**Authentication :**

This is HTTPS-only API. Authentication will be performed based in the email address of the user and their respective API Key.

Email and API key should be passed via HTTP Basic Authentication. Email address as username and their respective API Key as password.

**Endpoints:**

All API requests should be made to: **https://{domain}.agilecrm.com/dev/**

**Note:** All the data is Case-Sensitive. Emails, names and other values are case sensitive. For example, “Test” and “test” are considered as two different words.

**Contacts & Companies API**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Description | Value Type | Read-Only | Mandatory | Accepted values |
| id | Unique id generated when contact is created | integer | Yes | Yes, for update and delete calls. | NA |
| type | Type distinguishes a contact or company. | string | no | No.  Defaults to “PERSON” if not mentioned. | “PERSON” or “COMPANY” |
| tags | Unique identifiers added to contact, for easy management of contacts | list | no | no | a |
| lead\_score | Score of contact | integer | no | no | Any positive integer |
| star\_value | Rating of contact (Max value 5) | short | no | no | 0 to 5 |
| properties | Contact properties are represented by list of JSON objects, each JSON object should follow the prototype shown. Custom fields will have type as CUSTOM and others will have type SYSTEM.  { "type": "SYSTEM/CUSTOM", "name": "<name of the property>",  "value": "<value of the property>"  }, | List of JSON objects | no | first\_name is mandatory |  |

**Example**

**sample jason**

{

"id": 27001,

"type": "PERSON",

"created\_time": 1349974226,

"updated\_time": 1364483386,

"star\_value": 0,

"lead\_score": 0,

"tags": [

"Buyer",

"Deal Closed"

],

"properties": [

{

"type": "SYSTEM",

"name": "first\_name",

"value": "Adam"

},

{

"type": "SYSTEM",

"name": "last\_name",

"value": "Smith"

},

{

"type": "SYSTEM",

"name": "company",

"value": "Invox"

},

{

"type": "SYSTEM",

"name": "title",

"value": "Evangelist"

},

{

"type": "SYSTEM",

"name": "email",

"subtype": "",

"value": "jagan@clickdesk.com"

},

{

"type": "CUSTOM",

"name": "My Custom Field",

"value": "Custom value"

}

],

"widget\_properties": "{\"Twitter\":\"11458852\"}",

"domainUser": {

"id": 314001,

"domain": "mycompany",

"email": "crm-user@mycompany.com",

"is\_admin": true,

"is\_account\_owner": false,

"is\_disabled": false,

"name": "angel",

"password": "PASSWORD",

"info\_json\_string": "{\"logged\_in\_time\":1364879569,\"created\_time\":1361958281}",

"hashedString": "81dc9bdb52d04dc20036dbd8313ed055"

}

}

**Listing Contacts :**

**dev/api/contacts**

Method: GET

Returns list of contacts in domain, which are ordered by creation time

For the Response in the XML format, add the header Accept as application/xml. By default, the response will be in XML format.

Using curl

curl https://{domain}.agilecrm.com/dev/api/contacts -H “Accept : application/xml” -v -u {email}:{apikey}

Example XML response

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<contacts>

<contact>

<id>40002</id>

<type>PERSON</type>

<created\_time>1356589886</created\_time>

<updated\_time>1357107323</updated\_time>

<star\_value>0</star\_value>

<lead\_score>93</lead\_score>

<tags>sales</tags>

<tags>bruik</tags>

<tags>walt</tags>

<tags>los</tags>

<tags>vo</tags>

<tags>steady</tags>

<properties>

<type>SYSTEM</type>

<name>first\_name</name>

<value>Los </value>

</properties>

<properties>

<type>SYSTEM</type>

<name>last\_name</name>

<value>Bruikheilmer</value>

</properties>

<properties>

<type>SYSTEM</type>

<name>company</name>

<value>steady.inc</value>

</properties>

<properties>

<type>SYSTEM</type>

<name>title</name>

<value>VP Sales</value>

</properties>

<properties>

<type>SYSTEM</type>

<name>email</name>

<subtype>work<subtype/>

<value>bruik@walt.inc</value>

</properties>

<properties>

<type>SYSTEM</type>

<name>address</name>

<value>{"address":"","city":"","state":"","zip":"","country":""}</value>

</properties>

<domainUser>

<id>106010</id>

<domain>yaswanth</domain>

<email>praveen@invox.com</email>

<is\_admin>true</is\_admin>

<is\_account\_owner>true</is\_account\_owner>

<is\_disabled>false</is\_disabled>

<name>yaswanth</name>

<password>PASSWORD</password>

<info\_json\_string>

{"logged\_in\_time":1359434638,"created\_time":1358831084}

</info\_json\_string>

</domainUser>

</contact>

</contacts>

For the Response in the JSON format, add the header Accept as application/json.

Using curl

curl https://{domain}.agilecrm.com/dev/api/contacts -H “Accept : application/json”

-v -u {email}:{apikey}

Example JSON response

[

{

"count": "9",

"id": "40002",

"type": "PERSON",

"created\_time": "1356589886",

"updated\_time": "1357107323",

"star\_value": "0",

"lead\_score": "93",

"tags": [

"sales", "bruik", "walt", "los", "vo", "steady"

],

"properties": [

{

"type": "SYSTEM",

"name": "first\_name",

"value": "Los "

},

{

"type": "SYSTEM",

"name": "last\_name",

"value": "Bruikheilmer"

},

{

"type": "SYSTEM",

"name": "company",

"value": "steady.inc"

},

{

"type": "SYSTEM",

"name": "title",

"value": "VP Sales"

},

{

"type": "SYSTEM",

"name": "email",

"subtype": "work",

"value": "bruik@walt.inc"

},

{

"type": "SYSTEM",

"name": "address",

"value": "{\"address\":\"\",\"city\":\"\",\"state\":\"\",\"zip\":\"\",\"country\":\"\"}"

}

],

"domainUser": {

"id": "106010",

"domain": "yaswanth",

"email": "praveen@invox.com",

"is\_admin": "true",

"is\_account\_owner": "true",

"is\_disabled": "false",

"name": "yaswanth",

"password": "PASSWORD",

"info\_json\_string": "{\"logged\_in\_time\":1359434638,\"created\_time\":1358831084}"

}

},

{

"id": "41002",

"type": "PERSON",

"created\_time": "1356590542",

"updated\_time": "1359367255",

"star\_value": "4",

"lead\_score": "92",

"tags": [

"president",

"john",

"doe",

"cruzzetech"

],

"properties": [

{

"type": "SYSTEM",

"name": "first\_name",

"value": "John"

},

{

"type": "SYSTEM",

"name": "last\_name",

"value": "Doe"

},

{

"type": "SYSTEM",

"name": "company",

"value": "cruzzetech.com"

},

{

"type": "SYSTEM",

"name": "email",

"value": "john@cruzzetech.com"

},

{

"type": "SYSTEM",

"name": "title",

"value": "President"

}

],

"tags\_with\_time\_json": "{\"president\":1359365229885,\"john\":1359365229885,\"doe\":1359365229885,\"cruzzetech\":1359365229885}",

"domainUser": {

"id": "106010",

"domain": "yaswanth",

"email": "praveen@invox.com",

"is\_admin": "true",

"is\_account\_owner": "true",

"is\_disabled": "false",

"name": "yaswanth",

"password": "PASSWORD",

"info\_json\_string": "{\"logged\_in\_time\":1359434638,\"created\_time\":1358831084}"

}

}

]

**Other available Responses:**

Status 200: Gives the above JSON object in above format.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Get contact by id**

**dev/api/contacts/{id}**

Method: GET

Returns contact object which is associated given id

Using curl

curl https://{domain}.agilecrm.com/dev/api/contacts/{id} -H “Accept :application/xml”

-v -u {email}:{apikey}

Example Response

Returns created contact object with all parameters in it as mentioned in the above example.

