "name": "John Doe",
      "email": "sample@xyz.com",
      "job_title": null,
      "work_number": null,
      "mobile_number": null,
      "language": "en",
      "time_zone": "UTC",
      "is_active": true,
      "deal_pipeline_id": 154,
      "is_forgotten": false,
      "created_at": "2020-06-09T12:21:48Z",
      "updated_at": "2020-06-09T12:21:48Z",

```

```
"uuid": "190440313184929466",
"type": "user"
},
"contact": {
  "first_name": {
    "label": "First name",
    "type": "text",
    "value": "fff"
  },
  "last_name": {
    "label": "Last name",
    "type": "text",
    "value": null
  },
  "owner_id": {
    "label": "Sales owner",
    "type": "dropdown",
    "value": 1437
  },
  "recent_note": {
    "label": "Recent note",
    "type": "paragraph",
    "value": null
  },
  "tags": {
    "label": "Tags",
    "type": "auto_complete",
    "value": []
  },
  "job_title": {
    "label": "Job title",
    "type": "text",
    "value": null
  }
}
```

```
},
"emails": {
  "label": "Emails",
  "type": "group_field",
  "value": []
},
"work_number": {
  "label": "Work",
  "type": "phone_number",
  "value": null
},
"mobile_number": {
  "label": "Mobile",
  "type": "phone_number",
  "value": "9876543223456789"
},
"contact_status_id": {
  "label": "Status",
  "type": "dropdown",
  "value": null
},
"country": {
  "label": "Country",
  "type": "text",
  "value": null
},
"time_zone": {
  "label": "Time zone",
  "type": "dropdown",
  "value": null
},
"address": {
  "label": "Address",
```



```
    "type": "text",
    "value": null
  },
  "city": {
    "label": "City",
    "type": "text",
    "value": null
  },
  "state": {
    "label": "State",
    "type": "text",
    "value": null
  },
  "zipcode": {
    "label": "Zipcode",
    "type": "text",
    "value": null
  },
  "do_not_disturb": false,
  "linkedin": {
    "label": "LinkedIn",
    "type": "text",
    "value": null
  },
  "territory_id": {
    "label": "Territory",
    "type": "dropdown",
    "value": null
  },
  "lead_source_id": {
    "label": "Source",
    "type": "dropdown",
    "value": null
  }
}
```

```
,
"facebook": {
  "label": "Facebook",
  "type": "text",
  "value": null
},
"twitter": {
  "label": "Twitter",
  "type": "text",
  "value": null
},
"campaign_id": {
  "label": "Campaign",
  "type": "dropdown",
  "value": null
},
"medium": {
  "label": "Medium",
  "type": "text",
  "value": null
},
"keyword": {
  "label": "Keyword",
  "type": "text",
  "value": null
},
"last_contacted": {
  "label": "Last contacted time",
  "type": "date_time",
  "value": null
},
"web_form_ids": {
  "label": "Web forms",
```

```
    "type": "multi_select_dropdown",
    "value": []
  },
  "last_contacted_mode": {
    "label": "Last contacted mode",
    "type": "dropdown",
    "value": null
  },
  "created_at": {
    "label": "Created at",
    "type": "date_time",
    "value": "2020-06-09T15:21:39Z"
  },
  "last_contacted_sales_activity_mode": {
    "label": "Last activity type",
    "type": "dropdown",
    "value": null
  },
  "active_sales_sequences": {
    "label": "Active sales sequences",
    "type": "multi_select_dropdown",
    "value": []
  },
  "last_contacted_via_sales_activity": {
    "label": "Last activity date",
    "type": "date_time",
    "value": null
  },
  "lead_score": {
    "label": "Score",
    "type": "number",
    "value": 0
  },
}
```

```
"completed_sales_sequences": {
  "label": "Completed sales sequences",
  "type": "multi_select_dropdown",
  "value": []
},
"last_seen": {
  "label": "Last seen",
  "type": "date_time",
  "value": null
},
"updater_id": {
  "label": "Updated by",
  "type": "dropdown",
  "value": 1437
},
"updated_at": {
  "label": "Updated at",
  "type": "date_time",
  "value": "2020-06-09T15:21:39Z"
},
"last_assigned_at": {
  "label": "Last assigned at",
  "type": "date_time",
  "value": "2020-06-09T15:21:40Z"
},
"external_id": {
  "label": "External ID",
  "type": "text",
  "value": null
},
"work_email": {
  "label": "Work email",
  "type": "email",
```

```
    "value": null
  },
  "subscription_status": {
    "label": "Subscription status",
    "type": "dropdown",
    "value": 1
  },
  "subscription_types": {
    "label": "Subscription types",
    "type": "multi_select_dropdown",
    "value": [
      {
        "id": "1",
        "value": "Non-marketing emails from our company"
      }
    ]
  },
  "id": 5510,
  "is_deleted": false,
  "open_deals_amount": {
    "label": "Open deals amount",
    "dom_type": "decimal",
    "value": "0.0"
  },
  "name": {
    "label": "Name",
    "dom_type": "text",
    "value": "fff"
  },
  "merged_to": 0,
  "status": "Active",
  "first_assigned_at": {
    "label": "First assigned at",
```

```

    "dom_type": "date_time",
    "value": "2020-06-09T15:21:40Z"
  },
  "first_contacted": {
    "label": "First contacted",
    "dom_type": "date_time",
    "value": null
  },
  "won_deals_amount": {
    "label": "Won deals amount",
    "dom_type": "decimal",
    "value": "0.0"
  },
  "import_csv_id": {
    "label": "Import label",
    "dom_type": "multi_select_dropdown",
    "value": []
  },
  "sales_account_ids": [],
  "import_id": null,
  "avatar": null,
  "team_user_ids": [],
  "email_status": {
    "label": "Email status",
    "dom_type": "text",
    "value": null
  },
  "mcr_id": 1270375514142703600,
  "list_ids": [],
  "creator_id": {
    "label": "Created by",
    "type": "dropdown",
    "value": 1437
  }

```

```

    },
    "custom_fields": [],
    "source_additional_info": {}
  },
  "associations": {
    "owner": {
      "id": 1437,
      "name": "John Doe",
      "type": "user",
      "email": "sample@xyz.com",
      "is_active": true,
      "work_number": null,
      "mobile_number": null,
      "time_zone": "UTC",
      "created_at": "2020-06-09T12:21:48Z",
      "updated_at": "2020-06-09T12:21:48Z",
      "deal_pipeline_id": 154,
      "job_title": null,
      "uuid": "190440313184929466"
    },
    "source": null,
    "contact_status": null,
    "campaign": null,
    "territory": null,
    "updater": {
      "id": 1437,
      "name": "John Doe",
      "type": "user",
      "email": "sample@xyz.com",
      "is_active": true,
      "work_number": null,
      "mobile_number": null,
      "time_zone": "UTC",

```

```

        "created_at": "2020-06-09T12:21:48Z",
        "updated_at": "2020-06-09T12:21:48Z",
        "deal_pipeline_id": 154,
        "job_title": null,
        "uuid": "190440313184929466"
    },
    "team_users": [],
    "sales_accounts": [],
    "lists": [],
    "creator": {
        "id": 1437,
        "name": "John Doe",
        "type": "user",
        "email": "sample@xyz.com",
        "is_active": true,
        "work_number": null,
        "mobile_number": null,
        "time_zone": "UTC",
        "created_at": "2020-06-09T12:21:48Z",
        "updated_at": "2020-06-09T12:21:48Z",
        "deal_pipeline_id": 154,
        "job_title": null,
        "uuid": "190440313184929466"
    }
}
}
}

```

Attributes of the **data** object

ATTRIBUTE	DATA TYPE	DESCRIPTION
actor	<a href="#">actor object</a>	Details of the entity that triggered the onContactCreate event.



contact	<a href="#">contact object</a>	Details of the contact object created when the onContactCreate event is triggered.
associations	<a href="#">object</a>	Associated objects related to the contact object.

#### Attributes of the **actor** object

ATTRIBUTE	DATA TYPE	DESCRIPTION
id	number	Identifier associated with the actor.
name	string	Name of the actor.
email	string	Email address of the actor.
job_title	string	Designation of the actor.
work_number	number	Official phone number of the actor.
mobile_number	number	Phone number of the actor.
language	string	Language of the actor; by default, language is 'en'.
time_zone	string	Time zone of the actor, specified in the UTC format.
is_active	boolean	Set to 'true' if the actor is verified.
deal_pipeline_id	number	ID of deal pipelines that this actor is part of.
is_forgotten	boolean	If this attribute is 'true', specified data is permanently deleted.
created_at	string	Creation timestamp specified in the UTC format.
updated_at	string	Updation timestamp, specified in the UTC format. For onContactCreate, this value is the same as created_at.
uuid	string	Unique user identifier, this value is auto-generated.

type	string	Helps categorize a user.
------	--------	--------------------------

#### Attributes of the **contact** object

ATTRIBUTE	DATA TYPE	DESCRIPTION
open_deals_amount	object	Sum of all the open deals associated with the contact.
source_additional_info	object	Meta information specified as a custom JSON object.
first_name	object	First name of the contact.
last_name	object	Last name of the contact.
city	object	City where the contact resides.
zipcode	object	Postal code of the region where the contact resides.
name	object	First and last name of the contact.
custom_fields	array of strings	Key value pairs containing the names and values of custom fields.
first_assigned_at	object	First assigned timestamp.
avatar	string	Avatar URL.
last_contacted_mode	object	Last mode of contact. <b>Possible values:</b> <b>Email Incoming, Email Outgoing, Call Incoming, Call Outgoing, Email Opened, Chat, Sales Activity.</b>

state	object	State where the contact resides.
lead_source_id	number	Identifier of the source of the deal.
job_title	object	Job title of the contact.
tags	object	Tags associated with this contact.
linkedin	object	Linkedin address of the contact.
import_csv_id	object	Reference of all imports that this contact was part of.
creator_id	object	Identifier of the user who created the contact.
won_deals_amount	object	Sum of all won deals associated with the contact.
active_sales_sequences	object	Active sales campaigns that this contact was part of.
owner_id	object	User who owns the record.
contact_status_id	object	Identifier of the contact status.
is_deleted	boolean	Specifies whether a contact is deleted. This value is set to 'true' if the contact is deleted.
last_assigned_at	object	Last assigned timestamp.
country	object	Country of the contact.
medium	object	Medium tracked in UTM.
keyword	object	Keyword tracked in UTM.
id	number	Identifier of the contact.
territory_id	object	Identifier of the territory.
team_user_ids	array	Identifiers of the collaborators associated with

		the contact.
facebook	object	Facebook ID.
last_contacted	object	Last contacted timestamp.
work_email	object	Official email address of the contact.
last_contacted_via_sales_activity	object	Last contacted timestamp (through sales activity).
status	string	Status of the contact. <b>Possible values:</b> <b>active, merged, deleted</b>
mcr_id	number	Identifier of the master customer record to which the contact information belongs. This value is auto-generated.
external_id	object	Identifier of the external references from which the contact information is fetched.
address	object	Address of the contact.
created_at	object	Contact creation timestamp.
do_not_disturb	boolean	Set to 'true' if contact is not to be disturbed.
import_id	string	ID of the equivalent record in the external system from which it was imported.
sales_account_ids	array of numbers	IDs of the sales account associated with this contact.
twitter	object	Twitter handle of the contact.
recent_note	object	Recently added note against the contact.

merged_to	number	Reference of secondary contact merged to the primary contact.
emails	object	Primary and secondary emails associated with this contact.
updated_at	object	Updated timestamp.
work_number	object	Telephone number of the contact.
first_contacted	object	First contacted timestamp.
list_ids	array of numbers	Identifiers of the marketing lists associated with the contact.
subscription_types	object	Subscription types available. <b>Possible values:</b> <b>Promotional, Newsletter, Product updates, Conferences &amp; Events, Non-marketing emails from our company</b>
lead_score	object	Score to qualify a contact.
last_contacted_sales_activity_mode	string	Last mode of sales activity rendered to the contact.
completed_sales_sequences	object	Completed sales sequences that this contact was part of.
campaign_id	object	ID of the UTM campaign.
last_seen	object	Last seen timestamp.
time_zone	object	Time zone of the contact.
updater_id	object	ID of the user who updated the contact.
web_form_ids	object	Forms through which this record was captured.

mobile_number	object	Mobile number of the contact.
---------------	--------	-------------------------------

Attributes of the **associations** object

ATTRIBUTE	DATA TYPE	DESCRIPTION
lists	array	Marketing lists associated with the contact.
source	string	Source details of this contact.
campaign	string	UTM campaign.
creator	<a href="#">object</a>	Details of contact creator.
territory	string	Territory of the contact.
contact_status	string	Qualification status of the contact.
owner	<a href="#">object</a>	Details of the contact owner.
updater	<a href="#">object</a>	Details of the contact updater. For onContactCreate, this value is null.
team_users	array	Collaborators associated with the contact.
sales_accounts	array of numbers	Sales accounts of the contact.

Attributes of the **user** object. Here, the user can be the **owner**, **updater**, or **creator**.

ATTRIBUTE	DATA TYPE	DESCRIPTION
deal_pipeline_id	number	ID of deal pipelines that this user is part of.

name	string	Name of the user.
is_active	boolean	Set to 'true' if the user is verified.
email	string	Primary email of the user.
job_title	string	Job title of the user.
uuid	string	Unique user identifier, this value is auto-generated.
id	number	ID of the user.
created_at	string	Creation timestamp.
type	string	Helps categorize a user.
updated_at	string	Updated timestamp.
work_number	string	Phone number of the user.
time_zone	string	Time zone of the user.
mobile_number	string	Mobile number of the user.

## onContactUpdate

When a contact is updated, the **onContactUpdate** event is triggered. To register the **onContactUpdate** event and the corresponding callback, use the following sample **server.js** content.

