



FIVE CRM

API Version 2

Copyright 2018
Quality System Solutions Ltd
Lansdowne Court
Bumpers Way
Chippenham
Wiltshire
SN14 6RZ
United Kingdom

E&OE. All rights reserved. Quality System Solutions Ltd reserves the right to change specifications without prior notice. Questions should be directed to the support desk: support@fivecrm.com

This api functionality is covered by the same Terms and Conditions as already agreed for the FIVE CRM customer instance.

Contents

2. Introduction	2
2.1 Global Parameters.....	2
3. Modes	3
3.1 System.....	3
3.2 Privacy Details.....	3
3.3 Master Details.....	4
3.4 Account information.....	4
4. Global Error Messages	5
5. System Modes	6
5.1 Mode: getDataBaseArea.....	6
5.2 Mode - getDatabaseAreas.....	11
5.3 Mode - validateHash.....	15
6. Privacy Details	19
6.1 Mode: getPrivacyDetails.....	19
Error Messages.....	23
6.2 Mode - getProcessingChannels.....	26
6.3 Mode - getProcessingCategories.....	30
7. Master Details	35
7.1 Mode – getAccountAdditionalFields.....	35
7.2 Mode - getContactAdditionalFields.....	41
7.3 Mode - getCustomerStatuses.....	47
7.4 Mode - getOwners.....	53
7.5 Mode - getProjects.....	58
7.6 Mode - getSalesPeople.....	64
8 Account information	70
8.1 Mode: importCustomerProducts.....	70
8.2 Mode - updateCustomerProducts.....	81

2. Introduction

This API2 is not a replacement for the original API, rather, it provides additional functionality which is not available in the original API.

This API endpoint is available to all FIVE CRM system types.

The base API endpoint url is as follows: <path to FIVE CRM installation>/api2.php

e.g. <https://europeX.fivecrm.com/12345/api2.php>

All data returned from this API endpoint will be in a JSON format.

2.1 Global Parameters

All methods on this API endpoint require the following parameters.

Parameter	Description	Data Type	Parameter Type (GET/POST)	Required
mode	The API mode to use. This must be one of the available API modes detailed in the 'API Modes' section.	string	GET or POST	Yes
hash	A valid API authentication hash (used to authenticate the request). Each user profile has a hash generated that can be retrieved when editing a user profile.	string	GET or POST	Yes

3. Modes

3.1 System

API Mode Name	Description
getDatabaseArea	Used to retrieve information about a specific database area.
getDatabaseAreas	Used to retrieve information about all allowable database areas.
validateHash	Used to confirm whether or not the provided API hash is authorized or not.

3.2 Privacy Details

API Mode Name	Description
getPrivacyDetails	Used to retrieve privacy details from the system based on search criteria provided. E.g. get all privacy details where the account name is equal to 'FIVE CRM'.
getProcessingCategories	Used to retrieve all of the processing categories for a database area.
getProcessingChannels	Used to retrieve all of the processing channels for a database area.

3.3 Master Details

API Mode Name	Description
getAccountAdditionalFields	Used to retrieve all of the account additional field configurations for a database area.
getContactAdditionalFields	Used to retrieve all of the contact additional field configurations for a database area.
getCustomerStatuses	Used to retrieve all of the configured allowable customer statuses for a database area.
getOwners	Used to retrieve all of the configured, allowable owners for a database area.
getProjects	Used to retrieve all of the configured, allowable projects for a database area.
getSalesPeople	Used to retrieve all of the configured sales people for a specific database area.

3.4 Account information

API Mode Name	Description
importCustomerProducts	Used to import new customer products into the system.
updateCustomerProducts	Used to update pre-existing customer products in the system.

4. Global Error Messages

The following error messages can be returned by any of the API modes.

Message	Description
System has expired	The system has expired, i.e. is no longer being paid for.
No API mode specified	The mode parameter was not provided, or was empty.
Unknown API mode specified	An unknown API mode was specified. Valid API modes are listed in the API modes section.

Example Return Data

```
{
  "STATUS": "error",
  "ERROR": [
    "No API mode specified"
  ]
}
```

5. System Modes

5.1 Mode: getDataBaseArea

This API mode endpoint can be used to retrieve information about a specific database area.

Request Data

Below are the available parameters which can be provided. These are in addition to the global parameters.

Parameter	Description	Data Type (length)	Parameter Type (GET/POST)	Required
area	The database area	string (3)	GET or POST	Yes

Example Request Data

```
{
  "area": "001",
  "hash": "89a0aa9a95d2b5bc48eeab2287ad65g7",
  "mode": "getDataBaseArea"
}
```

Request Return Data

The request can return one of two statuses. Each of these statuses depict the result of the request. The two statuses are as follows: 'success' and 'error'. Each status and returned data structure is outlined below.

Success Request Return Data

If the request is successful then the returned data will follow the structure below. The fields will be returned in the same order as configured in the system.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'success' when the request was successful.	string
AREA	N/A	An object containing information about the requested database area.	object
number	AREA	The database area number (including leading zeros).	string
name	AREA	The configured database area name.	string
description	AREA	The configured database area description.	string
active	AREA	Whether or not the database area is currently active.	boolean
isMultipleProjects	AREA	Whether or not the database area has been configured to use multiple projects per account.	boolean

Example Return Data

```
{
  "STATUS": "success",
  "AREA": {
    "number": "001",
    "name": "Demo sales Area",
    "description": "my area description",
    "active": true,
    "isMultipleProjects": false
  }
}
```

Failed Request Return Data

If the request is unsuccessful then the returned data will follow the structure below.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'error' when the request was unsuccessful.	string
ERROR	N/A	An array of errors for the request.	array

Example Return Data

```
{
  "STATUS": "error",
  "ERROR": [
    "The specified database area does not exist"
  ]
}
```

Error Messages

Below are the specific error messages which can be returned by this API mode endpoint. Please note that any global error messages can also be returned.

Message	Description
Unauthorised	The provided API authentication hash did not match any known hash in the system.
No database area has been specified	No 'area' parameter was specified in the request.
The specified database area does not exist	The 'area' parameter specified was not a valid database area.
You do not have the necessary user permissions to access this database area.	The user profile associated with the provided API hash does not have access to the database area requested.
An internal error occurred while trying to get the Database Area.	An internal error occurred.