**Other available Responses:**

Status 200: Gives the above JSON object in above format.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Creating a contact**

**dev/api/contacts**

Method: POST

Accepts contact JSON as post data along with the credentials of domain User (User name and API Key).

Acceptable request Representation:

{

"star\_value": "4",

"lead\_score": "92",

"tags": ["Lead","Likely Buyer","You\_can-user $special chars"],

"properties": [

{

"type": "SYSTEM",

"name": "first\_name",

"value": "Los "

},

{

"type": "SYSTEM",

"name": "last\_name",

"value": "Bruikheilmer"

},

{

"type": "SYSTEM",

"name": "company",

"value": "steady.inc"

},

{

"type": "SYSTEM",

"name": "title",

"value": "VP Sales"

},

{

"type": "SYSTEM",

"name": "email",

"subtype": "work",

"value": "bruik@walt.inc"

},

{

"type": "SYSTEM",

"name": "address",

"value": "{\"address\":\"\",\"city\":\"\",\"state\":\"\",\"zip\":\"\",\"country\":\"\"}"

},

{

"type": "CUSTOM",

"name": "My Custom Field",

"value": "Custom value"

}

]

}

**Response:**

Status 200: Contact added successfully. Returns the newly added contact object in the response.

Status 401: Unauthorised. When the user name and password fields are wrong.

Status 400: If the input is in wrong format.

Status 406: If the Limit of the contacts exceeded.

**Updating contact**

**dev/api/contacts**

**Method:** PUT

Accepts contact object with valid id parameter in it, where ‘id’ refers the contact to be updated.

Acceptable request Representation:

{

“id”:8573489754593

"star\_value": "4",

"lead\_score": "92",

"tags": ["Lead","Likely Buyer","You\_can-user $special chars"],

"properties": [

{

"type": "SYSTEM",

"name": "first\_name",

"value": "Los "

},

{

"type": "SYSTEM",

"name": "last\_name",

"value": "Bruikheilmer"

},

{

"type": "SYSTEM",

"name": "company",

"value": "steady.inc"

},

{

"type": "SYSTEM",

"name": "title",

"value": "VP Sales"

},

{

"type": "SYSTEM",

"name": "email",

"subtype": "work",

"value": "bruik@walt.inc"

},

{

"type": "SYSTEM",

"name": "address",

"value": "{\"address\":\"\",\"city\":\"\",\"state\":\"\",\"zip\":\"\",\"country\":\"\"}"

},

{

"type": "CUSTOM",

"name": "My Custom Field",

"value": "Custom value"

}

]

}

If there is no ID, it will considered as a new contact.

**Response:**

Status 200: Contact updated successfully. Returns the updated contact object in the response.

Status 401: Unauthorised. When the user name and password fields are wrong.

Status 400: If the input is in wrong format.

**Delete single contact**

**dev/api/contacts/{id}**

Method: DELETE

Using curl

curl https://{domain}.agilecrm.com/dev/api/contacts/{id} \

-v -u {email}:{apikey} -X DELETE

Deletes contact based on the id of the contact, which is sent in request url path.

**Response:**

Status 204: Contact deleted successfully.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Search Contact by Email**

**dev/api/contacts/search/email**

Method: POST

Searches for the contact with given email address. Email address should be sent as a form parameter(**Content-Type**: application/x-www-form-urlencoded ). We can search for multiple contacts using their Email-Id’s.

Using curl

curl https://{domain}.agilecrm.com/dev/api/contacts/search/email -H "Accept: application/xml"

-H “Content-Type :application/x-www-form-urlencoded”

-d ‘**email\_ids**=["notifications@basecamp.com"]’

-v -u {email}:{apikey} -X POST

**Response:**

Status 200: Gives the Contact as JSON object in above format. If email doesn’t match, it will return an empty object.

Status 401: Unauthorised. When the user name and password fields are wrong.

Status 400: If the email is in wrong format.

**Adding Tags to a contact based on Email:**

**dev/api/contacts/email/tags/add**

Method: POST

Searches for the contact based on the given email address and add the given tags to the contact. You can add multiple tags. Tags should be sent as a array. Email address (email) and tags (tags) array should be sent as a form parameter(**Content-Type**: application/x-www-form-urlencoded )

Using curl

curl https://{domain}.agilecrm.com/dev/api/contacts/email/tags/add -H "Accept: application/xml"

-H “Content-Type :application/x-www-form-urlencoded”

-d ‘**email**=notifications@basecamp.com&**tags**=[“testing”]’

-v -u {email}:{apikey} -X POST

**Response:**

Status 204: Tags added successfully.

Status 401: Unauthorised. When the user name and password fields are wrong.

Status 400: If the input is in wrong format.

**Delete Tags to a contact based on Email:**

**dev/api/contacts/email/tags/delete**

Method: POST

Searches for the contact based on the given email address and search for the given tag in the contact tags list. If the there is, then delete that tag. You can delete multiple tags. Tags should be sent as a array. Email address (email) and tags (tags) array should be sent as a form parameter(**Content-Type**: application/x-www-form-urlencoded )

Using curl

curl https://{domain}.agilecrm.com/dev/api/contacts/email/tags/delete -H "Accept: application/xml"

-H “Content-Type :application/x-www-form-urlencoded”

-d ‘**email**=notifications@basecamp.com&**tags**=[“testing”]’

-v -u {email}:{apikey} -X POST

**Response:**

Status 204: Tags deleted successfully.

Status 401: Unauthorised. When the user name and password fields are wrong.

Status 400: If the input is in wrong format.

**Add Score to a Contact using Email-ID:**

**dev/api/contacts/add-score**

Method: POST

It is used to change the score of the contact by using the email address. If we want to decrease a quantity from the existing score, then use negative values for the score parameter.

Using curl

curl https://{domain}.agilecrm.com/dev/api/contacts/add-score -H "Accept: application/xml" -H “Content-Type :application/x-www-form-urlencoded”

-d ‘**email**=[notifications@basecamp.com](mailto:notifications@basecamp.com)&**score**=5’ -v -u {email}:{apikey} -X POST

**Response:**

Status 200: Score changed successfully and returns the Contact object.

Status 401: Unauthorised. When the user name and password fields are wrong.

Status 400: If the input is in wrong format

**Get Tasks related to Contact:**

**dev/api/contacts/{contact\_id}/tasks/sort**

Method: GET

Retrieve the tasks related to contact sorted on the date (Latest first.).

Using curl

curl https://{domain}.agilecrm.com/dev/api/contacts/{contact\_id}/tasks/sort -H "Accept: application/xml" -v -u {email}:{apikey}

**Response:**

Status 200: Returns the task list related to the contact.

[

{

"id": 5152971570544640,

"type": "EMAIL",

"priority\_type": "NORMAL",

"due": 1409596200,

"created\_time": 1409567774,

"is\_complete": false,

"contacts": [

{

"id": 5716606839685120,

"type": "PERSON",

"created\_time": 1403057154,

"updated\_time": 1403057154,

"viewed\_time": 0,

"viewed": {

"viewed\_time": 1409567737463,

"viewer\_id": 5345980119515136

},

"star\_value": 0,

"lead\_score": 0,

"tags": [],

"tagsWithTime": [],

"properties": [

{

"type": "SYSTEM",

"name": "first\_name",

"value": "Basecamp (2Desk)"

},

{

"type": "SYSTEM",

"name": "last\_name",

"value": "Basecamp (2Desk)"

},

{

"type": "SYSTEM",

"name": "email",

"value": "notifications@basecamp.com"

}

],

"campaignStatus": [],

"entity\_type": "contact\_entity",

"unsubscribeStatus": [],

"emailBounceStatus": [],

"owner": {

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

"is\_admin": true,

"is\_account\_owner": true,

"is\_disabled": false,

"name": "prabath kolipaka"

}

}

],

"subject": "Contact created in agile crm",

"entity\_type": "task",

"notes": [],

"progress": 0,

"status": "YET\_TO\_START",

"ownerPic": "https://s3.amazonaws.com/agilecrm/panel/uploaded-logo/1401427105618?id=upload-container",

"taskOwner": {

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

"is\_admin": true,

"is\_account\_owner": true,

"is\_disabled": false,

"name": "prabath kolipaka"

}

}

]

Status 401: Unauthorised. When the user name and password fields are wrong.