```
exports = {
  events: [
    { event: "onContactUpdate", callback:
      "onContactUpdateCallback" }
  ],

```

```

onContactUpdateCallback: function(payload) {
    console.log("Logging arguments from onContactUpdate
event: " + JSON.stringify(payload));
}
}

```

## Sample Payload

```

{
  "iparams": {},
  "region": "US",
  "productContext": {
    "url": "https://sample.myfreshworks.com/crm",
    "name": "freshworks_crm"
  },
  "data": {
    "actor": {
      "deal_pipeline_id": 27149,
      "name": "Jim Doe",
      "is_active": true,
      "email": "jim.doe@xyz.com",
      "job_title": "Engineer",
      "uuid": "195818342547907164",
      "id": 39897,
      "language": "en",
      "created_at": "2020-06-24T08:32:11Z",
      "type": "user",
      "updated_at": "2020-06-24T08:34:20Z",
      "work_number": null,
      "time_zone": "UTC",
      "is_forgotten": false,
      "mobile_number": null
    },
    "contact": {

```



```
"open_deals_amount": {
  "label": "Open deals amount",
  "dom_type": "decimal",
  "value": "0.0"
},
"source_additional_info": {},
"first_name": {
  "label": "First name",
  "type": "text",
  "value": "Test"
},
"city": {
  "label": "City",
  "type": "text",
  "value": null
},
"zipcode": {
  "label": "Zipcode",
  "type": "text",
  "value": null
},
"name": {
  "label": "Name",
  "dom_type": "text",
  "value": "Test owner Update"
},
"custom_fields": [],
"subscription_status": {
  "label": "Subscription status",
  "type": "dropdown",
  "value": 0
},
"first_assigned_at": {
```

```
    "label": "First assigned at",
    "dom_type": "date_time",
    "value": "2020-06-24T18:23:28Z"
  },
  "avatar": null,
  "last_contacted_mode": {
    "label": "Last contacted mode",
    "type": "dropdown",
    "value": null
  },
  "state": {
    "label": "State",
    "type": "text",
    "value": null
  },
  "lead_source_id": {
    "label": "Source",
    "type": "dropdown",
    "value": null
  },
  "job_title": {
    "label": "Job title",
    "type": "text",
    "value": null
  },
  "tags": {
    "label": "Tags",
    "type": "auto_complete",
    "value": []
  },
  "linkedin": {
    "label": "LinkedIn",
    "type": "text",
```

```
    "value": null
  },
  "import_csv_id": {
    "label": "Import label",
    "dom_type": "multi_select_dropdown",
    "value": []
  },
  "creator_id": {
    "label": "Created by",
    "type": "dropdown",
    "value": 39897
  },
  "won_deals_amount": {
    "label": "Won deals amount",
    "dom_type": "decimal",
    "value": "0.0"
  },
  "active_sales_sequences": {
    "label": "Active sales sequences",
    "type": "multi_select_dropdown",
    "value": []
  },
  "owner_id": {
    "label": "Sales owner",
    "type": "dropdown",
    "value": 39897
  },
  "contact_status_id": {
    "label": "Status",
    "type": "dropdown",
    "value": null
  },
  "is_deleted": false,
```

```
"last_assigned_at": {
  "label": "Last assigned at",
  "type": "date_time",
  "value": "2020-06-24T18:23:28Z"
},
"country": {
  "label": "Country",
  "type": "text",
  "value": null
},
"medium": {
  "label": "Medium",
  "type": "text",
  "value": null
},
"keyword": {
  "label": "Keyword",
  "type": "text",
  "value": null
},
"team_user_ids": [],
"id": 1399554,
"territory_id": {
  "label": "Territory",
  "type": "dropdown",
  "value": null
},
"email_status": {
  "label": "Email status",
  "dom_type": "text",
  "value": null
},
"facebook": {
```

```
    "label": "Facebook",
    "type": "text",
    "value": null
  },
  "last_contacted": {
    "label": "Last contacted time",
    "type": "date_time",
    "value": null
  },
  "last_name": {
    "label": "Last name",
    "type": "text",
    "value": "owner Update"
  },
  "work_email": {
    "label": "Work email",
    "type": "email",
    "value": null
  },
  "last_contacted_via_sales_activity": {
    "label": "Last activity date",
    "type": "date_time",
    "value": null
  },
  "status": "Active",
  "mcr_id": 1275857082467778600,
  "phone_numbers": {
    "label": "Other phone numbers",
    "type": "group_field",
    "value": []
  },
  "external_id": {
    "label": "External ID",
```

```
    "type": "text",
    "value": null
  },
  "address": {
    "label": "Address",
    "type": "text",
    "value": null
  },
  "created_at": {
    "label": "Created at",
    "type": "date_time",
    "value": "2020-06-24T18:23:27Z"
  },
  "do_not_disturb": true,
  "import_id": null,
  "sales_account_ids": [],
  "twitter": {
    "label": "Twitter",
    "type": "text",
    "value": null
  },
  "recent_note": {
    "label": "Recent note",
    "type": "paragraph",
    "value": null
  },
  "merged_to": 0,
  "emails": {
    "label": "Emails",
    "type": "group_field",
    "value": []
  },
  "updated_at": {
```

```
    "label": "Updated at",
    "type": "date_time",
    "value": "2020-06-24T18:27:01Z"
  },
  "work_number": {
    "label": "Work",
    "type": "phone_number",
    "value": null
  },
  "first_contacted": {
    "label": "First contacted",
    "dom_type": "date_time",
    "value": null
  },
  "list_ids": [],
  "last_contacted_sales_activity_mode": {
    "label": "Last activity type",
    "type": "dropdown",
    "value": null
  },
  "subscription_types": {
    "label": "Subscription types",
    "type": "multi_select_dropdown",
    "value": []
  },
  "lead_score": {
    "label": "Score",
    "type": "number",
    "value": 0
  },
  "completed_sales_sequences": {
    "label": "Completed sales sequences",
    "type": "multi_select_dropdown",
```

```

        "value": []
    },
    "campaign_id": {
        "label": "Campaign",
        "type": "dropdown",
        "value": null
    },
    "last_seen": {
        "label": "Last seen",
        "type": "date_time",
        "value": null
    },
    "time_zone": {
        "label": "Time zone",
        "type": "dropdown",
        "value": null
    },
    "updater_id": {
        "label": "Updated by",
        "type": "dropdown",
        "value": 39897
    },
    "web_form_ids": {
        "label": "Web forms",
        "type": "multi_select_dropdown",
        "value": []
    },
    "mobile_number": {
        "label": "Mobile",
        "type": "phone_number",
        "value": "1231231230"
    }
},

```



```
"associations": {
  "lists": [],
  "source": null,
  "campaign": null,
  "creator": {
    "deal_pipeline_id": 27149,
    "name": "Jim Doe",
    "is_active": true,
    "email": "jim.doe@freshworks.com",
    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "created_at": "2020-06-24T08:32:11Z",
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
    "mobile_number": null
  },
  "territory": null,
  "contact_status": null,
  "owner": {
    "deal_pipeline_id": 27149,
    "name": "Jim Doe",
    "is_active": true,
    "email": "jim.doe@xyz.com",
    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "created_at": "2020-06-24T08:32:11Z",
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
  }
}
```

```

    "time_zone": "UTC",
    "mobile_number": null
  },
  "updater": {
    "deal_pipeline_id": 27149,
    "name": "Jim Doe",
    "is_active": true,
    "email": "jim.doe@xyz.com",
    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "created_at": "2020-06-24T08:32:11Z",
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
    "mobile_number": null
  },
  "team_users": [],
  "sales_accounts": []
},
"changes": {
  "model_changes": {
    "last_name": [
      "owner",
      "owner Update"
    ],
    "created_at": [
      "2020-06-24T18:23:27Z",
      "2020-06-24T18:23:27Z"
    ],
    "updater_id": [
      null,

```

```

        39897
      ],
      "updated_at": [
        "2020-06-24T18:23:27Z",
        "2020-06-24T18:27:01Z"
      ]
    },
    "system_changes": {},
    "misc_changes": {}
  }
},
"account_id": "195818342531129946",
"domain": "sample.myfreshworks.com",
"event": "onContactUpdate",
"timestamp": 1593023221.9586976,
"version": "2.0"
}

```

The changes field is passed along with the payload. When a contact is deleted, the **is\_deleted** field is set to **'true'**.

Attributes of the **data** object

ATTRIBUTE	DATA TYPE	DESCRIPTION
actor	<a href="#">actor object</a>	Details of the entity that triggered the onContactUpdate event.
contact	<a href="#">contact object</a>	Details of the contact object.
associations	<a href="#">object</a>	Associated objects related to the contact object.
changes	object	Updated field values.

Attributes of the onContactUpdate **contact** object

ATTRIBUTE	DATA TYPE	DESCRIPTION
open_deals_amount	object	Sum of all the open deals associated with the contact.
source_additional_info	object	Meta information specified as a custom JSON object.
first_name	object	First name of the contact.
last_name	object	Last name of the contact.
city	object	City where the contact resides.
zipcode	object	Postal code of the region where the contact resides.
name	object	First and last name of the contact.
custom_fields	array of strings	Key value pairs containing the names and values of custom fields.
first_assigned_at	object	First assigned timestamp.
avatar	string	Avatar URL.
last_contacted_mode	object	Last mode of contact. <b>Possible values:</b> <b>Email Incoming, Email Outgoing, Call Incoming, Call Outgoing, Email Opened, Chat, Sales Activity.</b>
state	object	State where the contact resides.
lead_source_id	number	ID of the source of the deal.
job_title	object	Job title of the contact.

tags	object	Tags associated with this contact.
linkedin	object	Linkedin address of the contact.
import_csv_id	object	Reference of all imports that this contact was part of.
creator_id	object	ID of the user who created the contact.
won_deals_amount	object	Sum of all won deals associated with the contact.
active_sales_sequences	object	Active sales campaigns that this contact was part of.
owner_id	object	User who owns the record.
contact_status_id	object	ID of the contact status.
is_deleted	boolean	Set to 'true' if contact is deleted.
last_assigned_at	object	Last assigned timestamp.
country	object	Country of the contact.
medium	object	Medium tracked in UTM.
keyword	object	Keyword tracked in UTM.
id	number	ID of the contact.
territory_id	object	ID of the territory.
team_user_ids	array	Identifiers of the collaborators associated with the contact.
email_status	object	Status of the email.

		<b>Possible values:</b> <b>active, invalid, bounced</b>
facebook	object	Facebook ID.
last_contacted	object	Last contacted timestamp.
work_email	object	Official email address of the contact.
last_contacted_ia_sales_activity	object	Last contacted timestamp (through sales activity).
status	string	Status of the contact. <b>Possible values:</b> <b>active, merged, deleted</b>
mcr_id	number	Identifier of the master customer record. This value is auto-generated.
external_id	object	Identifier of external references of the contact.
address	object	Address of the contact.
created_at	object	Contact creation timestamp.
do_not_disturb	boolean	Set to 'true' if contact is not to be disturbed.
import_id	string	ID of the equivalent record in the external system from which it was imported.
sales_account_ids	array of numbers	IDs of the sales account associated with this contact.
twitter	object	Twitter handle of the contact.
recent_note	object	Recently added note against the contact.
merged_to	number	Reference of secondary contact merged to the

		primary contact.
emails	object	Primary and secondary emails associated with this contact.
updated_at	object	Updated timestamp.
work_number	object	Telephone number of the contact.
first_contacted	object	First contacted timestamp.
list_ids	array	Identifiers of the marketing lists associated with the contact.
subscription_types	object	Subscription types available. <b>Possible values:</b> <b>Promotional, Newsletter, Product updates, Conferences &amp; Events, Non-marketing emails from our company</b>
lead_score	object	Score to qualify a contact.
last_contacted_sales_activity_mode	string	Last mode of sales activity rendered to the contact.
completed_sales_sequences	object	Completed sales sequences that this contact was part of.
campaign_id	object	ID of the UTM campaign.
last_seen	object	Last seen timestamp.
time_zone	object	Time zone of the contact.
updater_id	object	ID of the user who updated the contact.
web_form_ids	object	Forms through which this record was captured.

mobile_number	object	Mobile number of the contact.
---------------	--------	-------------------------------

## onDealCreate

When a deal is created, the **onDealCreate** event is triggered. To register the onDealCreate event and the corresponding callback, use the following sample **server.js** content.