Code Examples

Javascript (with jQuery)

```
$(function() {
  $.ajax(
    {
      url: 'yourSystemUrl/api2.php?' +
        'mode=' + encodeURIComponent('getDatabaseArea') +
        '&area=' + encodeURIComponent('001'),
      data: {
        hash: '86ab15zaf95c3431ac68e115f515g1a0'
      },
      type: 'POST',
      dataType: 'json',
      success: function(response) {
        //Manage the returned data.
      }
    }
  );
});
```

```
<?php

$baseUrl = 'yourSystemUrl/api2.php';

$getRequestData = array(
    'mode'     => 'getDatabaseArea',
    'area'     => '001'
);

$postRequestData = array(
    'hash' => '86ab15zaf95c3431ac68e115f515g1a0',
);

$url = $baseUrl . '?' . http_build_query($getRequestData);
$curl = curl_init($url);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt(
    $curl,
    CURLOPT_POSTFIELDS,
    http_build_query($postRequestData)
);
$response = curl_exec($curl);
curl_close($curl);

$decodedResponse = json_decode($response, true);
//Manage the response using $decodedResponse
```

5.2 Mode - getDatabaseAreas

This API mode endpoint can be used to retrieve information about configured database areas.

Request Data

No additional request parameters are required.

Example Request Data

```
{
  "hash": "89a0aa9a95d2b5bc48eeab2287ad65g7",
  "mode": "getDatabaseAreas"
}
```

Request Return Data

The request can return one of two statuses. Each of these statuses depict the result of the request. The two statuses are as follows: 'success' and 'error'. Each status and returned data structure is outlined below.

Success Request Return Data

If the request is successful then the returned data will follow the structure below. The fields will be returned in the same order as configured in the system.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'success' when the request was successful.	string
AREA	N/A	An object containing information about the requested database area.	object
number	AREA	The database area number (including leading zeros).	string
name	AREA	The configured database area name.	string
description	AREA	The configured database area description.	string
active	AREA	Whether or not the database area is currently active.	boolean
isMultipleProjects	AREA	Whether or not the database area has been configured to use multiple projects per account.	boolean

Example Return Data

```
{
  "STATUS": "success",
  "AREAS": [
    {
      "number": "001",
      "name": "Sales Area",
      "description": "For sales to manage accounts",
      "active": true,
      "isMultipleProjects": false
    },
    {
      "number": "002",
      "name": "Marketing Area",
      "description": "",
      "active": true,
      "isMultipleProjects": true
    }
  ]
}
```

Failed Request Return Data

If the request is unsuccessful then the returned data will follow the structure below.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'error' when the request was unsuccessful.	string
ERROR	N/A	An array of errors for the request.	array

Example Return Data

```
{
  "STATUS": "error",
  "ERROR": [
    "Unauthorised"
  ]
}
```

Error Messages

Below are the specific error messages which can be returned by this API mode endpoint. Please note that any global error messages can also be returned.

Message	Description
Unauthorised	The provided API authentication hash did not match any known hash in the system.
An internal error occurred while trying to get the Database Areas.	An internal error occurred while trying to get the Database Areas.

Code Examples

Javascript (with jQuery)

```
$(function() {
    $.ajax(
        {
            url: 'yourSystemUrl/api2.php?' +
                'mode=' + encodeURIComponent('getDatabaseAreas'),
            data: {
                hash: '86ab15zaf95c3431ac68e115f515g1a0'
            },
            type: 'POST',
            dataType: 'json',
            success: function(response) {
                //Manage the returned data.
            }
        }
    );
});
```

```
<?php

$baseUrl = 'yourSystemUrl/api2.php';

$getRequestData = array(
    'mode' => 'getDatabaseAreas'
);

$postRequestData = array(
    'hash' => '86ab15zaf95c3431ac68e115f515g1a0',
);

$url = $baseUrl . '?' . http_build_query($getRequestData);
$curl = curl_init($url);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt(
    $curl,
    CURLOPT_POSTFIELDS,
    http_build_query($postRequestData)
);
$response = curl_exec($curl);
curl_close($curl);

$decodedResponse = json_decode($response, true);
//Manage the response using $decodedResponse
```


5.3 Mode - validateHash

This API mode endpoint can be used to retrieve the configured projects for a database area.

Request Data

No additional request parameters are required.

Example Request Data

```
{
  "hash": "89a0aa9a95d2b5bc48eeab2287ad65g7",
  "mode": "getSalesPeople"
}
```

Request Return Data

The request can return one of two statuses. Each of these statuses depict the result of the request. The two statuses are as follows: 'success' and 'error'. Each status and returned data structure is outlined below.

Success Request Return Data

If the request is successful then the returned data will follow the structure below. The fields will be returned in the same order as configured in the system.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'success' when the request was successful.	string
MESSAGE	N/A	An array of messages.	array

Example Return Data

```
{
  "STATUS": "success",
  "MESSAGE": [
    "Hash is valid"
  ]
}
```

Failed Request Return Data

If the request is unsuccessful then the returned data will follow the structure below.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'error' when the request was unsuccessful.	string
ERROR	N/A	An array of errors for the request.	array

Example Return Data

```
{
  "STATUS": "error",
  "ERROR": [
    "Unauthorised"
  ]
}
```

Error Messages

Below are the specific error messages which can be returned by this API mode endpoint. Please note that any global error messages can also be returned.

Message	Description
Unauthorised	The provided API authentication hash did not match any known hash in the system.

Code Examples

Javascript (with jQuery)

```
$(function() {  
  $.ajax(  
    {  
      url: 'yourSystemUrl/api2.php?' +  
        'mode=' + encodeURIComponent('validateHash'),  
      data: {  
        hash: '86ab15zaf95c3431ac68e115f515g1a0'  
      },  
      type: 'POST',  
      dataType: 'json',  
      success: function(response) {  
        //Manage the returned data.  
      }  
    }  
  );  
});
```

```
<?php

$baseUrl = 'yourSystemUrl/api2.php';

$getRequestData = array(
    'Mode' => 'validateHash'
);

$postRequestData = array(
    'hash' => '86ab15zaf95c3431ac68e115f515g1a0',
);

$url = $baseUrl . '?' . http_build_query($getRequestData);
$curl = curl_init($url);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt(
    $curl,
    CURLOPT_POSTFIELDS,
    http_build_query($postRequestData)
);
$response = curl_exec($curl);
curl_close($curl);

$decodedResponse = json_decode($response, true);
//Manage the response using $decodedResponse
```

6. Privacy Details

6.1 Mode: getPrivacyDetails

This API mode endpoint can be used to retrieve privacy details for particular contacts based on the search criteria specified.