**Deals API**

**Fields**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Description | Value Type | Read Only | Mandatory | Accepted Values |
| id | Unique id is assigned to every deal, when it is created | integer | no | Yes, if request is to update an existing deal | N/A |
| name | Name of the deal | string | no | Yes | N/A |
| description | Brief description about deal. | string | no | No | N/A |
| expected\_value | Estimated value of a deal. | long | no | Yes | Max up to 1000000000000 |
| pipeline\_id | Track id of the deal. | long | no | Yes | There should be a track with this Id. |
| milestone | Milestone of deal (won, progress, lost..). These are depending up on the track selected. Milestones may differ based on the tracks. | String | no | Yes | MIlestone in the above mentioned track. |
| probability |  | integer | no | should be ranging between 0-100 | 0 to 100 |
| close\_date | close date of deal | long | no | Yes | Epoch time |
| created\_time | Created time of the deal. Returns time in seconds | long | yes | No, generated automatically on deal creation | Epoch time |
| owner\_id | owner\_id represents id of domain user | string | No, write access is provided to set owner of deal | Yes | The id of the user. |
| prefs |  | JSON string |  |  | N/A |
| contacts | Relates list of contacts to deal | List of contact. | Yes | No |  |
| contact\_ids | Relates list of contacts to deal. Should be sent in the request while creating and updating contact. | List of contact id.  ex: [“122”, 145,201] | Write access to relate contacts, while returning contact jsons respective to ids set | No | IDs of contacts. |

**Deal JSON Example**

{

"id" : "822",

"name" : "Game development",

"description" : "Visual 3D based Online Game.",

"expected\_value" : "50000",

"pipeline\_id" : "43535822",

"milestone" : "progress",

"probability ": "70",

"close\_date" : "1349980200",

"created\_time" : "1356325843",

"entity\_type" : "deal",

"Prefs": {

"id" : "984",

"template" : "blue",

"currency" : "USD",

"signature" : "- Powered by AgileCRM",

"task\_reminder" : "true"

}

}

**Listing deals**

**dev/api/opportunity**

Method: GET

Returns list of all “Deal” in the domain in JSON format, which are ordered on created time

**Using curl**

curl https:{domain}.agilecrm.com/dev/api/opportunity -H "Accept:application/json" -v -u {email}:{API Key}

**Example response**:

[

{

"id": 981,

"name": "premium subscriptions",

"contacts": [

{

"id": 2363,

"type": "PERSON",

"created\_time": 1359033520,

"updated\_time": 1359033533,

"star\_value": 0,

"lead\_score": 0,

"tags": [],

"properties": [

{

"type": "SYSTEM",

"name": "first\_name",

"value": "John"

},

{

"type": "SYSTEM",

"name": "last\_name",

"value": "Olsen"

},

{

"type": "SYSTEM",

"name": "email",

"value": "john@example.com.com"

}

],

"domainUser": {

"id": 516,

"email": "test@example.com",

"is\_admin": true,

"is\_account\_owner": false,

"is\_disabled": false,

"name": "ram"

}

}

],

"description": "Deal premium plan subscription",

"expected\_value": 300,

"pipeline\_id" : "43535822",

"milestone": "Open",

"probability": 46,

"close\_date": 1371753000,

"created\_time": 1360503631,

"entity\_type": "deal",

"owner": {

"id": 516,

"email": "test@example.com",

"is\_admin": true,

"is\_account\_owner": false,

"is\_disabled": false,

"name": "Test"

},

"prefs": {

"id": 984,

"pic": "",

"template": "blue",

"width": "",

"currency": "USD",

"signature": "- Powered by AgileCRM",

"task\_reminder": true,

"currentDomainUserName": "?"

},

"pic": ""

},

{

"id": 981,

"name": "premium subscriptions",

"contacts": []

}

]

**Other Response Statuses:**

Status 401: Unauthorised. When the user name and password fields are wrong.

**Get deal by its id**

**dev/api/opportunity/{id}**

Method: GET

**Using curl :**

curl https:{domain}.agilecrm.com/dev/api/opportunity/981 -H "Accept : application/json" -v -u {email}:{API Key}

**Example Response :**

{

"id": 981,

"name": "500 premium subscriptions",

"contacts": [

{

"id": 2363,

"type": "PERSON",

"created\_time": 1359033520,

"updated\_time": 1359033533,

"star\_value": 0,

"lead\_score": 0,

"tags": [

],

"properties": [

{

"type": "SYSTEM",

"name": "first\_name",

"value": "John"

},

{

"type": "SYSTEM",

"name": "last\_name",

"value": "Olsen"

},

{

"type": "SYSTEM",

"name": "email",

"value": "john@example.com"

},

],

"domainUser": {

"id": 516,

"email": "test@example.com",

"is\_admin": true,

"is\_account\_owner": false,

"is\_disabled": false,

"name": "ram",

"password": "PASSWORD",

"info\_json\_string": "{\"logged\_in\_time\":1360503513,\"created\_time\":1353507148}"

}

}

],

"description": "Deal premium plan subscription",

"expected\_value": 300,

"pipeline\_id" :"43535822",

"milestone": "Open",

"probability": 46,

"close\_date": 1371753000,

"created\_time": 1360503631,

"entity\_type": "deal",

"owner": {

"id": 516,

"email": "test@example.com",

"is\_admin": true,

"is\_account\_owner": false,

"is\_disabled": false,

"name": "ram",

"password": "PASSWORD",

"info\_json\_string": "{\"logged\_in\_time\":1360503513,\"created\_time\":1353507148}"

},

"prefs": {

"id": 984,

"pic": "",

"template": "blue",

"width": "",

"currency": "USD",

"signature": "- Powered by AgileCRM",

"task\_reminder": true,

"currentDomainUserName": "?"

},

"pic": ""

}

**Other Response Statuses:**

Status 401: Unauthorised. When the user name and password fields are wrong.

**Create Deal**

**dev/api/opportunity**

Method: POST

Accepts deal JSON as data in post request to the url specified above, which creates new deal and returns the deal JSON with id field generated when new deal is created. If post data includes valid deal id, respective deal is updated with the data sent in request.

Milestones name should be same as the the one in the website and it is case sensitive. If the milestone name is given wrong even the case, it will not be shown in the milestone view.

**Using curl**

curl https://{domain}.agilecrm.com/dev/api/opportunity -H "Content-Type: application/json" -d ‘{\“name\” : \“deal\”, \”contact\_ids\” : [\“2358\”, \”2356\”] , \”owner\_id\” : \“516\”, \“pipeline\_id\” :\”43535822\”, \”milestone\” : \“milestone\”}’ 000-v -u test@example.com:4uet78u6atfn38m9dounnq9g4u -X POST

**Response Statuses:**

Status 200: Deal added successfully and return the new created deal as JSON in response.

Status 401: Unauthorised. When the user name and password fields are wrong.