```
exports = {
  events: [
    { event: "onDealCreate", callback:
"onDealCreateCallback" },
  ],
  onDealCreateCallback: function(payload) {
    console.log("Logging arguments from onDealCreate event:
" + JSON.stringify(payload));
  }
}
```

### Sample payload

```
{
  "iparams": {},
  "region": "US",
  "productContext": {
    "url": "https://sample.myfreshworks.com/crm",
    "name": "freshworks_crm"
  },
  "data": {
```



```

"actor": {
  "deal_pipeline_id": 27149,
  "name": "Jim Doe",
  "is_active": true,
  "email": "jim.doe@xyz.com",
  "job_title": "Engineer",
  "uuid": "195818342547907164",
  "id": 39897,
  "language": "en",
  "created_at": "2020-06-24T08:32:11Z",
  "type": "user",
  "updated_at": "2020-06-24T08:34:20Z",
  "work_number": null,
  "time_zone": "UTC",
  "is_forgotten": false,
  "mobile_number": null
},
"deal": {
  "deal_pipeline_id": {
    "label": "Deal pipeline",
    "type": "dropdown",
    "value": 27149
  },
  "source_additional_info": {},
  "deal_product_id": {
    "label": "Product",
    "type": "dropdown",
    "value": null
  },
  "name": {
    "label": "Name",
    "type": "text",
    "value": "Deal with owner"
  }
}

```

```
,
"custom_fields": [],
"upcoming_activities_time": {
  "label": "Upcoming activities",
  "type": "date_time",
  "value": null
},
"deal_type_id": {
  "label": "Type",
  "type": "dropdown",
  "value": null
},
"first_assigned_at": {
  "label": "First assigned at",
  "dom_type": "date_time",
  "value": "2020-06-24T18:30:55Z"
},
"web_form_id": {
  "label": "Web form",
  "type": "dropdown",
  "value": null
},
"contact_ids": [],
"lead_source_id": {
  "label": "Source",
  "type": "dropdown",
  "value": null
},
"tags": {
  "label": "Tags",
  "type": "auto_complete",
  "value": []
},
```

```
"import_csv_id": {
  "label": "Import label",
  "dom_type": "multi_select_dropdown",
  "value": []
},
"creator_id": {
  "label": "Created by",
  "type": "dropdown",
  "value": 39897
},
"active_sales_sequences": {
  "label": "Active sales sequences",
  "type": "multi_select_dropdown",
  "value": []
},
"owner_id": {
  "label": "Sales owner",
  "type": "dropdown",
  "value": 39897
},
"is_deleted": false,
"last_assigned_at": {
  "label": "Last assigned at",
  "type": "date_time",
  "value": "2020-06-24T18:30:55Z"
},
"expected_close": {
  "label": "Expected close date",
  "type": "date",
  "value": null
},
"won_from": null,
"age": {
```

```
    "label": "Age (in days)",
    "type": "number",
    "value": 0
  },
  "amount": {
    "label": "Deal value",
    "type": "decimal",
    "value": "123.0"
  },
  "product_ids": [],
  "deal_stage_id": {
    "label": "Deal stage",
    "type": "dropdown",
    "value": 178962
  },
  "base_currency_amount": {
    "label": "Deal value in Base Currency",
    "type": "decimal",
    "value": "123.0"
  },
  "team_user_ids": [],
  "id": 1357538,
  "deal_reason_id": {
    "label": "Lost reason",
    "type": "dropdown",
    "value": null
  },
  "territory_id": {
    "label": "Territory",
    "type": "dropdown",
    "value": null
  },
  "last_contacted_via_sales_activity": {
```

```
    "label": "Last activity date",
    "type": "date_time",
    "value": null
  },
  "sales_account_id": {
    "label": "Account name",
    "type": "auto_complete",
    "value": null
  },
  "deal_payment_status_id": {
    "label": "Payment status",
    "type": "dropdown",
    "value": null
  },
  "created_at": {
    "label": "Created at",
    "type": "date_time",
    "value": "2020-06-24T18:30:54Z"
  },
  "import_id": null,
  "expected_deal_value": {
    "label": "Expected deal value",
    "type": "decimal",
    "value": "123.0"
  },
  "currency_id": {
    "label": "Currency",
    "type": "dropdown",
    "value": 6615
  },
  "recent_note": {
    "label": "Recent note",
    "type": "paragraph",
```

```
    "value": null
  },
  "stage_updated_time": {
    "label": "Deal stage updated at",
    "type": "date_time",
    "value": "2020-06-24T18:30:54Z"
  },
  "updated_at": {
    "label": "Updated at",
    "type": "date_time",
    "value": "2020-06-24T18:30:54Z"
  },
  "last_contacted_sales_activity_mode": {
    "label": "Last activity type",
    "type": "dropdown",
    "value": null
  },
  "completed_sales_sequences": {
    "label": "Completed sales sequences",
    "type": "multi_select_dropdown",
    "value": []
  },
  "campaign_id": {
    "label": "Campaign",
    "type": "dropdown",
    "value": null
  },
  "closed_date": {
    "label": "Closed date",
    "type": "date",
    "value": null
  },
  "lost_from": null,
```

```
"deal_prediction_tag": {
  "label": "Deal Prediction",
  "dom_type": "text",
  "value": null
},
"updater_id": {
  "label": "Updated by",
  "type": "dropdown",
  "value": null
},
"probability": {
  "label": "Probability (%)",
  "type": "decimal",
  "value": 100
}
},
"associations": {
  "source": null,
  "deal_payment_status": null,
  "campaign": null,
  "deal_pipeline": {
    "id": 27149,
    "name": "Default Pipeline",
    "position": 1,
    "is_default": true
  },
  "products": [],
  "contacts": [],
  "creator": {
    "deal_pipeline_id": 27149,
    "name": "Jim Doe",
    "is_active": true,
    "email": "jim.doe@xyz.com",
```

```
    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "created_at": "2020-06-24T08:32:11Z",
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
    "mobile_number": null
  },
  "sales_account": null,
  "deal_stage": {
    "id": 178962,
    "name": "New",
    "position": 1,
    "forecast_type": "Open"
  },
  "territory": null,
  "deal_product": null,
  "currency": {
    "id": 6615,
    "currency_code": "USD",
    "exchange_rate": "1.0"
  },
  "owner": {
    "deal_pipeline_id": 27149,
    "name": "Jim Doe",
    "is_active": true,
    "email": "jim.doe@xyz.com",
    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "created_at": "2020-06-24T08:32:11Z",
```



```

        "type": "user",
        "updated_at": "2020-06-24T08:34:20Z",
        "work_number": null,
        "time_zone": "UTC",
        "mobile_number": null
    },
    "updater": null,
    "deal_reason": null,
    "deal_type": null,
    "team_users": []
}
},
"account_id": "195818342531129946",
"domain": "sample.myfreshworks.com",
"event": "onDealCreate",
"timestamp": 1593023454.5852625,
"version": "2.0"
}

```

#### Attributes of the **data** object

ATTRIBUTE	DATA TYPE	DESCRIPTION
actor	<a href="#">actor object</a>	Details of the entity that triggered the onDealCreate event.
deal	<a href="#">deal object</a>	Details of the deal object created when the onDealCreate event is triggered.
associations	<a href="#">object</a>	Associated objects related to the deal object.

#### Attributes of the **deal** object

ATTRIBUTE	DATA TYPE	DESCRIPTION
deal_pipeline_id	object	Identifier of the deal pipeline.
source_additional_info	array	Meta information specified as a custom JSON object.
deal_product_id	object	ID of the product that the deal is associated with.
name mandatory	object	Name of the deal.
custom_fields	array of strings	Key value pairs containing the names and values of custom fields.
upcoming_activities_time	object	Timestamp for next scheduled activity.
deal_type_id	object	ID of the deal type.
first_assigned_at	object	First assigned timestamp of the deal.
web_form_id	object	Form through which this deal was captured.
contact_ids	array	IDs of contacts associated with this deal.
lead_source_id	object	ID of the source of the deal.
tags	object	Tags associated with this deal.
import_csv_id	object	Reference of all imports that this deal was part of.
creator_id	object	ID of the user who initiated the deal.
active_sales_sequences	object	Active sales campaigns that this deal was part of.
owner_id	object	User who owns the record.

is_deleted	boolean	Set to 'true' if the deal is deleted.
last_assigned_at	object	Last assigned timestamp.
expected_close	object	Expected close date of the deal.
won_from	object	Stage from which the deal was won.
job_title	object	Job title of the contact.
age	object	Time elapsed (in days) since the creation of the deal.
amount mandatory	object	Value of the deal.
product_ids	array	IDs of products associated with this deal.
deal_stage_id	object	ID of the deal stage.
base_currency_amount mandatory	object	Deal amount converted to base currency configured by admin.
team_user_ids	array	Identifiers of the collaborators who work on a task associated with the deal.
id	number	ID of the deal.
deal_reason_id	object	ID of the reason why the deal was lost.
territory_id	object	ID of the territory.
last_contacted_via_sales_activity	object	Last contacted timestamp (through sales activity).
sales_account_id	object	ID of sales account associated with this deal.
deal_payment_status	object	ID of deal payment status.

s_id		
created_at	object	Created timestamp.
import_id	string	ID of the equivalent record in the external system from which it was imported.
expected_deal_value	object	Expected deal amount.
currency_id	object	ID of currency associated with the deal.
recent_note	object	Recently added note against the deal.
stage_updated_time	object	Timestamp when the deal stage was updated.
updated_at	object	Updated timestamp.
last_contacted_sales_activity_mode	object	Last mode of sales activity rendered for the deal.
completed_sales_sequences	object	Completed sales sequences that this deal was part of.
campaign_id	object	ID of the UTM campaign.
closed_date	object	Closed date of the deal.
lost_from	string	Stage where the deal was lost.
deal_prediction_tag	object	ID of the user who updated the deal.
updater_id	object	Probability of closing the deal.
probability	object	Associations of the deal.

Attributes of the deal **associations** object

ATTRIBUTE	DATA TYPE	DESCRIPTION
source	string	Source details of this contact.
deal_payment_status	string	Status of the deal payment.
campaign	string	UTM campaign.
deal_pipeline	object	Details of deal pipeline.
products	array	Products associated with this deal.
contacts	array	Contacts associated with this deal.
creator	<a href="#">object</a>	Details of contact creator.
sales_accounts	array of numbers	Sales accounts of the contact.
deal_stage	object	Details of the deal stage.
territory	string	Territory of the contact.
deal_product	string	Product that the deal is associated with.
currency	object	Details of the deal currency.
owner	<a href="#">object</a>	Details of the contact owner.
updater	<a href="#">object</a>	Details of the deal updater. For the onDealCreate event, this value is null.
deal_reason	string	Reason why the deal was lost.
deal_type	string	Type of deal.
team_users	array	Collaborators who worked on a task related to the deal.

## onDealUpdate

When a deal is updated, the **onDealUpdate** event is triggered. To register the **onDealUpdate** event and the corresponding callback, use the following sample **server.js** content.