Request Data

Below are the available parameters which can be provided. These are in addition to the global parameters. 'area' is the only required field which must be provided as part of the request. However, at least one of the optional parameters must be provided. Multiple optional parameters can be provided on the same request. When providing multiple optional parameters, the search will be performed based on a contact who matches all of the criteria.

Parameter	Description	Data Type	Parameter Type (GET/POST)	Required
area	The database area	string	GET	Yes
contName	A contact's name	string	GET	No, however at least one of the optional parameters must be provided.
contEmail	A contact's email address	string	GET	No, however at least one of the optional parameters must be provided.
contExternalContactId	A contact's external id	string	GET	No, however at least one of the optional parameters must be provided.
contTelephone	A contact's telephone OR mobile number	string	GET	No, however at least one of the optional parameters must be provided.
accCompany	A company/account name	string	GET	No, however at least one of the optional parameters must be provided.

Parameter	Description	Data Type	Parameter Type (GET/POST)	Required
accExtLeadId	An account's external lead id	string	GET	No, however at least one of the optional parameters must be provided.
accExtAccountId	An account's external account id	string	GET	No, however at least one of the optional parameters must be provided.
accPostcode	An account's postcode	string	GET	No, however at least one of the optional parameters must be provided.

Example Request Data

```
{  
  "area": "001",  
  "hash": "89a0aa9a95d2b5bc48eeab2287ad65g7",  
  "mode": "getPrivacyDetails",  
  "contName": "Joe Bloggs",  
  "contEmail": "joe.bloggs@example.com"  
}
```

Successful Request Return Data

If the request is successful then the returned data will follow the structure below.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'success' when the request was successful.	string
CONTACTS	N/A	An array of arrays of those contact who match the search criteria.	array
ID	CONTACTS	The contact's system id. This is a unique identifier used to be able to differentiate between contacts.	string
ACCOUNT	CONTACTS	The account number the contact belongs to.	string
PRIVACY_DETAILS	CONTACTS	An array of arrays of the privacy details for the contact.	array
LAWFUL_TYPE	PRIVACY_DETAILS	The lawful type for the privacy detail.	string
START_DATE	PRIVACY_DETAILS	The date when the privacy detail starts. All dates and times are in UTC.	string
END_DATE	PRIVACY_DETAILS	The date when the privacy detail ends. All dates and times are in UTC.	string
ADDITIONAL_INFO	PRIVACY_DETAILS	Any additional information provided for the privacy detail.	string
WITHDRAWN_DATE	PRIVACY_DETAILS	The date when the privacy detail was withdrawn. In scenarios where the privacy detail is not withdrawn, this will be empty. All dates and times are in UTC.	string
PROCESSING_DETAILS	PRIVACY_DETAILS	An array of processing details for the privacy details. These are the channels and categories which have been selected in the privacy detail. The	array

		array indices will be the categories. Within each category will be the channels selected for that particular category.	
--	--	--	--

Example Return Data

```
{
  "STATUS": "success",
  "CONTACTS": [
    {
      "ID": "1",
      "ACCOUNT": "1",
      "PRIVACY_DETAILS": [
        {
          "LAWFUL_TYPE": "Consent",
          "START_DATE": "2017-09-06 21:00:00",
          "END_DATE": "2017-09-15 20:59:59",
          "ADDITIONAL_INFO": "",
          "WITHDRAWN_DATE": "2017-09-15 20:59:59",
          "PROCESSING_DETAILS": {
            "Car Insurance": [
              "Email Tracking",
              "Emails",
              "Phone"
            ],
            "Home Insurance": [
              "Phone"
            ]
          }
        }
      ]
    }
  ]
}
```


Failed Request Return Data

If the request is unsuccessful then the returned data will follow the structure below.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'error' when the request was unsuccessful.	string
ERROR	N/A	An error message explaining why the request was unsuccessful. This can be one of the error messages listed below.	string

Example Return Data

```
{
  "STATUS": "error",
  "ERROR": [
    "The specified database area does not exist"
  ]
}
```

Error Messages

Below are the specific error messages which can be returned by this API mode endpoint. Please note that any global error messages can also be returned.

Message	Description
Unauthorised	The provided API authentication hash did not match any known hash in the system.
No database area has been specified	No area parameter was specified in the request.
The specified database area does not exist	The area parameter specified was not a valid database area.
At least one search criteria must be specified	No optional parameters were provided with the request.
Unable to locate any matching contacts	No contacts could be found matching the specified search criteria.
An unexpected error occurred during contact acquisition	There was an internal API error when trying to retrieve the Privacy Details.

Code Examples

Javascript (with jQuery)

```
$(function() {  
    $.ajax(  
        {  
            url: 'yourSystemUrl/api2.php?' +  
                'mode=' + encodeURIComponent('getPrivacyDetails') +  
                '&area=' + encodeURIComponent('001') +  
                '&contName=' + encodeURIComponent('Test Contact Name') +  
                '&companyName=' + encodeURIComponent('Test Company Name'),  
            data: {  
                hash: '86ab15zaf95c3431ac68e115f515g1a0'  
            },  
            type: 'POST',  
            dataType: 'json',  
            success: function(response) {  
                //Manage the returned data.  
            }  
        }  
    );  
});
```

```
<?php

$baseUrl = 'yourSystemUrl/api2.php';

$getRequestData = array(
    'mode'      => 'getPrivacyDetails',
    'area'      => '001',
    'contEmail' => 'test@example.com'
);

$postRequestData = array(
    'hash' => '86ab15zaf95c3431ac68e115f515g1a0',
);

$url = $baseUrl . '?' . http_build_query($getRequestData);
$curl = curl_init($url);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt(
    $curl,
    CURLOPT_POSTFIELDS,
    http_build_query($postRequestData)
);
$response = curl_exec($curl);
curl_close($curl);

$decodedResponse = json_decode($response, true);
//Manage the response using $decodedResponse
```

6.2 Mode - getProcessingChannels

This API mode endpoint can be used to retrieve the Processing Channels for a database area.