Status 400: If the input is in wrong format

**Update Deal**

**dev/api/opportunity**

Method: PUT

Accepts Deal JSON. Id parameter of the deal should be specified, it indicates which deal to be updated with the new data sent. Milestones name should be same as the the one in the website and it is case sensitive. If the milestone name is given wrong even the case, it will not be shown in the milestone view.

**Using curl**

curl https://{domain}.agilecrm.com/dev/api/opportunity -H "Content-Type: application/json" -d "{\"name\" : \"deal\", \”contact\_ids\” : [\“2358\”,\”2356\”] , \”owner\_id\” : \“516\”, \“pipeline\_id\” :\”43535822\”, \”milestone\” : \“milestone\”}" -v -u test@example.com:4uet78u6atfn38m9dounnq9g4u -X PUT

**Response Statuses:**

Status 200: Deal updated successfully and return the updated deal as JSON in response.

Status 401: Unauthorised. When the user name and password fields are wrong.

Status 400: If the input is in wrong format

**Create Deal to a contact using Email Id**

**dev/api/opportunity/email/{email}**

Method: POST

Accepts deal JSON as data in post request to the url specified above, which creates new deal and returns the deal JSON with id field generated when new deal is created. If post data includes valid deal id, respective deal is updated with the data sent in request.

Milestones name should be same as the the one in the website and it is case sensitive. If the milestone name is given wrong even the case, it will not be shown in the milestone view.

**Using curl**

curl https://{domain}.agilecrm.com/dev/api/opportunity/test@agilecrm.com -H "Content-Type: application/json" -d ‘{"name" : "deal", ”owner\_id” : “516”, \“pipeline\_id\” :\”43535822\”, \”milestone\” : \“milestone\”}’ 000-v -u test@example.com:4uet78u6atfn38m9dounnq9g4u -X POST

Creates a Deals for the contact with the given email address. If there is no contact, it will not create any deal.

**Response Statuses:**

Status 200: Deal added successfully and return the new created deal as JSON in response.

Status 401: Unauthorised. When the user name and password fields are wrong.

Status 400: If the input is in wrong format

**Delete Deal**

**dev/api/opportunity/{id}**

Method: DELETE

Deletes the deal based the id specified in the url.

**Using curl**

curl https://{domain}.agilecrm.com/dev/api/opportunity/{id} -H "Content-Type: application/json" -v -u test@example.com:4uet78u6atfn38m9dounnq9g4u -X DELETE

**Response Statuses:**

Status 200: Deal deleted successfully.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Bulk Delete**

**dev/api/opportunity/bulk**

Method: POST

Accepts list of deal ids in post request, add deletes all deals based on list of ids sent.

**Using curl :**

curl https://{domain}.agilecrm.com/dev/api/opportunity/bulk -H "Content-Type: application/x-www-form-urlencoded" -d ‘{“ids” : [“123”, ”234”, ”235”]}’ -v -u {email}:{API Key} -X POST

**Response Statuses:**

Status 200: Deal deleted successfully.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Get Deals from Default track separated based milestones:**

**dev/api/opportunity/byMilestone**

Method: GET

Fetches list of deals from the default track separated by the milestones. Number of milestones is depended upon the milestones added by user to default track.

**Using curl :**

curl https://{domain}.agilecrm.com/dev/api/opportunity/byMilestone -H “Accept : application/json” -v -u {email} : {API Key}

Example Response :

{

"Lost": [

{

"cursor": "",

"count": "",

"id": 5086100271923200,

"name": " Nike Endorsement",

"contact\_ids": [

],

"custom\_data": [

{

"name": "Ddate",

"value": "07/11/2014"

}

],

"description": "Advertisements and apparel for Nike",

"expected\_value": 2000000.0,

"pipeline\_id" :"43535822",

"milestone": "Lost",

"probability": 98,

"close\_date": 1468800,

"owner\_id": "",

"created\_time": 1405085631,

"track": "",

"entity\_type": "deal",

"notes": [

],

"note\_description": "",

"owner": {

"cursor": "",

"count": "",

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

"is\_admin": true,

"is\_account\_owner": true,

"is\_disabled": false,

"email\_template": "",

"name": "prabath kolipaka"

},

"contacts": [

],

"pic": "https://s3.amazonaws.com/agilecrm/panel/uploaded-logo/1401427105618?id=upload-container"

}

],

Open: [

{

"cursor": "",

"count": "",

"id": 5115435166990336,

"name": " test deal",

"contact\_ids": [

],

"custom\_data": [

{

"name": "Ddate",

"value": "07/23/2014"

}

],

"description": "testing",

"expected\_value": 1235.0,

"pipeline\_id" :"43535822",

"milestone": "Lost",

"probability": 123,

"close\_date": 1405555200,

"owner\_id": "",

"created\_time": 1405518785,

"track": "",

"entity\_type": "deal",

"notes": [

],

"note\_description": "",

"owner": {

"cursor": "",

"count": "",

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

"is\_admin": true,

"is\_account\_owner": true,

"is\_disabled": false,

"email\_template": "",

"name": "prabath kolipaka"

},

"contacts": [

],

"pic": "https://s3.amazonaws.com/agilecrm/panel/uploaded-logo/1401427105618?id=upload-container"

}

],

Won: [

{

"cursor": "",

"count": "",

"id": 5123586478047232,

"name": " deal test",

"contact\_ids": [

],

"custom\_data": [

{

"name": "Ddate",

"value": "07/21/2014"

}

],

"description": "",

"expected\_value": 1.0E10,

"pipeline\_id" :"43535822",

"milestone": "Lost",

"probability": 10,

"close\_date": 1405036800,

"owner\_id": "",

"created\_time": 1405575543,

"track": "",

"entity\_type": "deal",

"notes": [

],

"note\_description": "",

"owner": {

"cursor": "",

"count": "",

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

"is\_admin": true,

"is\_account\_owner": true,

"is\_disabled": false,

"email\_template": "",

"name": "prabath kolipaka"

},

"contacts": [

],

"pic": "https://s3.amazonaws.com/agilecrm/panel/uploaded-logo/1401427105618?id=upload-container"

}

],

Stage1: [

{

"cursor": "",

"count": "",

"id": 5193799529660416,

"name": " Nike Endorsement",

"contact\_ids": [

],

"custom\_data": [

],

"description": "Advertisements and apparel for Nike",

"expected\_value": 2.0E11,

"pipeline\_id" :"43535822",

"milestone": "Lost",

"probability": 98,

"close\_date": 0,

"owner\_id": "",

"created\_time": 1402580362,

"track": "",

"entity\_type": "deal",

"notes": [

],

"note\_description": "",

"owner": {

"cursor": "",

"count": "",

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

"is\_admin": true,

"is\_account\_owner": true,

"is\_disabled": false,

"email\_template": "",

"name": "prabath kolipaka"

},

"contacts": [

],

"pic": "https://s3.amazonaws.com/agilecrm/panel/uploaded-logo/1401427105618?id=upload-container"

}

]

}

In this above example Lost, Open, Won, Stage 1 are different milestone. These may vary depending on the user choice.

**Response Statuses:**

Status 200: Successfully retrieved the deals list.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Get Deals for a particular Track (grouped on Milestone).**

**dev/api/opportunity/byPipeline/based**

Method: GET

Fetches list of deals from the particular track separated by the milestones. Number of milestones is depended upon the milestones added by user to that track. We need to send the track id as a query parameter.

If you sent the parameter as 0, it will give the list of deals from the default track.