```
exports = {
  events: [
    { event: "onDealUpdate", callback:
"onDealUpdateCallback" },
  ],
  onDealUpdateCallback: function(payload) {
    console.log("Logging arguments from onDealUpdate event:
" + JSON.stringify(payload));
  }
}
```

## Sample Payload

```
{
  "iparams": {},
  "region": "US",
  "productContext": {
    "url": "https://sample.myfreshworks.com/crm",
    "name": "freshworks_crm"
  },
  "data": {
    "actor": {
      "deal_pipeline_id": 27149,
      "name": "Jim Doe",
      "is_active": true,
```

```
"email": "jim.doe@xyz.com",
"job_title": "Engineer",
"uuid": "195818342547907164",
"id": 39897,
"language": "en",
"created_at": "2020-06-24T08:32:11Z",
"type": "user",
"updated_at": "2020-06-24T08:34:20Z",
"work_number": null,
"time_zone": "UTC",
"is_forgotten": false,
"mobile_number": null
},
"deal": {
  "deal_pipeline_id": {
    "label": "Deal pipeline",
    "type": "dropdown",
    "value": 27149
  },
  "source_additional_info": {},
  "deal_product_id": {
    "label": "Product",
    "type": "dropdown",
    "value": null
  },
  "name": {
    "label": "Name",
    "type": "text",
    "value": "Deal update with owner"
  },
  "custom_fields": [],
  "upcoming_activities_time": {
    "label": "Upcoming activities",
```

```
    "type": "date_time",
    "value": null
  },
  "deal_type_id": {
    "label": "Type",
    "type": "dropdown",
    "value": null
  },
  "first_assigned_at": {
    "label": "First assigned at",
    "dom_type": "date_time",
    "value": "2020-06-24T18:30:55Z"
  },
  "web_form_id": {
    "label": "Web form",
    "type": "dropdown",
    "value": null
  },
  "contact_ids": [],
  "lead_source_id": {
    "label": "Source",
    "type": "dropdown",
    "value": null
  },
  "tags": {
    "label": "Tags",
    "type": "auto_complete",
    "value": []
  },
  "import_csv_id": {
    "label": "Import label",
    "dom_type": "multi_select_dropdown",
    "value": []
  }
```



```

},
"creator_id": {
  "label": "Created by",
  "type": "dropdown",
  "value": 39897
},
"active_sales_sequences": {
  "label": "Active sales sequences",
  "type": "multi_select_dropdown",
  "value": []
},
"owner_id": {
  "label": "Sales owner",
  "type": "dropdown",
  "value": 39897
},
"is_deleted": false,
"last_assigned_at": {
  "label": "Last assigned at",
  "type": "date_time",
  "value": "2020-06-24T18:30:55Z"
},
"expected_close": {
  "label": "Expected close date",
  "type": "date",
  "value": null
},
"won_from": null,
"age": {
  "label": "Age (in days)",
  "type": "number",
  "value": 0
},

```

```
"amount": {
  "label": "Deal value",
  "type": "decimal",
  "value": "1234.0"
},
"product_ids": [],
"deal_stage_id": {
  "label": "Deal stage",
  "type": "dropdown",
  "value": 178962
},
"base_currency_amount": {
  "label": "Deal value in Base Currency",
  "type": "decimal",
  "value": "1234.0"
},
"team_user_ids": [],
"id": 1357538,
"deal_reason_id": {
  "label": "Lost reason",
  "type": "dropdown",
  "value": null
},
"territory_id": {
  "label": "Territory",
  "type": "dropdown",
  "value": null
},
"last_contacted_via_sales_activity": {
  "label": "Last activity date",
  "type": "date_time",
  "value": null
},
```

```
"sales_account_id": {
  "label": "Account name",
  "type": "auto_complete",
  "value": null
},
"deal_payment_status_id": {
  "label": "Payment status",
  "type": "dropdown",
  "value": null
},
"created_at": {
  "label": "Created at",
  "type": "date_time",
  "value": "2020-06-24T18:30:54Z"
},
"import_id": null,
"expected_deal_value": {
  "label": "Expected deal value",
  "type": "decimal",
  "value": "1234.0"
},
"currency_id": {
  "label": "Currency",
  "type": "dropdown",
  "value": 6615
},
"recent_note": {
  "label": "Recent note",
  "type": "paragraph",
  "value": null
},
"stage_updated_time": {
  "label": "Deal stage updated at",
```

```
    "type": "date_time",
    "value": "2020-06-24T18:30:54Z"
  },
  "updated_at": {
    "label": "Updated at",
    "type": "date_time",
    "value": "2020-06-24T18:31:09Z"
  },
  "last_contacted_sales_activity_mode": {
    "label": "Last activity type",
    "type": "dropdown",
    "value": null
  },
  "completed_sales_sequences": {
    "label": "Completed sales sequences",
    "type": "multi_select_dropdown",
    "value": []
  },
  "campaign_id": {
    "label": "Campaign",
    "type": "dropdown",
    "value": null
  },
  "closed_date": {
    "label": "Closed date",
    "type": "date",
    "value": null
  },
  "lost_from": null,
  "deal_prediction_tag": {
    "label": "Deal Prediction",
    "dom_type": "text",
    "value": null
  }
```

```

    },
    "updater_id": {
      "label": "Updated by",
      "type": "dropdown",
      "value": 39897
    },
    "probability": {
      "label": "Probability (%)",
      "type": "decimal",
      "value": 100
    }
  },
  "associations": {
    "source": null,
    "deal_payment_status": null,
    "campaign": null,
    "deal_pipeline": {
      "id": 27149,
      "name": "Default Pipeline",
      "position": 1,
      "is_default": true
    },
    "products": [],
    "contacts": [],
    "creator": {
      "deal_pipeline_id": 27149,
      "name": "Jim Doe",
      "is_active": true,
      "email": "jim.doe@xyz.com",
      "job_title": "Engineer",
      "uuid": "195818342547907164",
      "id": 39897,
      "created_at": "2020-06-24T08:32:11Z",

```

```
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
    "mobile_number": null
  },
  "sales_account": null,
  "deal_stage": {
    "id": 178962,
    "name": "New",
    "position": 1,
    "forecast_type": "Open"
  },
  "territory": null,
  "deal_product": null,
  "currency": {
    "id": 6615,
    "currency_code": "USD",
    "exchange_rate": "1.0"
  },
  "owner": {
    "deal_pipeline_id": 27149,
    "name": "Jim Doe",
    "is_active": true,
    "email": "jim.doe@xyz.com",
    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "created_at": "2020-06-24T08:32:11Z",
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
```

```

        "mobile_number": null
    },
    "updater": {
        "deal_pipeline_id": 27149,
        "name": "Jim Doe",
        "is_active": true,
        "email": "jim.doe@xyz.com",
        "job_title": "Engineer",
        "uuid": "195818342547907164",
        "id": 39897,
        "created_at": "2020-06-24T08:32:11Z",
        "type": "user",
        "updated_at": "2020-06-24T08:34:20Z",
        "work_number": null,
        "time_zone": "UTC",
        "mobile_number": null
    },
    "deal_reason": null,
    "deal_type": null,
    "team_users": []
},
"changes": {
    "model_changes": {
        "name": [
            "Deal with owner",
            "Deal update with owner"
        ],
        "amount": [
            123,
            1234
        ],
        "base_currency_amount": [
            123,

```

```

        1234
    ],
    "expected_deal_value": [
        123,
        1234
    ],
    "updated_at": [
        "2020-06-24T18:30:54Z",
        "2020-06-24T18:31:09Z"
    ],
    "updater_id": [
        null,
        39897
    ]
},
"system_changes": {},
"misc_changes": {}
}
},
"account_id": "195818342531129946",
"domain": "sample.myfreshworks.com",
"event": "onDealUpdate",
"timestamp": 1593023469.8293262,
"version": "2.0"
}

```

The changes field is passed along with the payload. When a deal is deleted, the **is\_deleted** field is set to **'true'**.

Attributes of the **data** object

ATTRIBUTE	DATA TYPE	DESCRIPTION
actor	<a href="#">actor object</a>	Details of the entity that triggered the onDealUpdate event.



deal	<a href="#">deal object</a>	Details of the deal object.
associations	<a href="#">object</a>	Associated objects related to the deal object.
changes	object	Updated field values.

## onSalesAccountCreate

When a sales account is created, the **onSalesAccountCreate** event is triggered. To register the **onSalesAccountCreate** event and the corresponding callback, use the following sample **server.js** content.

```
exports = {
  events: [
    { event: "onSalesAccountCreate", callback:
"onSalesAccountCreateCallback" },
  ],
  onSalesAccountCreateCallback: function(payload) {
    console.log("Logging arguments from onSalesAccountCreate
event: " + JSON.stringify(payload));
  }
}
```

### Sample payload

```
{
  "iparams": {},
  "region": "US",
  "productContext": {
    "url": "https://sample.myfreshworks.com/crm",
    "name": "freshworks_crm"
  },
}
```

```
"data": {
  "actor": {
    "deal_pipeline_id": 27149,
    "name": "Jim Doe",
    "is_active": true,
    "email": "jim.doe@xyz.com",
    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "language": "en",
    "created_at": "2020-06-24T08:32:11Z",
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
    "is_forgotten": false,
    "mobile_number": null
  },
  "sales_account": {
    "open_deals_amount": {
      "label": "Open deals amount",
      "dom_type": "decimal",
      "value": "0.0"
    },
    "business_type_id": {
      "label": "Business type",
      "type": "dropdown",
      "value": null
    },
    "website": {
      "label": "Website",
      "type": "url",
      "value": null
    }
  }
}
```

```
,
"city": {
  "label": "City",
  "type": "text",
  "value": null
},
"zipcode": {
  "label": "Zipcode",
  "type": "text",
  "value": null
},
"name": {
  "label": "Name",
  "type": "text",
  "value": "sales account with owner"
},
"custom_fields": [],
"first_assigned_at": {
  "label": "First assigned at",
  "dom_type": "date_time",
  "value": "2020-06-24T18:31:38Z"
},
"avatar": null,
"last_contacted_mode": {
  "label": "Last contacted mode",
  "type": "dropdown",
  "value": null
},
"state": {
  "label": "State",
  "type": "text",
  "value": null
},
```

```
"parent_territory_id": null,
"tags": {
  "label": "Tags",
  "type": "auto_complete",
  "value": []
},
"linkedin": {
  "label": "LinkedIn",
  "type": "text",
  "value": null
},
"import_csv_id": {
  "label": "Import label",
  "dom_type": "multi_select_dropdown",
  "value": []
},
"creator_id": {
  "label": "Created by",
  "type": "dropdown",
  "value": 39897
},
"won_deals_amount": {
  "label": "Won deals amount",
  "dom_type": "decimal",
  "value": "0.0"
},
"active_sales_sequences": {
  "label": "Active sales sequences",
  "type": "multi_select_dropdown",
  "value": []
},
"owner_id": {
  "label": "Sales owner",
```

```
    "type": "dropdown",
    "value": 39897
  },
  "is_deleted": false,
  "last_assigned_at": {
    "label": "Last assigned at",
    "type": "date_time",
    "value": "2020-06-24T18:31:38Z"
  },
  "annual_revenue": {
    "label": "Annual revenue",
    "type": "decimal",
    "value": null
  },
  "country": {
    "label": "Country",
    "type": "text",
    "value": null
  },
  "team_user_ids": [],
  "id": 1100844,
  "territory_id": {
    "label": "Territory",
    "type": "dropdown",
    "value": null
  },
  "facebook": {
    "label": "Facebook",
    "type": "text",
    "value": null
  },
  "last_contacted": {
    "label": "Last contacted time",
```

```
    "type": "date_time",
    "value": null
  },
  "last_contacted_via_sales_activity": {
    "label": "Last activity date",
    "type": "date_time",
    "value": null
  },
  "status": "Active",
  "number_of_employees": {
    "label": "Number of employees",
    "type": "dropdown",
    "value": null
  },
  "mcr_id": 1275859140088463400,
  "industry_type_id": {
    "label": "Industry type",
    "type": "dropdown",
    "value": null
  },
  "address": {
    "label": "Address",
    "type": "text",
    "value": null
  },
  "created_at": {
    "label": "Created at",
    "type": "date_time",
    "value": "2020-06-24T18:31:37Z"
  },
  "import_id": null,
  "twitter": {
    "label": "Twitter",
```

```
    "type": "text",
    "value": null
  },
  "recent_note": {
    "label": "Recent note",
    "type": "paragraph",
    "value": null
  },
  "merged_to": 0,
  "updated_at": {
    "label": "Updated at",
    "type": "date_time",
    "value": "2020-06-24T18:31:37Z"
  },
  "first_contacted": {
    "label": "First contacted",
    "dom_type": "date_time",
    "value": null
  },
  "last_contacted_sales_activity_mode": {
    "label": "Last activity type",
    "type": "dropdown",
    "value": null
  },
  "completed_sales_sequences": {
    "label": "Completed sales sequences",
    "type": "multi_select_dropdown",
    "value": []
  },
  "phone": {
    "label": "Phone",
    "type": "phone_number",
    "value": null
  }
```