Request Data

Below are the available parameters which can be provided. These are in addition to the global parameters.

Parameter	Description	Data Type (length)	Parameter Type (GET/POST)	Required
area	The database area	string (3)	POST	Yes

Example Request Data

```
{
  "area": "001",
  "hash": "89a0aa9a95d2b5bc48eeab2287ad65g7",
  "mode": "getProcessingChannels"
}
```

Request Return Data

A getProcessingChannels request can return one of two statuses. Each of these statuses depict the result of the request. The two statuses are as follows: 'success' and 'error'. Each status and returned data structure is outlined below.

Success Request Return Data

If the request is successful then the returned data will follow the structure below.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'success' when the request was successful.	string
PROCESSING_CHANNELS	N/A	An array of Processing Channels for the database area.	array
id	PROCESSING_CHANNELS	A unique identifier for the Processing Channel.	integer
name	PROCESSING_CHANNELS	The name of the Processing Channel.	string

enabled	PROCESSING_CHANNELS	Whether or not the Processing Channel is enabled. 'true' if it is enabled, 'false' if it is disabled.	boolean
---------	---------------------	---	---------

Example Return Data

```
{
  "STATUS": "success",
  "PROCESSING_CHANNELS": [
    {
      "id": 1,
      "name": "Email",
      "enabled": true
    },
    {
      "id": 2,
      "name": "SMS",
      "enabled": true
    },
    {
      "id": 3,
      "name": "Phone",
      "enabled": false
    }
  ]
}
```

Failed Request Return Data

If the request is unsuccessful then the returned data will follow the structure below.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'error' when the request was unsuccessful.	string
ERROR	N/A	An array of errors for the request.	array

Example Return Data

```
{
  "STATUS": "error",
  "ERROR": [
    "The specified database area does not exist"
  ]
}
```

Error Messages

Below are the specific error messages which can be returned by this API mode endpoint. Please note that any global error messages can also be returned.

Message	Description
Unauthorised	The provided API authentication hash did not match any known hash in the system.
No database area has been specified	No 'area' parameter was specified in the request.
The specified database area does not exist	The 'area' parameter specified was not a valid database area.
An internal error occurred while trying to get the Processing Channels.	An internal error occurred.

Code Examples

Javascript (with jQuery)

```
$(function() {
    $.ajax(
        {
            url: 'yourSystemUrl/api2.php?' +
                'mode=' + encodeURIComponent('getProcessingChannels') +
                '&area=' + encodeURIComponent('001'),
            data: {
                hash: '86ab15zaf95c3431ac68e115f515g1a0'
            },
            type: 'POST',
            dataType: 'json',
            success: function(response) {
                //Manage the returned data.
            }
        }
    );
});
```

```
<?php

$baseUrl = 'yourSystemUrl/api2.php';

$getRequestData = array(
    'mode'     => 'getProcessingChannels',
    'area'     => '001'
);

$postRequestData = array(
    'hash' => '86ab15zaf95c3431ac68e115f515g1a0',
);

$url = $baseUrl . '?' . http_build_query($getRequestData);
$curl = curl_init($url);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt(
    $curl,
    CURLOPT_POSTFIELDS,
    http_build_query($postRequestData)
);
$response = curl_exec($curl);
curl_close($curl);

$decodedResponse = json_decode($response, true);
//Manage the response using $decodedResponse
```

6.3 Mode - getProcessingCategories

This API mode endpoint can be used to retrieve the Processing Categories for a database area.

Request Data

Below are the available parameters which can be provided. These are in addition to the global parameters.

Parameter	Description	Data Type (length)	Parameter Type (GET/POST)	Required
area	The database area	string (3)	POST	Yes

Example Request Data

```
{
  "area": "001",
  "hash": "89a0aa9a95d2b5bc48eeab2287ad65g7",
  "mode": "getProcessingCategories"
}
```

Request Return Data

A getProcessingCategories request can return one of two statuses. Each of these statuses depict the result of the request. The two statuses are as follows: 'success' and 'error'. Each status and returned data structure is outlined below.

Success Request Return Data

If the request is successful then the returned data will follow the structure below.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'success' when the request was successful.	string
PROCESSING_CATEGORIES	N/A	An array of Processing Categories for the database area.	array
id	PROCESSING_CATEGORIES	A unique identifier for the Processing Category.	integer
name	PROCESSING_CATEGORIES	The name of the Processing Category.	string
enabled	PROCESSING_CATEGORIES	Whether or not the Processing Category is enabled. 'true' if it is enabled, 'false' if it is disabled.	boolean

Example Return Data

```
{
  "STATUS": "success",
  "PROCESSING_CATEGORIES": [
    {
      "id": 1,
      "name": "Car Insurance",
      "enabled": true
    },
    {
      "id": 2,
      "name": "Home Insurance",
      "enabled": true
    },
    {
      "id": 3,
      "name": "Pet Insurance",
      "enabled": false
    }
  ]
}
```

Failed Request Return Data

If the request is unsuccessful then the returned data will follow the structure below.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'error' when the request was unsuccessful.	string
ERROR	N/A	An array of errors for the request.	array

Example Return Data

```
{
  "STATUS": "error",
  "ERROR": [
    "The specified database area does not exist"
  ]
}
```

Error Messages

Below are the specific error messages which can be returned by this API mode endpoint. Please note that any global error messages can also be returned.

Message	Description
Unauthorised	The provided API authentication hash did not match any known hash in the system.
No database area has been specified	No 'area' parameter was specified in the request.
The specified database area does not exist	The 'area' parameter specified was not a valid database area.
An internal error occurred while trying to get the Processing Categories..	An internal error occurred.