**Using curl :**

curl https://{domain}.agilecrm.com/dev/api/opportunity/byPipeline/based?pipeline\_id=4756437 -H “Accept : application/json” -v -u {email} : {API Key}

Example Response :

{

"Lost": [

{

"cursor": "",

"count": "",

"id": 5086100271923200,

"name": " Nike Endorsement",

"contact\_ids": [

],

"custom\_data": [

{

"name": "Ddate",

"value": "07/11/2014"

}

],

"description": "Advertisements and apparel for Nike",

"expected\_value": 2000000.0,

"pipeline\_id" :"4756437",

"milestone": "Lost",

"probability": 98,

"close\_date": 1468800,

"owner\_id": "",

"created\_time": 1405085631,

"track": "",

"entity\_type": "deal",

"notes": [

],

"note\_description": "",

"owner": {

"cursor": "",

"count": "",

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

"is\_admin": true,

"is\_account\_owner": true,

"is\_disabled": false,

"email\_template": "",

"name": "prabath kolipaka"

},

"contacts": [

],

"pic": "https://s3.amazonaws.com/agilecrm/panel/uploaded-logo/1401427105618?id=upload-container"

}

],

Open: [

{

"cursor": "",

"count": "",

"id": 5115435166990336,

"name": " test deal",

"contact\_ids": [

],

"custom\_data": [

{

"name": "Ddate",

"value": "07/23/2014"

}

],

"description": "testing",

"expected\_value": 1235.0,

"pipeline\_id" :"4756437",

"milestone": "Lost",

"probability": 123,

"close\_date": 1405555200,

"owner\_id": "",

"created\_time": 1405518785,

"track": "",

"entity\_type": "deal",

"notes": [

],

"note\_description": "",

"owner": {

"cursor": "",

"count": "",

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

"is\_admin": true,

"is\_account\_owner": true,

"is\_disabled": false,

"email\_template": "",

"name": "prabath kolipaka"

},

"contacts": [

],

"pic": "https://s3.amazonaws.com/agilecrm/panel/uploaded-logo/1401427105618?id=upload-container"

}

]

}

In this above example Lost, Open, Won, Stage 1 are different milestone. These may vary depending on the user choice.

**Response Statuses:**

Status 200: Successfully retrieved the deals list.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Get Deals from particular track:**

**dev/api/opportunity/based**

Method: GET

Fetches list of deals from the particular track. Number of milestones is depended upon the milestones added by user to that track. We need to send the track id as a query parameter.

If you sent the parameter as 0, it will give the list of deals from the default track.

If you sent the parameter as 1, it will give the list of all deals from all tracks.

**Using curl :**

curl https://{domain}.agilecrm.com/dev/api/opportunity/based?pipeline\_id=4756437 -H “Accept : application/json” -v -u {email} : {API Key}

Example Response :

[

{

"cursor": "",

"count": "",

"id": 5086100271923200,

"name": " Nike Endorsement",

"contact\_ids": [

],

"custom\_data": [

{

"name": "Ddate",

"value": "07/11/2014"

}

],

"description": "Advertisements and apparel for Nike",

"expected\_value": 2000000.0,

"pipeline\_id" :"4756437",

"milestone": "Lost",

"probability": 98,

"close\_date": 1468800,

"owner\_id": "",

"created\_time": 1405085631,

"track": "",

"entity\_type": "deal",

"notes": [

],

"note\_description": "",

"owner": {

"cursor": "",

"count": "",

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

"is\_admin": true,

"is\_account\_owner": true,

"is\_disabled": false,

"email\_template": "",

"name": "prabath kolipaka"

},

"contacts": [

],

"pic": "https://s3.amazonaws.com/agilecrm/panel/uploaded-logo/1401427105618?id=upload-container"

},

{

"cursor": "",

"count": "",

"id": 5115435166990336,

"name": " test deal",

"contact\_ids": [

],

"custom\_data": [

{

"name": "Ddate",

"value": "07/23/2014"

}

],

"description": "testing",

"expected\_value": 1235.0,

"pipeline\_id" :"4756437",

"milestone": "Lost",

"probability": 123,

"close\_date": 1405555200,

"owner\_id": "",

"created\_time": 1405518785,

"track": "",

"entity\_type": "deal",

"notes": [

],

"note\_description": "",

"owner": {

"cursor": "",

"count": "",

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

"is\_admin": true,

"is\_account\_owner": true,

"is\_disabled": false,

"email\_template": "",

"name": "prabath kolipaka"

},

"contacts": [

],

"pic": "https://s3.amazonaws.com/agilecrm/panel/uploaded-logo/1401427105618?id=upload-container"

}

]

In this above example Lost, Open, Won, Stage 1 are different milestone. These may vary depending on the user choice.

**Response Statuses:**

Status 200: Successfully retrieved the deals list.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Get Deals Related To Specific Contact**

**dev/api/contacts/{id}/deals**

Method: GET

Fetches list of deals related to specified contact using the contact id.

**Using curl :**

curl https://{domain}.agilecrm.com/dev/api/contacts/{contact\_id}/deals -H “Accept : application/json” -v -u {email} : {API Key}

Example Response :

[

{

“id” : 822,

“name” : “Game Development”,

“created\_time” : 1356325843,

“contacts” : [

{

id : “445”,

created\_time : 1356315843

…

}

...

]

…

},

{

“id” : 850,

“name” : “Product Design”,

“created\_time” : “1357794282”,

…

},

...

]

**Response Statuses:**

Status 200: Successfully retrieved the deals list.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Get Deals of Current user (my deals)**

**dev/api/opportunity/my/deals**

Method: GET

Fetches list of deals of the current user

**Using curl :**

curl https://{domain}.agilecrm.com/dev/api/opportunity/my/deals -H “Accept : application/json” -v -u {email} : {API Key}

Example Response :

[

{

“id” : 822,

“name” : “Game Development”,

“created\_time” : 1356325843,

“contacts” : [

{

id : “445”,

created\_time : 1356315843

…

}

...

]

…

},

{

“id” : 850,

“name” : “Product Design”,

“created\_time” : “1357794282”,

…

},

...

]

**Response Statuses:**

Status 200: Successfully retrieved the deals list.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Notes API**

**Fields :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Description | Value Type | Read Only | Mandatory | Accepted values |
| id | Unique id generated when note is created | Long | no | no | N/A |
| created\_time | Creation time of note | Long | yes | no | epoch time |
| subject | Subject of the note | String | no | yes | N/A |
| description | Description of note | String | no | yes | N/A |
| contact\_json | list of contact ids | String array | no | no | N/A |

**Create a note and relate to contacts :**

**/dev/api/notes**

Method: POST

Creates a note and relates it to contacts, which are sent in the note JSON contact field.

**Using curl :**

curl https://{domain}.agilecrm.com/dev/api/notes/ -H “Content-Type : application/json” -H “Accept : application/json” -d ‘{”subject” : ” Note subject” , ”description” : ”Note description”, ”contact\_ids” : [”721001”, ”722001”] }’ -v -u {email} : {APi Key} -X POST

**Example response :**

{

“id” : 80001

“created\_time” : 1360561958,

“subject” : “Note subject”,

“description” : “Note description”,

“contacts” : [“721002”],

“entity\_type” : “note”

}

**Response Statuses:**

Status 200: Note added successfully and return the newly created note as JSON in response.

Status 401: Unauthorised. When the user name and password fields are wrong.

Status 400: If the input is in wrong format

**Add Note to a Contact using Email-ID:**

**dev/api/contacts/email/note/add**

Method: POST

Add a note to the contact with given Email-ID. We need to email address and the note object as a url encoded form parameters to the above given URL.

Using curl

curl https://{domain}.agilecrm.com/dev/api/contacts/email/note/add -H "Accept: application/xml"

-H “Content-Type :application/x-www-form-urlencoded”

-d ‘**email**=[notifications@basecamp.com](mailto:notifications@basecamp.com)&**note**={“subject”:”test”,”description”:”testing description”}’ -v -u {email}:{apikey} -X POST

**Response:**

Status 204: Note added successfully.

Status 401: Unauthorised. When the user name and password fields are wrong.