```

    },
    "updater_id": {
      "label": "Updated by",
      "type": "dropdown",
      "value": null
    },
    "parent_sales_account_id": {
      "label": "Parent account",
      "type": "auto_complete",
      "value": null
    },
    "parent_owner_id": null
  },
  "associations": {
    "creator": {
      "deal_pipeline_id": 27149,
      "name": "Jim Doe",
      "is_active": true,
      "email": "jim.doe.com",
      "job_title": "Engineer",
      "uuid": "195818342547907164",
      "id": 39897,
      "created_at": "2020-06-24T08:32:11Z",
      "type": "user",
      "updated_at": "2020-06-24T08:34:20Z",
      "work_number": null,
      "time_zone": "UTC",
      "mobile_number": null
    },
    "territory": null,
    "business_type": null,
    "parent_sales_account": null,
    "owner": {

```



```

    "deal_pipeline_id": 27149,
    "name": "Jim Doe",
    "is_active": true,
    "email": "jim.doe@xyz.com",
    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "created_at": "2020-06-24T08:32:11Z",
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
    "mobile_number": null
  },
  "updater": null,
  "industry_type": null,
  "team_users": []
}
},
"account_id": "195818342531129946",
"domain": "sample.myfreshworks.com",
"event": "onSalesAccountCreate",
"timestamp": 1593023498.1006536,
"version": "2.0"
}

```

Attributes of the **data** object

ATTRIBUTE	DATA TYPE	DESCRIPTION
actor	<a href="#">actor object</a>	Details of the entity that triggered the onSalesAccountCreate event.
sales_account	<a href="#">sales account object</a>	Details of the sales account object created when the onSalesAccountCreate event is triggered.

associations	<a href="#">object</a>	Associated objects related to the sales account object.
--------------	------------------------	---

Attributes of the sales\_account **actor** object

ATTRIBUTE	DATA TYPE	DESCRIPTION
deal_pipeline_id	number	ID of deal pipelines that this actor is part of.
name mandatory	string	Name of the actor.
is_active	boolean	Set to 'true' if the actor is verified.
email	string	Email address of the actor.
job_title	string	Designation of the actor.
uuid	string	Unique user identifier, this is auto-generated.
id	number	ID of the actor
language	string	Language of the actor; by default, language is 'en'.
created_at	string	Creation timestamp, specified in the UTC format.
type	string	Helps categorize an actor.
updated_at	string	Updation timestamp, specified in the UTC format. For onSalesAccountCreate, this value is the same as created_at.
work_number	number	Official phone number of the actor.
time_zone	string	Time zone of the actor, specified in the UTC format.
is_forgotten	boolean	If this attribute is 'true', specified data is permanently deleted.

mobile_number	string	Mobile number of the user.
---------------	--------	----------------------------

Attributes of the **sales account** object

ATTRIBUTE	DATA TYPE	DESCRIPTION
open_deals_amount	object	Sum of all the open deals associated with the account.
business_type_id	object	ID of the business type that the account belongs to.
website	object	URL of the account website.
city	object	City where the account was created.
zipcode	object	Postal code of the region.
name	object	Name of the account.
custom_fields	array	Key value pairs containing the names and values of custom fields.
first_assigned_at	object	First assigned timestamp.
avatar	object	Avatar URL.
last_contacted_mode	object	Last mode of contact. <b>Possible values:</b> <b>Email Incoming, Email Outgoing, Call Incoming, Call Outgoing, Email Opened, Chat, Sales Activity</b>
state	object	State where the account was created.
parent_territory_id	number	ID of the territory of the parent account.

tags	array of strings	Tags associated with this account.
linkedin	object	Linkedin address of the account owner.
import_csv_id	object	Reference of all imports that this account was part of.
creator_id	object	ID of the user who created the account.
won_deals_amount	object	Sum of all won deals associated with the account.
active_sales_sequences	object	Active sales campaigns that this account is part of.
owner_id	object	ID of the user who owns the record.
is_deleted	boolean	Set to 'true' if the account is deleted.
last_assigned_at	object	Last assigned timestamp.
annual_revenue	object	Annual revenue of the account.
country	object	Country where the account was created.
team_user_ids	array	Identifiers of the collaborators linked to a task associated with the sales account.
id	number	ID of the account.
territory_id	object	ID of the territory.
facebook	object	Facebook ID.
last_contacted	object	Last contacted timestamp.
last_contacted_via_sales_activity	object	Last contacted timestamp (through sales activity).

status	string	Status of the account.
number_of_employees	object	Number of employees associated with this account.
mcr_id	number	Identifier of the master customer record. This value is auto-generated.
industry_type_id	object	ID of the industry type that the account belongs to.
address	object	Address associated with the account.
created_at	object	Account creation timestamp.
import_id	string	ID of the equivalent record in the external system from which it was imported.
twitter	object	Twitter handle of the account owner.
recent_note	object	Recently added note against the account.
merged_to	object	Reference of secondary accounts merged to the primary account.
updated_at	object	Updated timestamp.
first_contacted	object	Last contacted via chat timestamp.
last_contacted_sales_activity_mode	object	Last mode of sales activity rendered.
completed_sales_sequences	object	Completed sales sequences that this account was part of.
phone	object	Phone number associated with this account.
updater_id	object	ID of the user who updated the account.

parent_sales_account_id	object	ID of parent account.
parent_owner_id	object	ID of the owner of the parent account.

Attributes of the sales\_account **associations** object

ATTRIBUTE	DATA TYPE	DESCRIPTION
creator	<a href="#">object</a>	Details of the account creator.
territory	string	Territory associated with the account.
business_type	string	Business type associated with this account.
parent_sales_account	string	Parent account.
owner	<a href="#">object</a>	Details of the account owner.
updater	<a href="#">object</a>	Details of the account updater.
industry_type	string	Industry type that the account belongs to.
team_users	array	Collaborators who work on a task associated with the sales account.

## onSalesAccountUpdate

When a sales account is updated, the **onSalesAccountUpdate** event is triggered. To register the **onSalesAccountUpdate** event and the corresponding callback, use the following sample **server.js** content.

```

exports = {
  events: [
    { event: "onSalesAccountUpdate", callback:
"onSalesAccountUpdateCallback" },
  ],
  onSalesAccountUpdateCallback: function(payload) {
    console.log("Logging arguments from onSalesAccountUpdate
event: " + JSON.stringify(payload));
  }
}

```

### Sample Payload

```

{
  "iparams": {},
  "region": "US",
  "productContext": {
    "url": "https://sample.myfreshworks.com/crm",
    "name": "freshworks_crm"
  },
  "data": {
    "actor": {
      "deal_pipeline_id": 27149,
      "name": "Jim Doe",
      "is_active": true,
      "email": "jim.doe@xyz.com",
      "job_title": "Engineer",
      "uuid": "195818342547907164",
      "id": 39897,
      "language": "en",
      "created_at": "2020-06-24T08:32:11Z",
      "type": "user",

```

```
"updated_at": "2020-06-24T08:34:20Z",
"work_number": null,
"time_zone": "UTC",
"is_forgotten": false,
"mobile_number": null
},
"sales_account": {
  "open_deals_amount": {
    "label": "Open deals amount",
    "dom_type": "decimal",
    "value": "0.0"
  },
  "business_type_id": {
    "label": "Business type",
    "type": "dropdown",
    "value": null
  },
  "website": {
    "label": "Website",
    "type": "url",
    "value": null
  },
  "city": {
    "label": "City",
    "type": "text",
    "value": null
  },
  "zipcode": {
    "label": "Zipcode",
    "type": "text",
    "value": null
  },
  "name": {
```



```

    "label": "Name",
    "type": "text",
    "value": "sales account update with owner"
  },
  "custom_fields": [],
  "first_assigned_at": {
    "label": "First assigned at",
    "dom_type": "date_time",
    "value": "2020-06-24T18:31:38Z"
  },
  "avatar": null,
  "last_contacted_mode": {
    "label": "Last contacted mode",
    "type": "dropdown",
    "value": null
  },
  "state": {
    "label": "State",
    "type": "text",
    "value": null
  },
  "parent_territory_id": null,
  "tags": {
    "label": "Tags",
    "type": "auto_complete",
    "value": []
  },
  "linkedin": {
    "label": "LinkedIn",
    "type": "text",
    "value": null
  },
  "import_csv_id": {

```

```
    "label": "Import label",
    "dom_type": "multi_select_dropdown",
    "value": []
  },
  "creator_id": {
    "label": "Created by",
    "type": "dropdown",
    "value": 39897
  },
  "won_deals_amount": {
    "label": "Won deals amount",
    "dom_type": "decimal",
    "value": "0.0"
  },
  "active_sales_sequences": {
    "label": "Active sales sequences",
    "type": "multi_select_dropdown",
    "value": []
  },
  "owner_id": {
    "label": "Sales owner",
    "type": "dropdown",
    "value": 39897
  },
  "is_deleted": false,
  "last_assigned_at": {
    "label": "Last assigned at",
    "type": "date_time",
    "value": "2020-06-24T18:31:38Z"
  },
  "annual_revenue": {
    "label": "Annual revenue",
    "type": "decimal",
```

```
        "value": null
    },
    "country": {
        "label": "Country",
        "type": "text",
        "value": null
    },
    "team_user_ids": [],
    "id": 1100844,
    "territory_id": {
        "label": "Territory",
        "type": "dropdown",
        "value": null
    },
    "facebook": {
        "label": "Facebook",
        "type": "text",
        "value": null
    },
    "last_contacted": {
        "label": "Last contacted time",
        "type": "date_time",
        "value": null
    },
    "last_contacted_via_sales_activity": {
        "label": "Last activity date",
        "type": "date_time",
        "value": null
    },
    "status": "Active",
    "number_of_employees": {
        "label": "Number of employees",
        "type": "dropdown",
```

```
      "value": null
    },
    "mcr_id": 1275859140088463400,
    "industry_type_id": {
      "label": "Industry type",
      "type": "dropdown",
      "value": null
    },
    "address": {
      "label": "Address",
      "type": "text",
      "value": null
    },
    "created_at": {
      "label": "Created at",
      "type": "date_time",
      "value": "2020-06-24T18:31:37Z"
    },
    "import_id": null,
    "twitter": {
      "label": "Twitter",
      "type": "text",
      "value": null
    },
    "recent_note": {
      "label": "Recent note",
      "type": "paragraph",
      "value": null
    },
    "merged_to": 0,
    "updated_at": {
      "label": "Updated at",
      "type": "date_time",
```

```
    "value": "2020-06-24T18:32:02Z"
  },
  "first_contacted": {
    "label": "First contacted",
    "dom_type": "date_time",
    "value": null
  },
  "last_contacted_sales_activity_mode": {
    "label": "Last activity type",
    "type": "dropdown",
    "value": null
  },
  "completed_sales_sequences": {
    "label": "Completed sales sequences",
    "type": "multi_select_dropdown",
    "value": []
  },
  "phone": {
    "label": "Phone",
    "type": "phone_number",
    "value": "123123123"
  },
  "updater_id": {
    "label": "Updated by",
    "type": "dropdown",
    "value": 39897
  },
  "parent_sales_account_id": {
    "label": "Parent account",
    "type": "auto_complete",
    "value": null
  },
  "parent_owner_id": null
```

```
},
"associations": {
  "creator": {
    "deal_pipeline_id": 27149,
    "name": "Jim Doe",
    "is_active": true,
    "email": "jim.doe@xyz.com",
    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "created_at": "2020-06-24T08:32:11Z",
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
    "mobile_number": null
  },
  "territory": null,
  "business_type": null,
  "parent_sales_account": null,
  "owner": {
    "deal_pipeline_id": 27149,
    "name": "Jim Doe",
    "is_active": true,
    "email": "jim.doe@xyz.com",
    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "created_at": "2020-06-24T08:32:11Z",
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
```

```

        "mobile_number": null
    },
    "updater": {
        "deal_pipeline_id": 27149,
        "name": "Jim Doe",
        "is_active": true,
        "email": "jim.doe@xyz.com",
        "job_title": "Engineer",
        "uuid": "195818342547907164",
        "id": 39897,
        "created_at": "2020-06-24T08:32:11Z",
        "type": "user",
        "updated_at": "2020-06-24T08:34:20Z",
        "work_number": null,
        "time_zone": "UTC",
        "mobile_number": null
    },
    "industry_type": null,
    "team_users": []
},
"changes": {
    "model_changes": {
        "name": [
            "sales account with owner",
            "sales account update with owner"
        ],
        "created_at": [
            "2020-06-24T18:31:37Z",
            "2020-06-24T18:31:37Z"
        ],
        "updated_at": [
            "2020-06-24T18:31:37Z",
            "2020-06-24T18:32:02Z"
        ]
    }
}

```

```

    ],
    "phone": [
        null,
        "123123123"
    ],
    "updater_id": [
        null,
        39897
    ]
  },
  "system_changes": {},
  "misc_changes": {}
}
},
"account_id": "195818342531129946",
"domain": "sample.myfreshworks.com",
"event": "onSalesAccountUpdate",
"timestamp": 1593023522.2470818,
"version": "2.0"
}

```

The **changes** field is passed along with the payload. When a sales account is deleted, the **is\_deleted** field is set to **'true'**.

Attributes of the **data** object.

ATTRIBUTE	DATA TYPE	DESCRIPTION
actor	<a href="#">actor object</a>	Details of the entity that triggered the onSalesAccountUpdate event.
sales_account	<a href="#">sales account object</a>	Details of the sales account object.
associations	<a href="#">object</a>	Associated objects related to the sales account object.



changes	object	Updated field values.
---------	--------	-----------------------

## onCustomModuleCreate

When a custom module record is created, the **onCustomModuleCreate** event is triggered. To register the **onCustomModuleCreate** event and the corresponding callback, use the following sample **server.js** content.