Code Examples

Javascript (with jQuery)

```
$(function() {
    $.ajax(
        {
            url: 'yourSystemUrl/api2.php?' +
                'mode=' + encodeURIComponent('getProcessingCategories') +
                '&area=' + encodeURIComponent('001'),
            data: {
                hash: '86ab15zaf95c3431ac68e115f515g1a0'
            },
            type: 'POST',
            dataType: 'json',
            success: function(response) {
                //Manage the returned data.
            }
        }
    );
});
```

```
<?php

$baseUrl = 'yourSystemUrl/api2.php';

$getrequestData = array(
    'mode'      => 'getProcessingCategories',
    'area'      => '001'
);

$postrequestData = array(
    'hash' => '86ab15zaf95c3431ac68e115f515g1a0',
);

$url = $baseUrl . '?' . http_build_query($getrequestData);
$curl = curl_init($url);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt(
    $curl,
    CURLOPT_POSTFIELDS,
    http_build_query($postrequestData)
);
$response = curl_exec($curl);
curl_close($curl);

$decodedResponse = json_decode($response, true);
//Manage the response using $decodedResponse
```

7.Master Details

7.1 Mode – getAccountAdditionalFields

This API mode endpoint can be used to retrieve the configuration of the account additional fields for a database area.

Request Data

Below are the available parameters which can be provided. These are in addition to the global parameters.

Parameter	Description	Data Type (length)	Parameter Type (GET/POST)	Required
area	The database area	string (3)	POST	Yes

Example Request Data

```
{
  "area": "001",
  "hash": "89a0aa9a95d2b5bc48eeab2287ad65g7",
  "mode": "getAccountAdditionalFields"
}
```

Request Return Data

A getAccountAdditionalFields request can return one of two statuses. Each of these statuses depict the result of the request. The two statuses are as follows: 'success' and 'error'. Each status and returned data structure is outlined below.

Success Request Return Data

If the request is successful then the returned data will follow the structure below. The fields will be returned in the same order as configured in the system.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'success' when the request was successful.	string
ACCOUNT_ADDITIONAL_FIELDS	N/A	An array of account additional fields for the database area.	array
name	ACCOUNT_ADDI	The system name for the additional	string

	TIONAL_FIELDS	field. E.g. ANAL02	
number	ACCOUNT_ADDITIONAL_FIELDS	The numerical value of the additional field. For example additional field 04 would have a number value of 4.	integer
heading	ACCOUNT_ADDITIONAL_FIELDS	The human-readable name for the additional field. E.g. Account Additional Field 9	string
sortId	ACCOUNT_ADDITIONAL_FIELDS	The value used to determine the sort order of the fields	integer
type	ACCOUNT_ADDITIONAL_FIELDS	Contains information about the type of field that has been configured.	object
type	Type	The type of data the field has been configured to store. E.g. varchar	string
friendlyName	Type	A more friendly way of describing the type of data the field stores. E.g. String	string
length	Type	The maximum number of characters this field has been configured to store. Note that this only applies to varchar fields.	integer
dropdownItems	ACCOUNT_ADDITIONAL_FIELDS	A list of all of the configured dropdown items for the additional field.	array
name	dropdownItems	The name of a dropdown item	string
sortId	dropdownItems	The value used to determine the sort order of the dropdown item.	integer

Example Return Data

```
{
  "STATUS": "success",
  "ACCOUNT_ADDITIONAL_FIELDS": [
    {
      "name": "ANAL02",
      "number": 2,
      "heading": "Lead Source",
      "sortId": 1150,
      "type": {
        "type": "varchar",
        "friendlyName": "String",
        "length": 100
      },
      "dropdownItems": [
        {
```

```
        "name": "Website Enquiry",
        "sortId": 0
    },
    {
        "name": "Survey Response",
        "sortId": 1
    },
    {
        "name": "Email",
        "sortId": 2
    }
]
},
{
    "name": "ANAL01",
    "number": 1,
    "heading": "Custom Appointment Date",
    "sortId": 1160,
    "type": {
        "type": "datetime",
        "friendlyName": "Date / Time",
        "length": 0
    },
    "dropdownItems": []
},
{
    "name": "ANAL50",
    "number": 50,
    "heading": "Sales Made",
    "sortId": 1640,
    "type": {
        "type": "int",
        "friendlyName": "Integer",
        "length": 0
    },
    "dropdownItems": []
},
...
//More fields would appear here.
]
}
```

Failed Request Return Data

If the request is unsuccessful then the returned data will follow the structure below.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'error' when the request was unsuccessful.	string
ERROR	N/A	An array of errors for the request.	array

Example Return Data

```
{
  "STATUS": "error",
  "ERROR": [
    "The specified database area does not exist"
  ]
}
```


Error Messages

Below are the specific error messages which can be returned by this API mode endpoint. Please note that any global error messages can also be returned.

Message	Description
Unauthorised	The provided API authentication hash did not match any known hash in the system.
No database area has been specified	No 'area' parameter was specified in the request.
The specified database area does not exist	The 'area' parameter specified was not a valid database area.
You do not have the necessary user permissions to access this database area.	The user profile associated with the provided API hash does not have access to the database area requested.
An internal error occurred while trying to get the Account Additional Fields.	An internal error occurred.
An error occurred when trying to get the dropdown items.	An internal error occurred when trying to get the dropdown items for one of the additional fields.

Code Examples

Javascript (with jQuery)

```
$(function() {
    $.ajax(
        {
            url: 'yourSystemUrl/api2.php?' +
                'mode=' + encodeURIComponent('getAccountAdditionalFields') +
                '&area=' + encodeURIComponent('001'),
            data: {
                hash: '86ab15zaf95c3431ac68e115f515g1a0'
            },
            type: 'POST',
            dataType: 'json',
            success: function(response) {
                //Manage the returned data.
            }
        }
    );
});
```

PHP

```
<?php

$baseUrl = 'yourSystemUrl/api2.php';

$getrequestData = array(
    'mode'      => 'getAccountAdditionalFields',
    'area'      => '001'
);

$postrequestData = array(
    'hash' => '86ab15zaf95c3431ac68e115f515g1a0',
);

$url = $baseUrl . '?' . http_build_query($getrequestData);
$curl = curl_init($url);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt(
    $curl,
    CURLOPT_POSTFIELDS,
    http_build_query($postrequestData)
);
$response = curl_exec($curl);
curl_close($curl);

$decodedResponse = json_decode($response, true);
//Manage the response using $decodedResponse
```

7.2 Mode - getContactAdditionalFields

This API mode endpoint can be used to retrieve the configuration of the contact additional fields for a database area.

Request Data

Below are the available parameters which can be provided. These are in addition to the global parameters.