Status 400: If the input is in wrong format.

**Gets notes related to specific contact :**

**/dev/api/contacts/{contact\_id}/notes**

**Method:** GET

Returns list of note JSONs related to the contact.

**Using curl :**

curl https://{domain}.agilecrm.com/dev/api/contacts/710002/notes -H “Accept : application/json” -v -u {email} : {API Key}

**Example response :**

[

{

“id” : 79001,

“created\_time” : 1360561958,

“subject” : “Note subject1”,

“description” : “Note description1”,

“contacts” : [“721002”],

“entity\_type” : “note”

},

{

“id” : 80001

“created\_time” : 1360561722,

“subject” : “Note subject2”,

“description” : “Note description2”,

“contacts” : [“721002”],

“entity\_type” : “note”

}

]

**Delete a specific note from specific contact :**

**/dev/api/contacts/{contact\_id}/notes/{note\_id}**

**Method:** DELETE

Delete the note of the specific contact. It means, it will remove the relationaship between the note and the contact.

**Using curl :**

curl https://{domain}.agilecrm.com/dev/api/contacts/{contact\_id}/notes/{note\_id} -H “Accept : application/json” -v -u {email} : {API Key} -X DELETE

**Response:**

Status 204: Note removed successfully.

Status 401: Unauthorised. When the user name and password fields are wrong.

Status 400: If the input is in wrong format

**Tasks API**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Description | Value Type | Read Only | Mandatory | Accepted values |
| id | Unique id generated when task is created | Long | no | no | N/A |
| type | Determines the type to which the task belongs. The type should be one in the given list. It is case sensitive. Should be specified as above only. | String | no | yes | CALL, EMAIL, FOLLOW\_UP, MEETING, MILESTONE, SEND, TWEET, OTHER. |
| priority\_type | Determines the priority of the task. It is case sensitive. Should be specified as above only. | String | no | yes | There are 3 types of priorities HIGH, NORMAL, LOW. |
| due | Due date of the task. It should be specified as Epoch Time. | Long | no | yes | Epoch time |
| created\_time | Creation time of task | Long | yes | no | Epoch time |
| is\_complete | Determines whether the task is completed or not. If completed then true or else false. | Boolean | no | no | true/false |
| contacts | List of contacts related to the task. While saving we need to give the contact ids as a string array to relate them to the task. | String array | no | no | ID’s of the contacts. |
| subject | Subject of the task. | String | no | yes | N/A |
| entity\_type | It is a read only variable. Its value is ‘task’ always. | String | yes | no | task |
| notes | It is the list of notes related to this task. While adding a task or editing a task, we need to send the list of note id’s as a string array in order to save them. | String array | no | no | N/A |
| progress | It is used to determine the progress of the task. (0-100%) | Int | no | no | 0 to 100 |
| status | Determines the status of the task. The following are the statuses of the task. These are case sensitive. | String | no | no | YET\_TO\_START, IN\_PROGRESS, COMPLETED. |
| taskOwner | It has the information about the owner of the task. | JSON object | yes | no | N/A |
| owner\_id | lId of the user who is the owner of the task. We need to send this field in the task object while adding or updating the task in order to assign the owner to the task. | Long | no | no | Id of the user(Owner) |

**Get the list of pending tasks:**

**dev/api/tasks**

Method: GET

Retrieve all the pending tasks of all the users. It will return a list of task as a JSON Array.

**Using curl**

curl https:{domain}.agilecrm.com/dev/api/tasks -H "Accept:application/json" -v -u {email}:{API Key}

**Response:**

Status 200: Get the list of tasks.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Example response:**

[

{

"id": 5149503652888576,

"type": "EMAIL",

"priority\_type": "HIGH",

"due": -19800,

"created\_time": 1409649528,

"is\_complete": false,

"contacts": [

{

"id": 5704147139559424,

"type": "PERSON",

"created\_time": 1398421585,

"updated\_time": 1406619697,

...

}

],

"subject": "Contact created in agile crm",

"entity\_type": "task",

"notes": [

],

"progress": 0,

"status": "YET\_TO\_START",

"taskOwner": {

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

"is\_admin": true,

...

},

{

"id": 5176288276905984,

"type": "EMAIL",

"priority\_type": "LOW",

"due": -19800,

"created\_time": 1409649695,

"is\_complete": false,

"contacts": [

{

"id": 5704147139559424,

"type": "PERSON",

"created\_time": 1398421585,

"updated\_time": 1406619697,

...

}

],

"subject": "Contact created in agile crm",

"entity\_type": "task",

"notes": [

],

"progress": 0,

"status": "YET\_TO\_START",

"taskOwner": {

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

...

}

]

**Get all tasks:**

**dev/api/tasks/all**

Method: GET

Get all the tasks of all the users. It will get both pending and and completed tasks of all the users.

**Using curl**

curl https:{domain}.agilecrm.com/dev/api/tasks/all -H "Accept:application/json" -v -u {email}:{API Key}

**Response:**

Status 200: Get the list of tasks.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Example response:**

[

{

"id": 5149503652888576,

"type": "EMAIL",

"priority\_type": "HIGH",

"due": -19800,

"created\_time": 1409649528,

"is\_complete": false,

"contacts": [

{

"id": 5704147139559424,

"type": "PERSON",

"created\_time": 1398421585,

"updated\_time": 1406619697,

...

}

],

"subject": "Contact created in agile crm",

"entity\_type": "task",

"notes": [

],

"progress": 0,

"status": "YET\_TO\_START",

"taskOwner": {

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

"is\_admin": true,

...

},

{

"id": 5176288276905984,

"type": "EMAIL",

"priority\_type": "LOW",

"due": -19800,

"created\_time": 1409649695,

"is\_complete": true,

"contacts": [

{

"id": 5704147139559424,

"type": "PERSON",

"created\_time": 1398421585,

"updated\_time": 1406619697,

...

}

],

"subject": "Contact created in agile crm",

"entity\_type": "task",

"notes": [

],

"progress": 0,

"status": "YET\_TO\_START",

"taskOwner": {

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

...

}

]

**Get the list of pending tasks depending on pending days:**

**dev/api/tasks/pending/{num-days}**

Method: GET

Retrieve the pending tasks of all the users which are given ‘**num-days**’ days away from the due date. It will return a list of task as a JSON Array.

**Using curl**

curl https:{domain}.agilecrm.com/dev/api/tasks/pending/3 -H "Accept:application/json" -v -u {email}:{API Key}

**Response:**

Status 200: Get the list of tasks.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Example response:**

[

{

"id": 5149503652888576,

"type": "EMAIL",

"priority\_type": "HIGH",

"due": -19800,

"created\_time": 1409649528,

"is\_complete": false,

"contacts": [

{

"id": 5704147139559424,

"type": "PERSON",

"created\_time": 1398421585,

"updated\_time": 1406619697,

...

}

],

"subject": "Contact created in agile crm",

"entity\_type": "task",

"notes": [

],

"progress": 0,

"status": "YET\_TO\_START",

"taskOwner": {

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

"is\_admin": true,

...

},

{

"id": 5176288276905984,

"type": "EMAIL",

"priority\_type": "LOW",

"due": -19800,

"created\_time": 1409649695,

"is\_complete": false,

"contacts": [

{

"id": 5704147139559424,

"type": "PERSON",

"created\_time": 1398421585,

"updated\_time": 1406619697,

...

}

],

"subject": "Contact created in agile crm",

"entity\_type": "task",

"notes": [

],

"progress": 0,

"status": "YET\_TO\_START",

"taskOwner": {

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

...

}

]

**Get the list of tasks of current user:**

**dev/api/tasks/pending/{num-days}**

Method: GET

Retrieve all tasks of the current user. It will return a list of task as a JSON Array.