```
exports = {
  events: [
    { event: "onCustomModuleCreate", callback:
"onCustomModuleCreateCallback" },
  ],
  onCustomModuleCreateCallback: function(payload) {
    console.log("Logging arguments from onCustomModuleCreate
event: " + JSON.stringify(payload));
  }
}
```

## Sample Payload

```
{
  "iparams": {},
  "region": "US",
  "productContext": {
    "url": "https://sample.myfreshworks.com/crm",
    "name": "freshworks_crm"
  },
  "data": {
    "actor": {
      "deal_pipeline_id": 27149,
      "name": "Jim Doe",
```

```

    "is_active": true,
    "email": "jim.doe@xyz.com",
    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "language": "en",
    "created_at": "2020-06-24T08:32:11Z",
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
    "is_forgotten": false,
    "mobile_number": null
  },
  "custom_module_record": {
    "name": {
      "label": "Name",
      "type": "text",
      "value": "custom_create"
    },
    "custom_fields": [
      {
        "name": "cf_cm_edit",
        "label": "CM_edit",
        "column": "cf_str01",
        "type": "text",
        "value": null
      }
    ],
    "creator_id": {
      "label": "Created by",
      "type": "auto_complete",
      "value": {

```

```
        "id": 39897,
        "model": "user"
    }
},
"owner_id": {
    "label": "Sales owner",
    "type": "auto_complete",
    "value": {
        "id": 39897,
        "model": "user"
    }
},
"is_deleted": false,
"id": 40764,
"status": 1,
"created_at": {
    "label": "Created at",
    "type": "date_time",
    "value": "2020-06-24T14:05:16Z"
},
"import_id": null,
"recent_note": {
    "label": "Recent note",
    "type": "paragraph",
    "value": null
},
"updated_at": {
    "label": "Updated at",
    "type": "date_time",
    "value": "2020-06-24T14:05:16Z"
},
"updater_id": {
    "label": "Updated by",
```

```
    "type": "auto_complete",
    "value": {
      "id": null,
      "model": "user"
    }
  },
  "module_name": "cm_create"
},
"associations": {
  "owner": {
    "deal_pipeline_id": 27149,
    "name": "Jim Doe",
    "is_active": true,
    "email": "jim.doe@xyz.com",
    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "created_at": "2020-06-24T08:32:11Z",
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
    "mobile_number": null
  },
  "updater": null,
  "creator": {
    "deal_pipeline_id": 27149,
    "name": "Jim Doe",
    "is_active": true,
    "email": "jim.doe@xyz.com",
    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
```

```

        "created_at": "2020-06-24T08:32:11Z",
        "type": "user",
        "updated_at": "2020-06-24T08:34:20Z",
        "work_number": null,
        "time_zone": "UTC",
        "mobile_number": null
    }
}
},
"account_id": "195818342531129946",
"domain": "sample.myfreshworks.com",
"event": "onCustomModuleCreate",
"timestamp": 1593007516.6833186,
"version": "2.0"
}

```

Attributes of the **data** object

ATTRIBUTE	DATA TYPE	DESCRIPTION
actor	<a href="#">actor object</a>	Details of the entity that triggered the onCustomModuleCreate event.
custom_module_record	<a href="#">custom module record object</a>	Details of the custom module record object created when the onCustomModuleCreate event is triggered.
associations	<a href="#">object</a>	Associated objects related to the custom module record object.

Attributes of the custom module record object

ATTRIBUTE	DATA TYPE	DESCRIPTION
-----------	-----------	-------------

name	object	Name in the record.
custom_fields	array	Key value pairs containing the names and values of custom fields.
creator_id	object	Details of the user who created the record.
owner_id	object	Details of the user who created the record.
is_deleted	boolean	Set to 'true' if record is deleted.
id	number	ID of the record.
status	number	Status of the record.
created_at	object	Record creation timestamp.
import_id	string	ID of the equivalent record in the external system from which it was imported.
recent_note	object	Recently added note against the record.
updated_at	object	Updated timestamp. For onCustomModuleCreate, updated_at is the same as created_at.
updater_id	object	Details of the user who updated the record.
module_name	object	Name of the module.

Attributes of the record **associations** object

ATTRIBUTE	DATA TYPE	DESCRIPTION
owner	<a href="#">object</a>	Details of custom module record creator.
updater	<a href="#">object</a>	Details of the custom module record updater.
creator	<a href="#">object</a>	Details of the custom module record creator.

## onCustomModuleUpdate

When a custom module record is updated, the **onCustomModuleUpdate** event is triggered. To register the **onCustomModuleUpdate** event and the corresponding callback, use the following sample **server.js** content.

```
exports = {  
  events: [  
    { event: "onCustomModuleUpdate", callback:  
      "onCustomModuleUpdateCallback" }  
  ],  
  onCustomModuleUpdateCallback: function(payload) {  
    console.log("Logging arguments from onCustomModuleUpdate  
event: " + JSON.stringify(payload));  
  }  
}
```

## Sample Payload

```
{  
  "iparams": {},  
  "region": "US",  
  "productContext": {  
    "url": "https://sample.myfreshworks.com/crm",  
    "name": "freshworks_crm"  
  },  
  "data": {  
    "actor": {  
      "deal_pipeline_id": 27149,  
      "name": "Jim Doe",  

```

```

    "is_active": true,
    "email": "jim.doe@xyz.com",
    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "language": "en",
    "created_at": "2020-06-24T08:32:11Z",
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
    "is_forgotten": false,
    "mobile_number": null
  },
  "custom_module_record": {
    "name": {
      "label": "Name",
      "type": "text",
      "value": "custom_create_update"
    },
    "custom_fields": [
      {
        "name": "cf_cm_edit",
        "label": "CM_edit",
        "column": "cf_str01",
        "type": "text",
        "value": null
      }
    ],
    "creator_id": {
      "label": "Created by",
      "type": "auto_complete",
      "value": {

```



```
        "id": 39897,
        "model": "user"
    }
},
"owner_id": {
    "label": "Sales owner",
    "type": "auto_complete",
    "value": {
        "id": 39897,
        "model": "user"
    }
},
"is_deleted": false,
"id": 40764,
"status": 1,
"created_at": {
    "label": "Created at",
    "type": "date_time",
    "value": "2020-06-24T14:05:16Z"
},
"import_id": null,
"recent_note": {
    "label": "Recent note",
    "type": "paragraph",
    "value": null
},
"updated_at": {
    "label": "Updated at",
    "type": "date_time",
    "value": "2020-06-24T14:06:21Z"
},
"updater_id": {
    "label": "Updated by",
```

```
    "type": "auto_complete",
    "value": {
      "id": 39897,
      "model": "user"
    }
  },
  "module_name": "cm_create"
},
"associations": {
  "owner": {
    "deal_pipeline_id": 27149,
    "name": "Jim Doe",
    "is_active": true,
    "email": "jim.doe@xyz.com",
    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "created_at": "2020-06-24T08:32:11Z",
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
    "mobile_number": null
  },
  "updater": {
    "deal_pipeline_id": 27149,
    "name": "Jim Doe",
    "is_active": true,
    "email": "jim.doe@xyz.com",
    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "created_at": "2020-06-24T08:32:11Z",
```

```

    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
    "mobile_number": null
  },
  "creator": {
    "deal_pipeline_id": 27149,
    "name": "Jim Doe",
    "is_active": true,
    "email": "jim.doe@xyz.com",
    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "created_at": "2020-06-24T08:32:11Z",
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
    "mobile_number": null
  }
},
"changes": {
  "model_changes": {
    "name": [
      "custom_create",
      "custom_create_update"
    ],
    "updater_id": [
      null,
      39897
    ],
    "updated_at": [

```

```

        "2020-06-24T14:05:16Z",
        "2020-06-24T14:06:21Z"
    ]
},
"system_changes": {},
"misc_changes": {}
}
},
"account_id": "195818342531129946",
"domain": "sample.myfreshworks.com",
"event": "onCustomModuleUpdate",
"timestamp": 1593007581.9710617,
"version": "2.0"
}

```

The changes field is passed along with the payload. When a **record** is deleted, the **is\_deleted** field is set to **'true'**.

Attributes of the **data** object

ATTRIBUTE	DATA TYPE	DESCRIPTION
actor	<a href="#">actor object</a>	Details of the entity that triggered the onCustomModuleUpdate event.
custom_module_record	<a href="#">custom module record object</a>	Details of the custom module record object.
associations	<a href="#">object</a>	Associated objects related to the custom module record object.
changes	object	Updated field values.

## onAppointmentCreate

When an appointment is created, the **onAppointmentCreate** event is triggered. To register the **onAppointmentCreate** event and the corresponding callback, use the following sample **server.js** content.

```
exports = {  
  events: [  
    { event: "onAppointmentCreate", callback:  
      "onAppointmentCreateCallback" }  
  ],  
  onAppointmentCreateCallback: function(payload) {  
    console.log("Logging arguments from onAppointmentCreate  
event: " + JSON.stringify(payload));  
  }  
}
```

## Sample Payload

```
{  
  "iparams": {},  
  "region": "US",  
  "productContext": {  
    "url": "https://sample.myfreshworks.com/crm",  
    "name": "freshworks_crm"  
  },  
  "data": {  
    "actor": {  
      "deal_pipeline_id": 27149,  
      "name": "Jim Doe",  
      "is_active": true,  
      "email": "jim.doe@xyz.com",  
      "job_title": "Engineer",  
    },  
  },  
}
```

```

    "uuid": "195818342547907164",
    "id": 39897,
    "language": "en",
    "created_at": "2020-06-24T08:32:11Z",
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
    "is_forgotten": false,
    "mobile_number": null
  },
  "appointment": {
    "source_additional_info": {},
    "targetable_id": null,
    "location": null,
    "latitude": null,
    "description": "asds",
    "creator_id": 39897,
    "targetable_type": null,
    "is_deleted": false,
    "longitude": null,
    "id": 283387,
    "is_allday": false,
    "end_date": "2020-06-24T17:00:45Z",
    "outcome_id": null,
    "created_at": "2020-06-24T16:18:10Z",
    "from_date": "2020-06-24T16:30:45Z",
    "title": "Test",
    "updated_at": "2020-06-24T16:18:10Z",
    "time_zone": "UTC",
    "checkedin_at": null,
    "updater_id": null
  },

```

```

"associations": {
  "targetable": null,
  "outcome": null,
  "creator": {
    "deal_pipeline_id": 27149,
    "name": "Jim Doe",
    "is_active": true,
    "email": "jim.doe@xyz.com",
    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "created_at": "2020-06-24T08:32:11Z",
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
    "mobile_number": null
  },
  "appointment_attendees": [
    {
      "_association_id": 346754,
      "id": 39897,
      "_model": "FdMultitenant::User"
    },
    {
      "_association_id": 346755,
      "id": 1399366,
      "_model": "Contact"
    }
  ],
  "updater": null
}
},

```