Parameter	Description	Data Type (length)	Parameter Type (GET/POST)	Required
area	The database area	string (3)	POST	Yes

Example Request Data

```
{
  "area": "001",
  "hash": "89a0aa9a95d2b5bc48eeab2287ad65g7",
  "mode": "getContactAdditionalFields"
}
```

Request Return Data

The request can return one of two statuses. Each of these statuses depict the result of the request. The two statuses are as follows: 'success' and 'error'. Each status and returned data structure is outlined below.

Success Request Return Data

If the request is successful then the returned data will follow the structure below. The fields will be returned in the same order as configured in the system.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'success' when the request was successful.	string
CONTACT_ADDITIONAL_FIELDS	N/A	An array of contact additional fields for the database area.	array
name	CONTACT_ADDITIONAL_FIELDS	The system name for the additional field. E.g. ANAL02	string
number	CONTACT_ADDITIONAL_FIELDS	The numerical value of the additional field. For example additional field 04 would have a number value of 4.	integer

heading	CONTACT_ADDITIONAL_FIELDS	The human-readable name for the additional field. E.g. Account Additional Field 9	string
sortId	CONTACT_ADDITIONAL_FIELDS	The value used to determine the sort order of the fields	integer
type	CONTACT_ADDITIONAL_FIELDS	Contains information about the type of field that has been configured.	object
type	type	The type of data the field has been configured to store. E.g. varchar	string
friendlyName	type	A more friendly way of describing the type of data the field stores. E.g. String	string
length	type	The maximum number of characters this field has been configured to store. Note that this only applies to varchar fields.	integer
dropdownItems	CONTACT_ADDITIONAL_FIELDS	A list of all of the configured dropdown items for the additional field.	array
name	dropdownItems	The name of a dropdown item	string
sortId	dropdownItems	The value used to determine the sort order of the dropdown item.	integer

Example Return Data

```
{
  "STATUS": "success",
  "CONTACT_ADDITIONAL_FIELDS": [
    {
      "name": "ADDN02",
      "number": 2,
      "heading": "Favourite Pet",
      "sortId": 1150,
      "type": {
        "type": "varchar",
        "friendlyName": "String",
        "length": 100
      }
    }
  ]
}
```

```

    },
    "dropdownItems": [
      {
        "name": "Website Enquiry",
        "sortId": 0
      },
      {
        "name": "Survey Response",
        "sortId": 1
      },
      {
        "name": "Email",
        "sortId": 2
      }
    ]
  },
  {
    "name": "ADDN01",
    "number": 1,
    "heading": "Last Contacted",
    "sortId": 1160,
    "type": {
      "type": "datetime",
      "friendlyName": "Date / Time",
      "length": 0
    },
    "dropdownItems": []
  },
  {
    "name": "ADDN50",
    "number": 50,
    "heading": "Books Owned",
    "sortId": 1640,
    "type": {
      "type": "int",
      "friendlyName": "Integer",
      "length": 0
    },
    "dropdownItems": []
  },
  ...
  //More fields would appear here.
]
}

```

Failed Request Return Data

If the request is unsuccessful then the returned data will follow the structure below.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'error' when the request was unsuccessful.	string
ERROR	N/A	An array of errors for the request.	array

Example Return Data

```
{
  "STATUS": "error",
  "ERROR": [
    "The specified database area does not exist"
  ]
}
```

Error Messages

Below are the specific error messages which can be returned by this API mode endpoint. Please note that any global error messages can also be returned.

Message	Description
Unauthorised	The provided API authentication hash did not match any known hash in the system.
No database area has been specified	No 'area' parameter was specified in the request.
The specified database area does not exist	The 'area' parameter specified was not a valid database area.
You do not have the necessary user permissions to access this database area.	The user profile associated with the provided API hash does not have access to the database area requested.
An error occurred when trying to get the contact additional fields.	An internal error occurred.
An error occurred when trying to get the dropdown items.	An internal error occurred when trying to get the dropdown items for one of the additional fields.

Code Examples

Javascript (with jQuery)

```
$(function() {  
    $.ajax(  
        {  
            url: 'yourSystemUrl/api2.php?' +  
                'mode=' + encodeURIComponent('getContactAdditionalFields') +  
                '&area=' + encodeURIComponent('001'),  
            data: {  
                hash: '86ab15zaf95c3431ac68e115f515g1a0'  
            },  
            type: 'POST',  
            dataType: 'json',  
            success: function(response) {  
                //Manage the returned data.  
            }  
        }  
    );  
});
```

```
<?php

$baseUrl = 'yourSystemUrl/api2.php';

$getrequestData = array(
    'mode'      => 'getContactAdditionalFields',
    'area'      => '001'
);

$postrequestData = array(
    'hash' => '86ab15zaf95c3431ac68e115f515g1a0',
);

$url = $baseUrl . '?' . http_build_query($getrequestData);
$curl = curl_init($url);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt(
    $curl,
    CURLOPT_POSTFIELDS,
    http_build_query($postrequestData)
);
$response = curl_exec($curl);
curl_close($curl);

$decodedResponse = json_decode($response, true);
//Manage the response using $decodedResponse
```


7.3 Mode - getCustomerStatuses

This API mode endpoint can be used to retrieve the configured customer statuses for a database area.

Request Data

Below are the available parameters which can be provided. These are in addition to the global parameters.

Parameter	Description	Data Type (length)	Parameter Type (GET/POST)	Required
area	The database area	string (3)	POST	Yes

Example Request Data

```
{
  "area": "001",
  "hash": "89a0aa9a95d2b5bc48eeab2287ad65g7",
  "mode": "getCustomerStatuses"
}
```

Request Return Data

The request can return one of two statuses. Each of these statuses depict the result of the request. The two statuses are as follows: 'success' and 'error'. Each status and returned data structure is outlined below.