**Using curl**

curl https:{domain}.agilecrm.com/dev/api/tasks/my/tasks -H "Accept:application/json" -v -u {email}:{API Key}

**Response:**

Status 200: Get the list of tasks.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Example response:**

[

{

"id": 5149503652888576,

"type": "EMAIL",

"priority\_type": "HIGH",

"due": -19800,

"created\_time": 1409649528,

"is\_complete": false,

"contacts": [

{

"id": 5704147139559424,

"type": "PERSON",

"created\_time": 1398421585,

"updated\_time": 1406619697,

...

}

],

"subject": "Contact created in agile crm",

"entity\_type": "task",

"notes": [

],

"progress": 0,

"status": "YET\_TO\_START",

"taskOwner": {

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

"is\_admin": true,

...

},

{

"id": 5176288276905984,

"type": "EMAIL",

"priority\_type": "LOW",

"due": -19800,

"created\_time": 1409649695,

"is\_complete": false,

"contacts": [

{

"id": 5704147139559424,

"type": "PERSON",

"created\_time": 1398421585,

"updated\_time": 1406619697,

...

}

],

"subject": "Contact created in agile crm",

"entity\_type": "task",

"notes": [

],

"progress": 0,

"status": "YET\_TO\_START",

"taskOwner": {

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

...

}

]

**Get the list of tasks based on given filters:**

**dev/api/tasks/fordue**

Method: GET

Retrive the list of tasks based on the given filters. The filters available or ‘type’, ‘owner’, ’pending’, ‘start\_time’, ‘end\_time’ and ‘page\_size’. These should be sent as a query parameters in the URL. Filters start\_time and end\_time are the epoch time and these will filter on due date of the task.

**Using curl**

curl https:{domain}.agilecrm.com/dev/api/tasks/based?pending=true&type=EMAIL -H "Accept:application/json" -v -u {email}:{API Key}

**Response:**

Status 200: Get the list of tasks.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Example response:**

[

{

"id": 5149503652888576,

"type": "EMAIL",

"priority\_type": "HIGH",

"due": -19800,

"created\_time": 1409649528,

"is\_complete": false,

"contacts": [

{

"id": 5704147139559424,

"type": "PERSON",

"created\_time": 1398421585,

"updated\_time": 1406619697,

...

}

],

"subject": "Contact created in agile crm",

"entity\_type": "task",

"notes": [

],

"progress": 0,

"status": "YET\_TO\_START",

"taskOwner": {

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

"is\_admin": true,

...

},

{

"id": 5176288276905984,

"type": "EMAIL",

"priority\_type": "LOW",

"due": -19800,

"created\_time": 1409649695,

"is\_complete": false,

"contacts": [

{

"id": 5704147139559424,

"type": "PERSON",

"created\_time": 1398421585,

"updated\_time": 1406619697,

...

}

],

"subject": "Contact created in agile crm",

"entity\_type": "task",

"notes": [

],

"progress": 0,

"status": "YET\_TO\_START",

"taskOwner": {

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

...

}

]

**Get the task based on ID:**

**dev/api/tasks/{id}**

Method: GET

Get the task with the give ID.

**Using curl**

curl https:{domain}.agilecrm.com/dev/api/tasks/5149503652888576 -H "Accept:application/json" -v -u {email}:{API Key}

**Response:**

Status 200: Get the list of tasks.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Example response:**

{

"id": 5149503652888576,

"type": "EMAIL",

"priority\_type": "HIGH",

"due": -19800,

"created\_time": 1409649528,

"is\_complete": false,

"contacts": [

{

"id": 5704147139559424,

"type": "PERSON",

"created\_time": 1398421585,

"updated\_time": 1406619697,

...

}

],

"subject": "Contact created in agile crm",

"entity\_type": "task",

"notes": [

],

"progress": 0,

"status": "YET\_TO\_START",

"taskOwner": {

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

"is\_admin": true,

...

}

**Create a task:**

**dev/api/tasks**

Method: POST

Creates a new task.

**Using curl**

curl https://{domain}.agilecrm.com/dev/api/tasks -H "Content-Type: application/json" -d ‘{"subject" : "test", ”contacts” : [“5704147139559424”] , ”owner\_id” : “5345980119515136”, \”type\” : \“EMAIL\”, \”priority\_type\” : \“HIGH\”, \”due\” : 11545245654}’ 000-v -u test@example.com:4uet78u6atfn38m9dounnq9g4u -X POST

**Response:**

Status 200: Get the list of tasks.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Example response:**

{

"id": 5149503652888576,

"type": "EMAIL",

"priority\_type": "HIGH",

"due": 11545245654,

"created\_time": 1409649528,

"is\_complete": false,

"contacts": [

{

"id": 5704147139559424,

"type": "PERSON",

"created\_time": 1398421585,

"updated\_time": 1406619697,

...

}

],

"subject": "test",

"entity\_type": "task",

"notes": [

],

"progress": 0,

"status": "YET\_TO\_START",

"taskOwner": {

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

"is\_admin": true,

...

}

**Update a task:**

**dev/api/tasks**

Method: PUT

Updates a task. We need to specify the ID of the task.

**Using curl**

curl https://{domain}.agilecrm.com/dev/api/tasks -H "Content-Type: application/json" -d ‘{"id" : 5149503652888576", "subject" : "test", ”contacts” : [“5704147139559424”] , ”owner\_id” : “5345980119515136”, \”type\” : \“EMAIL\”, \”priority\_type\” : \“HIGH\”, \”due\” : 11545245654}’ 000-v -u test@example.com:4uet78u6atfn38m9dounnq9g4u -X PUT

**Response:**

Status 200: Get the list of tasks.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Example response:**

{

"id": 5149503652888576,

"type": "EMAIL",

"priority\_type": "HIGH",

"due": 11545245654,

"created\_time": 1409649528,

"is\_complete": false,

"contacts": [

{

"id": 5704147139559424,

"type": "PERSON",

"created\_time": 1398421585,

"updated\_time": 1406619697,

...

}

],

"subject": "test",

"entity\_type": "task",

"notes": [

],

"progress": 0,

"status": "YET\_TO\_START",

"taskOwner": {

"id": 5345980119515136,

"domain": "prabathk",

"email": "prabath.kolipaka@gmail.com",

"is\_admin": true,

...

}

**Delete the task based on ID:**

**dev/api/tasks/{id}**

Method: DELETE

Delete the task having the given ID.

**Using curl**

curl https:{domain}.agilecrm.com/dev/api/tasks/5149503652888576 -H "Accept:application/json" -v -u {email}:{API Key} -X DELETE

**Response:**

Status 204: Task deleted successfully.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Events API**

**Fields :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Description | Value Type | Read Only | Mandatory | Accepted Values |
| id | Unique id generated when event is created | Long | no | no | N/A |
| created\_time | Creation time of Event | Long | yes | no | Epoch time |
| allDay | Determines whether the event lasts for the whole day or not. | Boolean | no | no | true / false |
| contacts | List of contacts related to the event. While saving we need to give the contact ids as a string array to relate them to the event. | String array | no | no | ID’s of the contacts |
| title | Title of the event | String | no | yes | N/A |
| color | Color states the priority of the event.  The colors are case sensitive. Should be specified in the same case. | String | no | no | High = red, Normal = #36C, Low = green. |
| start | Start time of Event in epoch time format. | Long | no | yes | epoch time |
| end | End time of Event in epoch time format. | Long | no | yes | epoch time |
| is\_event\_starred | It determines whether the particular event is starred or not. | Boolean | no | no | true / false |

**Get List of Events:**

**dev/api/events**

Method: GET

Fetches the list of events between particular time. We need send the start and end time in epoch format as query parameters.