```

    "account_id": "195818342531129946",
    "domain": "sample.myfreshworks.com",
    "event": "onAppointmentCreate",
    "timestamp": 1593015490.2012472,
    "version": "2.0"
  }

```

Attributes of the **data** object.

ATTRIBUTE	DATA TYPE	DESCRIPTION
actor	<a href="#">actor object</a>	Details of the entity that triggered the onAppointmentCreate event.
appointment	<a href="#">appointment object</a>	Details of the appointment object created when the onAppointmentCreate event is triggered.
associations	<a href="#">object</a>	Associated objects related to the appointment object.

Attributes of the **appointment** object.

ATTRIBUTE	DATA TYPE	DESCRIPTION
source_additional_info	object	Meta information specified as a custom JSON object.
targetable_id	string	ID of the contact/salesaccount/deal for whom appointment has been created.
location		Location where the appointment is scheduled.
latitude	string	Latitude of the location where the appointment is scheduled.
description	string	Description of the appointment.
creator_id	number	ID of the user who created the appointment.



targetable_type	string	Describes the entity value for whom the appointment was created. <b>Possible value:</b> Contact
is_deleted	boolean	Set to 'true' if the appointment is deleted.
longitude	string	Longitude of the location where the appointment is scheduled.
id	number	ID of the appointment.
is_allday	boolean	Specifies if the appointment is an all-day appointment or not, the default value is false.
end_date mandatory	string	Timestamp indicating the end of the appointment, or the end date, if this is an all-day appointment, specified in the UTC format.
outcome_id	number	ID of the appointment outcome.
created_at	string	Appointment creation timestamp, specified in the UTC format.
from_date mandatory	string	Timestamp indicating the start of the appointment, or the start date if this is an all-day appointment, specified in the UTC format.
title mandatory	string	Title of the appointment.
updated_at	string	Appointment updated timestamp, specified in the UTC format.
time_zone	string	Time zone of the region where the appointment is scheduled.

checkedin_at	string	Date and time of the scheduled appointment.
updater_id	string	ID of the user who created the appointment.

Attributes of the appointment **associations** object

ATTRIBUTE	DATA TYPE	DESCRIPTION
targetable	number	<p>Details of the contact/salesaccount/deal for whom the appointment has been created.</p> <p><b>Attributes of the targetable object</b></p> <p><b>id (number):</b> ID of the contact/salesaccount/deal</p> <p><b>_model (string):</b> Description of the entity value</p> <p><b>Possible values</b></p> <p><b>Contact, SalesAccount, Deal</b></p>
outcome	string	Details of the user who updated the appointment.
creator	<a href="#">object</a>	Details of the user who created the appointment.
appointment_attendees	array	Details of all the attendees associated with the appointment.
updater	<a href="#">object</a>	Details of the user who updated the appointment. For onAppointmentCreate, this value is null.

## onAppointmentUpdate

When an appointment is updated, the **onAppointmentUpdate** event is triggered. To register the **onAppointmentUpdate** event and the corresponding callback, use the following sample **server.js** content.

```
exports = {  
  events: [  
    { event: "onAppointmentUpdate", callback:  
      "onAppointmentUpdateCallback" }  
  ],  
  onAppointmentUpdateCallback: function(payload) {  
    console.log("Logging arguments from onAppointmentUpdate  
event: " + JSON.stringify(payload));  
  }  
}
```

## Sample Payload

```
{  
  "iparams": {},  
  "region": "US",  
  "productContext": {  
    "url": "https://sample.myfreshworks.com/crm",  
    "name": "freshworks_crm"  
  },  
  "data": {  
    "actor": {  
      "deal_pipeline_id": 27149,  
      "name": "Jim Doe",  
      "is_active": true,  
      "email": "jim.doe@xyz.com",  
    },  
  },  
}
```

```

    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "language": "en",
    "created_at": "2020-06-24T08:32:11Z",
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
    "is_forgotten": false,
    "mobile_number": null
  },
  "appointment": {
    "targetable_id": null,
    "location": null,
    "latitude": null,
    "description": "This is just an updated version of
sample Appointment.",
    "creator_id": 39897,
    "targetable_type": null,
    "is_deleted": false,
    "longitude": null,
    "id": 283270,
    "is_allday": false,
    "end_date": "2016-06-20T06:00:00Z",
    "outcome_id": null,
    "created_at": "2020-06-24T14:03:26Z",
    "from_date": "2016-06-20T05:00:00Z",
    "title": "Updated Appointment",
    "updated_at": "2020-06-24T14:03:27Z",
    "time_zone": "UTC",
    "checkedin_at": null,
    "updater_id": 39897
  }
}

```

```

},
"associations": {
  "targetable": null,
  "outcome": null,
  "creator": {
    "deal_pipeline_id": 27149,
    "name": "Jim Doe",
    "is_active": true,
    "email": "jim.doe@xyz.com",
    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "created_at": "2020-06-24T08:32:11Z",
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
    "mobile_number": null
  },
  "appointment_attendees": [
    {
      "_association_id": 346754,
      "id": 39897,
      "_model": "FdMultitenant::User"
    },
    {
      "_association_id": 346755,
      "id": 1399366,
      "_model": "Contact"
    }
  ],
  "updater": {
    "deal_pipeline_id": 27149,

```

```

    "name": "Jim Doe",
    "is_active": true,
    "email": "jim.doe@xyz.com",
    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "created_at": "2020-06-24T08:32:11Z",
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
    "mobile_number": null
  }
},
"changes": {
  "model_changes": {
    "title": [
      "Sample Appointment",
      "Updated Appointment"
    ],
    "description": [
      "This is just a sample Appointment.",
      "This is just an updated version of sample Appointment."
    ],
    "updater_id": [
      null,
      39897
    ],
    "updated_at": [
      "2020-06-24T14:03:26Z",
      "2020-06-24T14:03:27Z"
    ]
  }
}

```

```

    },
    "system_changes": {},
    "misc_changes": {}
  }
},
"account_id": "195818342531129946",
"domain": "sample.myfreshworks.com",
"event": "onAppointmentUpdate",
"timestamp": 1593007407.3063838,
"version": "2.0"
}

```

The changes field is passed along with the payload. When an appointment is deleted, the **is\_deleted** field is set to **'true'**.

Attributes of the **data** object.

ATTRIBUTE	DATA TYPE	DESCRIPTION
actor	<a href="#">actor object</a>	Details of the entity that triggered the onAppointmentUpdate event.
appointment	<a href="#">appointment object</a>	Details of the appointment object.
associations	<a href="#">object</a>	Associated objects related to the appointment object.
changes	object	Updated field values.

## onTaskCreate

When a task is created, the **onTaskCreate** event is triggered. To register the **onTaskCreate** event and the corresponding callback, use the following sample **server.js** content.

```

exports = {
  events: [
    { event: "onTaskCreate", callback:
"onTaskCreateCallback" }
  ],
  onTaskCreateCallback: function(payload) {
    console.log("Logging arguments from onTaskCreate event:
" + JSON.stringify(payload));
  }
}

```

### Sample Payload

```

{
  "iparams": {},
  "region": "US",
  "productContext": {
    "url": "https://sample.myfreshworks.com/crm",
    "name": "freshworks_crm"
  },
  "data": {
    "actor": {
      "deal_pipeline_id": 27149,
      "name": "Jim Doe",
      "is_active": true,
      "email": "jim.doe@xyz.com",
      "job_title": "Engineer",
      "uuid": "195818342547907164",
      "id": 39897,
      "language": "en",
      "created_at": "2020-06-24T08:32:11Z",
      "type": "user",
      "updated_at": "2020-06-24T08:34:20Z",

```



```

    "work_number": null,
    "time_zone": "UTC",
    "is_forgotten": false,
    "mobile_number": null
  },
  "task": {
    "source_additional_info": {},
    "targetable_id": null,
    "description": "task create",
    "creator_id": 39897,
    "owner_id": 39897,
    "targetable_type": null,
    "is_deleted": false,
    "task_type_id": null,
    "id": 301943,
    "status": "Open",
    "outcome_id": null,
    "completed_date": null,
    "due_date": "2020-06-25T14:30:00Z",
    "created_at": "2020-06-24T14:18:23Z",
    "title": "task create",
    "updated_at": "2020-06-24T14:18:23Z",
    "updater_id": null
  },
  "associations": {
    "targetable": null,
    "task_type": null,
    "outcome": null,
    "creator": {
      "deal_pipeline_id": 27149,
      "name": "Jim Doe",
      "is_active": true,
      "email": "jim.doe@xyz.com",

```

```

    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "created_at": "2020-06-24T08:32:11Z",
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
    "mobile_number": null
  },
  "owner": {
    "deal_pipeline_id": 27149,
    "name": "Jim Doe",
    "is_active": true,
    "email": "jim.doe@xyz.com",
    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "created_at": "2020-06-24T08:32:11Z",
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
    "mobile_number": null
  },
  "updater": null,
  "task_users": [
    {
      "_association_id": 428048,
      "user_id": 39897
    }
  ]
}

```

```

    },
    "account_id": "195818342531129946",
    "domain": "sample.myfreshworks.com",
    "event": "onTaskCreate",
    "timestamp": 1593008303.3584583,
    "version": "2.0"
  }

```

#### Attributes of the **data** object

ATTRIBUTE	DATA TYPE	DESCRIPTION
actor	<a href="#">actor object</a>	Details of the entity that triggered the onTaskCreate event.
task	<a href="#">task object</a>	Details of the task object created when the onTaskCreate event is triggered.
associations	<a href="#">object</a>	Associated objects related to the task object.

#### Attributes of the **task** object

ATTRIBUTE	DATA TYPE	DESCRIPTION
source_additional_info	object	Meta information specified as a custom JSON object.
targetable_id <b>mandatory</b>	number	ID of the contact/salesaccount/deal for whom the task has been created.
description	string	Description of the task.

creator_id	number	ID of the user who created the task.
owner_id	number	ID of the user to whom the task has been assigned.
targetable_type mandatory	string	Describes the entity value for whom the task was created. <b>Possible value:</b> Contact
is_deleted	boolean	Set to 'true' if the task has been deleted.
task_type_id	number	ID of task sales activity type.
id	number	ID of the task.
status	string	Status of the task. <b>Possible values:</b> <b>Open (0) and Completed (1)</b>
outcome_id	number	ID of outcome of task sales activity type.
completed_date	string	Timestamp indicating when the task was completed, specified in the UTC format.
due_date mandatory	string	Timestamp indicating when the task is to be completed, specified in the UTC format.
created_at	string	Task creation timestamp, specified in the UTC format.
title mandatory	string	Title of the task.
updated_at	string	Task updated timestamp, specified in the UTC format. For onTaskCreate, this value is the same as created_at.
updater_id	object	ID of the user who updated the task.

Attributes of the task **associations** object

ATTRIBUTE	DATA TYPE	DESCRIPTION
targetable	targetable object	<p>Details of the contact/salesaccount/deal for whom the task has been created.</p> <p><b>Attributes of the targetable object</b></p> <p><b>id (number):</b> ID of the contact/salesaccount/deal</p> <p><b>_model (string):</b> Description of the entity value.</p> <p><b>Possible values</b></p> <p>Contact, SalesAccount, Deal.</p>
task_type	task type object	<p>Description of the task sales activity type.</p> <p><b>Attributes of the task type object</b></p> <p><b>name (string):</b> Name of the task.</p> <p><b>sales_activity_type_id (number):</b> ID of the sales activity type.</p>
outcome	outcome object	<p>Description of the task sales activity type outcome.</p> <p><b>Attributes of the outcome object</b></p> <p><b>id (number):</b> ID of the task sales activity type outcome.</p> <p><b>name (string):</b> Description of the entity value.</p>
creator	<a href="#">object</a>	Details of the user who created the task.
owner	<a href="#">object</a>	Details of the user to whom the task has been assigned.
updater	<a href="#">object</a>	Details of the user who updated the task.
task_users	array of objects	Details (_association_id, user_id) of the task collaborators.

## onTaskUpdate

When a task is updated, the **onTaskUpdate** event is triggered. To register the **onTaskUpdate** event and the corresponding callback, use the following sample **server.js** content.