Success Request Return Data

If the request is successful then the returned data will follow the structure below. The fields will be returned in the same order as configured in the system.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'success' when the request was successful.	string
CUSTOMER_STATUSES	N/A	An array of customer statuses for the database area.	array
status	CUSTOMER_STATUSES	The name for the customer status.	string
subStatuses	CUSTOMER_STATUSES	An array containing the names of any sub-statuses configured.	array
analysisGroup	CUSTOMER_STATUSES	The analysis group configured on the status (primarily used for reporting purposes).	string
interval	CUSTOMER_STATUSES	The interval value configured on the status	integer

	USES	(in minutes).	
changeOwner	CUSTOMER_STAT USES	The name of the Owner that should be set when applying the customer status to an account.	string
addDays	CUSTOMER_STAT USES	The number of days to increment the account next date by when applying the status to an account.	integer
addTime	CUSTOMER_STAT USES	The number of minutes to increment the account next date by when applying the status to an account.	integer
colour	CUSTOMER_STAT USES	The hexadecimal colour value that should be used to display account on the account list which have this customer status.	string
sortId	CUSTOMER_STAT USES	Used to determine the sort order of the customer status.	integer
priority	CUSTOMER_STAT USES	Used to determine the status priority.	integer
project	CUSTOMER_STAT USES	A list of projects that can use the customer status. Each project will be separated by a ' ' character. E.g. 'Sales Marketing'	string
autoDelay	CUSTOMER_STAT USES	The amount of delay, in seconds, to have before automatically dialling when entering an account with this customer status.	integer
autoDialOption	CUSTOMER_STAT USES	Whether or not to disable automatic dialling when entering an account.	string

Example Return Data

```
{
  "STATUS": "success",
  "CUSTOMER_STATUSES": [
    {
      "status": "Deal Lost",
      "subStatuses": [],
      "analysisGroup": "",
      "interval": 0,
      "changeOwner": "",
      "addDays": 0,
      "addTime": 0,
      "colour": "",
      "sortId": 4,
      "priority": 0,
      "project": "Sales",
      "autoDelay": 0,
      "autoDialOption": ""
    },
    {
      "status": "Lead",
      "subStatuses": [
        "New Lead",
        "Marketing Qualified",
        "Sales Qualified",
        "Call Back",
        "Cold",
        "Warm",
        "Hot",
        "Not Available"
      ],
      "analysisGroup": "Positive",
      "interval": 0,
      "changeOwner": "",
      "addDays": 1,
      "addTime": 30,
      "colour": "33FF63",
      "sortId": 5,
      "priority": 5,
      "project": "Project|Sales",
      "autoDelay": 10,
      "autoDialOption": "T"
    }
  ]
}
```

Failed Request Return Data

If the request is unsuccessful then the returned data will follow the structure below.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'error' when the request was unsuccessful.	string
ERROR	N/A	An array of errors for the request.	array

Example Return Data

```
{
  "STATUS": "error",
  "ERROR": [
    "The specified database area does not exist"
  ]
}
```

Error Messages

Below are the specific error messages which can be returned by this API mode endpoint. Please note that any global error messages can also be returned.

Message	Description
Unauthorised	The provided API authentication hash did not match any known hash in the system.
No database area has been specified	No 'area' parameter was specified in the request.
The specified database area does not exist	The 'area' parameter specified was not a valid database area.
You do not have the necessary user permissions to access this database area.	The user profile associated with the provided API hash does not have access to the database area requested.
An internal error occurred while trying to get the Customer Statuses.	An internal error occurred.

Javascript (with jQuery)

```
$(function() {  
  $.ajax(  
    {  
      url: 'yourSystemUrl/api2.php?' +  
        'mode=' + encodeURIComponent('getCustomerStatuses') +  
        '&area=' + encodeURIComponent('001'),  
      data: {  
        hash: '86ab15zaf95c3431ac68e115f515g1a0'  
      },  
      type: 'POST',  
      dataType: 'json',  
      success: function(response) {  
        //Manage the returned data.  
      }  
    }  
  );  
});
```

```
<?php

$baseUrl = 'yourSystemUrl/api2.php';

$getrequestData = array(
    'mode'      => 'getCustomerStatuses',
    'area'      => '001'
);

$postrequestData = array(
    'hash' => '86ab15zaf95c3431ac68e115f515g1a0',
);

$url = $baseUrl . '?' . http_build_query($getrequestData);
$curl = curl_init($url);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt(
    $curl,
    CURLOPT_POSTFIELDS,
    http_build_query($postrequestData)
);
$response = curl_exec($curl);
curl_close($curl);

$decodedResponse = json_decode($response, true);
//Manage the response using $decodedResponse
```

7.4 Mode - getOwners

This API mode endpoint can be used to retrieve the configured owners for a database area.

Request Data

Below are the available parameters which can be provided. These are in addition to the global parameters.

Parameter	Description	Data Type (length)	Parameter Type (GET/POST)	Required
area	The database area	string (3)	GET or POST	Yes

Example Request Data

```
{
  "area": "001",
  "hash": "89a0aa9a95d2b5bc48eeab2287ad65g7",
  "mode": "getOwners"
}
```

Request Return Data

The request can return one of two statuses. Each of these statuses depict the result of the request. The two statuses are as follows: 'success' and 'error'. Each status and returned data structure is outlined below.

Success Request Return Data

If the request is successful then the returned data will follow the structure below. The fields will be returned in the same order as configured in the system.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'success' when the request was successful.	string
OWNERS	N/A	An array of owners for the database area.	array
owner	OWNERS	The owner code.	string
name	OWNERS	The name of the owner.	string
id	OWNERS	Currently unused	integer
group	OWNERS	The group configured on the owner. Primarily used for reporting purposes.	string
active	OWNERS	Whether or not the owner is currently active.	boolean
email	OWNERS	The email address associated with the	string

		owner.	
googleUsername	OWNERS	The Google username/email address configured on the owner.	string
googleCalendar	OWNERS	Currently unused	string
googleOptions	OWNERS	Indicates the configured Google Options value.	integer
projectPermissions	OWNERS	The projects configured against the owner. Each project will be separated by a ' ' character.	string
area	OWNERS	The database area number that the owner belongs to.	string

Example Return Data

```
{
  "STATUS": "success",
  "OWNERS": [
    {
      "owner": "Jim",
      "name": "Jim McDonald",
      "id": 0,
      "group": "",
      "active": true,
      "email": "",
      "googleUsername": "",
      "googleCalendar": null,
      "googleOptions": "",
      "projectPermissions": "Project 1|Project 2",
      "area": "001"
    },
    {
      "owner": "John",
      "name": "John Blair",
      "id": 0,
      "group": "The Avengers",
      "active": true,
      "email": "",
      "googleUsername": "test@gmail.com",
      "googleCalendar": null,
      "googleOptions": "1",
      "projectPermissions": "",
      "area": "001"
    }
  ]
}
```


Failed Request Return Data

If the request is unsuccessful then the returned data will follow the structure below.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'error' when the request was unsuccessful.	string
ERROR	N/A	An array of errors for the request.	array

Example Return Data

```
{
  "STATUS": "error",
  "ERROR": [
    "The specified database area does not exist"
  ]
}
```

Error Messages

Below are the specific error messages which can be returned by this API mode endpoint. Please note that any global error messages can also be returned.