**Using curl**

curl https:{domain}.agilecrm.com/dev/api/events?start=1409423400&end=1413052200 -H "Accept:application/json" -v -u {email}:{API Key}

**Response:**

Status 200: Get the list of events.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Example Response:**

[

{

"id": 5758048710688768,

"start": 1409682600,

"end": 1409768100,

"is\_event\_starred": false,

"allDay": true,

"title": "Today Test event",

"color": "red",

"created\_time": 1409569472,

"contacts": [

{

"id": 5716606839685120,

"type": "PERSON",

"star\_value": 0,

"lead\_score": 0,

"tags": [],

"properties": [

{

"type": "SYSTEM",

"name": "first\_name",

"value": "Basecamp (2Desk)"

},

...

],

...

}

]

},

{

"id": 5740240702537728,

"start": 1409769000,

"end": 1409855340,

"is\_event\_starred": false,

"allDay": false,

"title": "tlest",

"color": "green",

"created\_time": 1410421808,

"contacts": [

{

"id": 5727217287954432,

"type": "PERSON",

"star\_value": 0,

"lead\_score": 1,

"tags": [

" mobile",

"test"

],

"properties": [

{

"type": "SYSTEM",

"name": "first\_name",

"value": "test first"

}

...

],

...

}

]

}

]

**Get Events related to Contact:**

**dev/api/contacts/{contact\_id}/events/sort**

Method: GET

Retrieve the events related to contact sorted on the date.

Using curl

curl https://{domain}.agilecrm.com/dev/api/contacts/{contact\_id}/events/sort -H "Accept: application/xml" -v -u {email}:{apikey}

**Response:**

Status 200: Returns the events list related to the contact.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Example Response:**

[

{

"id": 5758048710688768,

"start": 1409682600,

"end": 1409768100,

"is\_event\_starred": false,

"allDay": true,

"title": "Today Test event",

"color": "red",

"created\_time": 1409569472,

"contacts": [

{

"id": 5716606839685120,

"type": "PERSON",

…...

}

]

}

]

**Create Event:**

**dev/api/events**

Method: POST

Create an event.

**Using curl :**

curl https://{domain}.agilecrm.com/dev/api/events -H “Content-Type : application/json” -H “Accept : application/json” -d ‘{”title” : ”Today Test event” , ”allDay” : true, ”color” : “red” , “start”:1409682600,”end”:1409768100, ”contact\_ids” : [”721001”, ”722001”] }’ -v -u {email} : {APi Key} -X POST

**Example response :**

{

"id": 5758048710688768,

"start": 1409682600,

"end": 1409768100,

"is\_event\_starred": false,

"allDay": true,

"title": "Today Test event",

"color": "red",

"created\_time": 1409569472,

"contacts": [

{

"id": 721001,

"type": "PERSON",

…...

}

]

}

**Response Statuses:**

Status 200: Event added successfully and return the newly created event as JSON in response.

Status 401: Unauthorised. When the user name and password fields are wrong.

Status 400: If the input is in wrong format

**Update Event:**

**dev/api/events**

Method: PUT

Update an event. To update the event, we should provide the event id in the request object, otherwise it will be considered as a new event.

**Using curl :**

curl https://{domain}.agilecrm.com/dev/api/events -H “Content-Type : application/json” -H “Accept : application/json” -d ‘{“id”: 5758048710688768, ”title” : ”Today Test event” , ”allDay” : true, ”color” : “red” , “start”:1409682600,”end”:1409768100, ”contact\_ids” : [”721001”, ”722001”] }’ -v -u {email} : {APi Key} -X PUT

**Example response :**

{

"id": 5758048710688768,

"start": 1409682600,

"end": 1409768100,

"is\_event\_starred": false,

"allDay": true,

"title": "Today Test event",

"color": "red",

"created\_time": 1409569472,

"contacts": [

{

"id": 721001,

"type": "PERSON",

…...

}

]

}

**Response Statuses:**

Status 200: Event updated successfully and return the updated event as JSON in response.

Status 401: Unauthorised. When the user name and password fields are wrong.

Status 400: If the input is in wrong format

**Delete an Event:**

**dev/api/events/{id}**

Method: DELETE

Delete the event with the particular id. The id passed in the url will be used to identify the event.

**Using curl**

curl https:{domain}.agilecrm.com/dev/api/events/5149503652888576 -H "Accept:application/json" -v -u {email}:{API Key} -X DELETE

**Response:**

Status 204: Event deleted successfully.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Track / Milestones API**

**Fields :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name | Description | Value Type | Read Only | Mandatory | Accepted values |
| id | Unique id generated when Track is created | Long | no | no | N/A |
| name | Name of the track | String | no | yes | N/A |
| milestones | Comma separated strings. Each string is a milestone and these are case sensitive. Should be specified in the same case. Need to specify the first letter of the track in upper case. | String | no | yes | N/A |

**Get all the Tracks:**

**dev/api/milestone/pipelines**

Method: GET

Get all the tracks. Each track will be having set of milestones.

Using curl

curl https://{domain}.agilecrm.com/dev/api/milestone/pipelines -H "Accept: application/xml" -v -u {email}:{apikey}

**Response:**

Status 200: Returns the events list related to the contact.

Status 401: Unauthorised. When the user name and password fields are wrong.

**Example Response:**

[

{

"id": 5146448186310656,

"milestones": "Open,New,Prospect,Proposal,Won,Lost",

"name": "Default"

}

]

**Create a Track:**

**dev/api/milestone/pipelines**

Method: POST

Create a track. Name should not be “Default”. Milestone fields should be a string with group of strings separated by comma. All the milestones first letter should be in uppercase.

**Using curl :**

curl https://{domain}.agilecrm.com/dev/api/milestone/pipelines -H “Content-Type : application/json” -H “Accept : application/json” -d ‘{”name” : ”Test” , ”milestone” : “Open,New,Prospect,Proposal,Won,Lost” }’ -v -u {email} : {APi Key} -X POST

**Example Response:**

{

"id": 5146448186310656,

"milestones": "Open,New,Prospect,Proposal,Won,Lost",

"name": "Default"

}

**Response Statuses:**

Status 200: Track created successfully and return the newly created track as JSON in response.

Status 401: Unauthorised. When the user name and password fields are wrong.

Status 400: If the input is in wrong format

**Update a Track:**

**dev/api/milestone/pipelines**

Method: PUT

Update a track. Name should not be “Default”. Milestone fields should be a string with group of strings separated by comma. All the milestones first letter should be in uppercase. Need to specify the Id of the track in the request json to update the track. Otherwise, it will be considered as a new track.

**Using curl :**

curl https://{domain}.agilecrm.com/dev/api/milestone/pipelines -H “Content-Type : application/json” -H “Accept : application/json” -d ‘{“id”: 5146448186310656,

”name” : ”Test” , ”milestone” : “Open,New,Prospect,Proposal,Won,Lost” }’ -v -u {email} : {APi Key} -X PUT

**Example Response:**

{

"id": 5146448186310656,

"milestones": "Open,New,Prospect,Proposal,Won,Lost",

"name": "Default"

}

**Response Statuses:**

Status 200: Track updated successfully and return the updated track as JSON in response.

Status 401: Unauthorised. When the user name and password fields are wrong.

Status 400: If the input is in wrong format

**Delete an Track:**

**dev/api/milestone/pipelines/{id}**

Method: DELETE

Delete the track with the particular id. The id passed in the url will be used to identify the track.

**Using curl**

curl https:{domain}.agilecrm.com/dev/api/milestone/pipelines/5149503652888576 -H "Accept:application/json" -v -u {email}:{API Key} -X DELETE

**Response:**

Status 204: Track deleted successfully.

Status 401: Unauthorised. When the user name and password fields are wrong.