```
exports = {  
  events: [  
    { event: "onTaskUpdate", callback:  
      "onTaskUpdateCallback" }  
  ],  
  onTaskUpdateCallback: function(payload) {  
    console.log("Logging arguments from onTaskUpdate event:  
" + JSON.stringify(payload));  
  }  
}
```

## Sample Payload

```
{  
  "iparams": {},  
  "region": "US",  
  "productContext": {  
    "url": "https://sample.myfreshworks.com/crm",  
    "name": "freshworks_crm"  
  },  
  "data": {  
    "actor": {  
      "deal_pipeline_id": 27149,  
      "name": "Jim Doe",  
      "is_active": true,  
      "email": "jim.doe@xyz.com",  
      "job_title": "Engineer",  
    },  
  },  
}
```

```
"uuid": "195818342547907164",
"id": 39897,
"language": "en",
"created_at": "2020-06-24T08:32:11Z",
"type": "user",
"updated_at": "2020-06-24T08:34:20Z",
"work_number": null,
"time_zone": "UTC",
"is_forgotten": false,
"mobile_number": null
},
"task": {
  "source_additional_info": {},
  "targetable_id": null,
  "description": "task update",
  "creator_id": 39897,
  "owner_id": 39897,
  "targetable_type": null,
  "is_deleted": false,
  "task_type_id": null,
  "id": 301943,
  "status": "Open",
  "outcome_id": null,
  "completed_date": null,
  "due_date": "2020-06-25T14:30:00Z",
  "created_at": "2020-06-24T14:18:23Z",
  "title": "task update",
  "updated_at": "2020-06-24T14:18:38Z",
  "updater_id": 39897
},
"associations": {
  "targetable": null,
  "task_type": null,
```

```

"outcome": null,
"creator": {
  "deal_pipeline_id": 27149,
  "name": "Jim Doe",
  "is_active": true,
  "email": "jim.doe@xyz.com",
  "job_title": "Engineer",
  "uuid": "195818342547907164",
  "id": 39897,
  "created_at": "2020-06-24T08:32:11Z",
  "type": "user",
  "updated_at": "2020-06-24T08:34:20Z",
  "work_number": null,
  "time_zone": "UTC",
  "mobile_number": null
},
"owner": {
  "deal_pipeline_id": 27149,
  "name": "Jim Doe",
  "is_active": true,
  "email": "jim.doe@xyz.com",
  "job_title": "Engineer",
  "uuid": "195818342547907164",
  "id": 39897,
  "created_at": "2020-06-24T08:32:11Z",
  "type": "user",
  "updated_at": "2020-06-24T08:34:20Z",
  "work_number": null,
  "time_zone": "UTC",
  "mobile_number": null
},
"updater": {
  "deal_pipeline_id": 27149,

```



```

    "name": "Jim Doe",
    "is_active": true,
    "email": "jim.doe@xyz.com",
    "job_title": "Engineer",
    "uuid": "195818342547907164",
    "id": 39897,
    "created_at": "2020-06-24T08:32:11Z",
    "type": "user",
    "updated_at": "2020-06-24T08:34:20Z",
    "work_number": null,
    "time_zone": "UTC",
    "mobile_number": null
  },
  "task_users": [
    {
      "_association_id": 428048,
      "user_id": 39897
    }
  ]
},
"changes": {
  "model_changes": {
    "title": [
      "task create",
      "task update"
    ],
    "description": [
      "task create",
      "task update"
    ],
    "updated_at": [
      "2020-06-24T14:18:23Z",
      "2020-06-24T14:18:38Z"
    ]
  }
}

```

```

    ],
    "updater_id": [
        null,
        39897
    ]
},
"system_changes": {},
"misc_changes": {}
}
},
"account_id": "195818342531129946",
"domain": "sample.myfreshworks.com",
"event": "onTaskUpdate",
"timestamp": 1593008318.8261375,
"version": "2.0"
}

```

The **changes** field is passed along with the payload. When a **task** is deleted, the **is\_deleted** field is set to **'true'**.

Attributes of the **data** object

ATTRIBUTE	DATA TYPE	DESCRIPTION
actor	<a href="#">actor object</a>	Details of the entity that triggered the onTaskUpdate event.
task	<a href="#">task object</a>	Details of the task object.
associations	<a href="#">object</a>	Associated objects related to the task object.
changes	object	Updated field values.

## Testing

### Testing an Omni App

#### Notes:

- To test your serverless app, use the latest version of Chrome.
- As testing is only a simulation of events, actual data or record associated with the event is not created in the back-end. To create actual data and test the events, publish your app as a custom app and test the events manually.
- Ensure that the JSON files that contain the sample payloads to test the events are available at **<app's root directory>/server/test\_data**.
- To request for a trial Freshworks CRM account to test your app, email us at [marketplace@freshworks.com](mailto:marketplace@freshworks.com).

To simulate an event and test your app,

1. From the command line, navigate to the directory that contains the app related files and run the following command: `$ fdk run`
2. In the address bar of the browser, enter **https://localhost:10001/web/test**. A dialog box is displayed.
3. Select the product.
4. Click **Select an event**. The list of all events configured in the **manifest** file is displayed.
5. Select an event. The corresponding payload is displayed. To test a different scenario other than the default, edit the payload.
6. Click **Simulate**. If the event is successfully simulated, the **Simulate** button changes to a **Success** button. If the event simulation fails because of invalid payload data, the **Simulate** button changes to a **Failed** button. Modify the payload appropriately and click **Simulate**.

Scheduled Events

Server Method Invocation

# ERRORS and VALIDATIONS

## [Troubleshooting](#)

## [Lint Validations](#)

## GUIDELINES

### UI Style Guide

The Freshworks CRM UI is built on a modified version of the [Bootstrap v3.3 framework](#). The user interface of apps that are to be published on Freshworks Marketplace need to be consistent with the Freshworks CRM UI. We do not enforce this for custom apps but it is highly recommended for better user experience.

Here are a few guidelines that you need to follow.

- Import the Freshworks CRM app stylesheet in all templates and ensure that the class names of UI elements used in the app match those used by Bootstrap.

#### **template.html**

```
<link rel="stylesheet" type="text/css"
href="https://static.freshdev.io/fdk/2.0/assets/freshworks_crm.cs
s">
```

**Note:** For an omni app, use:

#### **template.html**

```
<link rel="stylesheet" type="text/css"
href="https://static.freshdev.io/fdk/2.0/assets/freshworks.css">
```

- All template files should contain a div block with **class="fs-widget-wrapper"** inside the

body of the template. This is used to implement styles which are applicable to all elements of the app.

```
<body>
<div class="fs-widget-wrapper">
  //template code
</div>
</body>
```

- A few elements and their usage are listed in the following table.

ELEMENT	USAGE
<b>Label</b> - Defines a label for an input element	<code>&lt;span class="label"&gt;Company&lt;/span&gt;</code>
<b>Section Divider</b> - Separates various sections of the app.	<code>&lt;div class="fw-divider"&gt;&lt;/div&gt;</code>
<b>Links</b> - Allows users to move from one page to another. No class needs to be added.	<code>&lt;a href=""&gt;First Opportunity&lt;/a&gt;</code>
<b>List</b> - All divs within the parent are placed one after the other. If you are using this template to display field names and values, use the muted class for field names.	<code>&lt;div class="fw-content-list"&gt;   &lt;div class="muted"&gt;Email&lt;/div&gt;   &lt;div&gt;tom@outerspace.com&lt;/div&gt; &lt;/div&gt;</code>

**Crayons** - To build apps that are consistent with the Freshworks UI, you can use our UI component library - Crayons. Crayons leverages the custom CSS properties to provide a default theme and style for the app's UI components. To include components such as section dividers, links, and lists you can use the FCRM app stylesheet. For more information on how to use the stylesheet, see the UI Style Guide.

[App Content Guidelines](#)

[Code Review Guidelines](#)

# MARKETPLACE LISTING

## Types of Apps

**Note:** To request for a trial Freshworks CRM account to test your app, email us at [marketplace@freshworks.com](mailto:marketplace@freshworks.com).

Apps are classified into the following types:

- **Freshworks Apps:** These apps extend the functionalities of the product's user portal and are built using the Freshworks CLI. These apps can be installed from the Freshworks Marketplace. These apps can be compatible only with Freshworks CRM or with more products. Apps that are compatible with more than one Freshworks product are termed omni apps. For information on how to create an omni app, see the [Quick Start](#) and [App Manifest](#) sections. For information on how to submit a Freshworks app, see [Freshworks Apps](#).

**Note:** Omni apps have the same functionality across all compatible products. At present, the supported products for omni apps are Freshsales and Freshworks CRM.

- **Custom Apps:** These apps address specific customer requirements and are available only to that customer. The custom apps are not subject to review and are available for installation as soon as they are submitted. For information on how to submit a Custom app, see [Custom Apps](#).
- **External Apps:** These apps use the Freshworks CRM methods and integrate the Freshworks CRM functionalities with a third-party product. They are listed in the Freshworks Marketplace, but are installed from the third-party product. For information on how to submit an External app, see [External Apps](#).



# Freshworks Apps

**Note:** Before submitting an app in the Developer portal, to facilitate quick review and publishing, ensure that the app meets the [content](#) and [coding guidelines](#).

To submit a Freshworks app:

1. Navigate to the Freshworks CRM developer website and click **Log in**. If you are signed out of your **Freshworks CRM** account, the login page is displayed.
2. On the login page, enter the Freshworks CRM account credentials and click **SIGN IN**.
3. Click **+Add New App**. The Add New App window is displayed.
4. Click **Freshworks App**. The Freshworks App window is displayed.
5. Click **Upload app file**, select the packed app file, and click **Open**. The My New App page is displayed.
6. Under the **App Details** tab, provide appropriate information by using the following fields.

FIELD	DESCRIPTION
App Compatibility <b>mandatory</b>	To submit an app that is compatible only with Freshworks CRM, select <b>Single Product</b> .  To submit an omni app, select <b>Multi Product</b> . By default, <b>freshsales</b> and <b>freshworksCRM</b> are selected in the <b>For Product</b> field and the selection cannot be modified.
For Product <b>mandatory</b>	Select <b>freshworksCRM</b> .
App Icon <b>mandatory</b>	Upload an icon to represent the app. Ensure that the logo meets the following requirements. <ul style="list-style-type: none"><li>• Image format is PNG.</li><li>• Logo background is transparent.</li><li>• Aspect ratio is 1:1.</li><li>• Minimum resolution is 400 x 400 pixels.</li></ul>

Video Link <b>mandatory</b> if the app is to be listed as a featured app.	Enter the URL to the YouTube or Vimeo video that explains the app's use case. Ensure that the URL is in the following format. <ul style="list-style-type: none"> <li>• YouTube link - <a href="https://youtu.be/VIDEO_ID">https://youtu.be/VIDEO_ID</a></li> <li>• Vimeo link - <a href="https://player.vimeo.com/video/VIDEO_ID">https://player.vimeo.com/video/VIDEO_ID</a></li> </ul> For more information on the guidelines, see <a href="#">Video guidelines</a> .
App Specification/Demo video	Enter the URL to the app's specification document or demo video.
For omni apps, navigate to each product tab and specify the following information for all products.	
App Name <b>mandatory</b>	Enter a name for the app, comprising two to three words.
Overview <b>mandatory</b>	Enter an introduction to the app, not exceeding 100 characters.
Description <b>mandatory</b>	Enter a description of the app's functionality and features.
Instructions	If specific user actions are required when the app is installed, in this field enter the corresponding instructions as a sequential list.
Add Languages	To specify the app information in languages other than English (default language), click on the text area. A list of languages is displayed. Select a language and in the corresponding fields displayed, enter the appropriate values.
Screenshots <b>mandatory</b>	Upload screenshots to illustrate how the app works. Capture different states and process flows to give users a better insight into the app. You can upload a maximum of five screenshots. Ensure that the screenshots meet the following requirements. <ul style="list-style-type: none"> <li>• Image format is PNG.</li> </ul>

	<ul style="list-style-type: none"> <li>• Minimum resolution is 1350 x 850 pixels.</li> </ul>
App Category <b>mandatory</b>	Select the appropriate category to help users filter and find the app in the Marketplace gallery. You can select a maximum of two categories.
Support Email <b>mandatory</b>	Enter the email address that users can use to contact you in case of any queries regarding the app.
Support URL <b>mandatory</b>	Enter the URL to any location where users can find more information about the app.
Alternate Email <b>mandatory</b>	Enter the email address that Freshworks can use to contact you, when the app is under review.

7. To save the app details, click **Save**.
8. Click **Save and Publish**, to submit the app for the approval process.

**Note:** To submit an existing Freshsales app as an omni app that is compatible with FCRM, navigate to your **Freshsales account > My Apps**. From the list of apps displayed, select the app that you want to update as an omni app. The app details are displayed. In the **Development Version** widget, click **Replace file** and upload the latest version of the packed app file.

## [Custom Apps](#)

## [External Apps](#)