Message	Description
Unauthorised	The provided API authentication hash did not match any known hash in the system.
No database area has been specified	No 'area' parameter was specified in the request.
The specified database area does not exist	The 'area' parameter specified was not a valid database area.
You do not have the necessary user permissions to access this database area.	The user profile associated with the provided API hash does not have access to the database area requested.
An internal error occurred while trying to get the Owners.	An internal error occurred.

Javascript (with jQuery)

```
$(function() {
  $.ajax(
    {
      url: 'yourSystemUrl/api2.php?' +
        'mode=' + encodeURIComponent('getOwners') +
        '&area=' + encodeURIComponent('001'),
      data: {
        hash: '86ab15zaf95c3431ac68e115f515g1a0'
      },
      type: 'POST',
      dataType: 'json',
      success: function(response) {
        //Manage the returned data.
      }
    }
  );
});
```

```
<?php

$baseUrl = 'yourSystemUrl/api2.php';

$getRequestData = array(
    'mode'      => 'getOwners',
    'area'      => '001'
);

$postRequestData = array(
    'hash' => '86ab15zaf95c3431ac68e115f515g1a0',
);

$url = $baseUrl . '?' . http_build_query($getRequestData);
$curl = curl_init($url);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt(
    $curl,
    CURLOPT_POSTFIELDS,
    http_build_query($postRequestData)
);
$response = curl_exec($curl);
curl_close($curl);

$decodedResponse = json_decode($response, true);
//Manage the response using $decodedResponse
```

7.5 Mode - getProjects

This API mode endpoint can be used to retrieve the configured projects for a database area.

Request Data

Below are the available parameters which can be provided. These are in addition to the global parameters.

Parameter	Description	Data Type (length)	Parameter Type (GET/POST)	Required
area	The database area	string (3)	GET or POST	Yes

Example Request Data

```
{  
  "area": "001",  
  "hash": "89a0aa9a95d2b5bc48eeab2287ad65g7",  
  "mode": "getProjects"  
}
```

Request Return Data

The request can return one of two statuses. Each of these statuses depict the result of the request. The two statuses are as follows: 'success' and 'error'. Each status and returned data structure is outlined below.

Success Request Return Data

If the request is successful then the returned data will follow the structure below. The fields will be returned in the same order as configured in the system.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'success' when the request was successful.	string
PROJECTS	N/A	An array of projects for the database area.	array
project	PROJECTS	The project name/code.	string
description	PROJECTS	The project description.	string
startDate	PROJECTS	The project's start date.	string
endDate	PROJECTS	The project's end date.	string
dialPrefix	PROJECTS	The dialling prefix configured on the project.	string
defaultView	PROJECTS	The default system view to use when viewing accounts assigned to this project.	string
defaultScripts	PROJECTS	The default script to use when viewing accounts assigned to this project.	string
outboundIdentity	PROJECTS	The outbound CloudCall caller identity to use when dialling contacts from an account assigned to this project.	string

Example Return Data

```
{
  "STATUS": "success",
  "PROJECTS": [
    {
      "project": "Project 1",
      "description": "Project 1 description",
      "startDate": "2016-04-23 00:00:00",
      "endDate": "2030-12-31 00:00:00",
      "dialPrefix": "",
      "defaultView": "",
      "defaultScripts": "",
      "outboundIdentity": ""
    },
    {
      "project": "Project 2",
      "description": "Project 2 description",
      "startDate": "2017-04-22 00:00:00",
      "endDate": "2030-12-31 00:00:00",
      "dialPrefix": "",
      "defaultView": "Great Vista",
      "defaultScripts": "My little script",
      "outboundIdentity": "01234567890"
    }
  ]
}
```

Failed Request Return Data

If the request is unsuccessful then the returned data will follow the structure below.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'error' when the request was unsuccessful.	string
ERROR	N/A	An array of errors for the request.	array

Example Return Data

```
{
  "STATUS": "error",
  "ERROR": [
    "The specified database area does not exist"
  ]
}
```

Error Messages

Below are the specific error messages which can be returned by this API mode endpoint. Please note that any global error messages can also be returned.

Message	Description
Unauthorised	The provided API authentication hash did not match any known hash in the system.
No database area has been specified	No 'area' parameter was specified in the request.
The specified database area does not exist	The 'area' parameter specified was not a valid database area.
You do not have the necessary user permissions to access this database area.	The user profile associated with the provided API hash does not have access to the database area requested.
An internal error occurred while trying to get the Projects.	An internal error occurred.

Javascript (with jQuery)

```
$(function() {
  $.ajax(
    {
      url: 'yourSystemUrl/api2.php?' +
        'mode=' + encodeURIComponent('getProjects') +
        '&area=' + encodeURIComponent('001'),
      data: {
        hash: '86ab15zaf95c3431ac68e115f515g1a0'
      },
      type: 'POST',
      dataType: 'json',
      success: function(response) {
        //Manage the returned data.
      }
    }
  );
});
```



```
<?php

$baseUrl = 'yourSystemUrl/api2.php';

$getrequestData = array(
    'mode'      => 'getProjects',
    'area'      => '001'
);

$postrequestData = array(
    'hash' => '86ab15zaf95c3431ac68e115f515g1a0',
);

$url = $baseUrl . '?' . http_build_query($getrequestData);
$curl = curl_init($url);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt(
    $curl,
    CURLOPT_POSTFIELDS,
    http_build_query($postrequestData)
);
$response = curl_exec($curl);
curl_close($curl);

$decodedResponse = json_decode($response, true);
//Manage the response using $decodedResponse
```

7.6 Mode - getSalesPeople

This API mode endpoint can be used to retrieve the configured projects for a database area.

Request Data

Below are the available parameters which can be provided. These are in addition to the global parameters.

Parameter	Description	Data Type (length)	Parameter Type (GET/POST)	Required
area	The database area	string (3)	GET or POST	Yes

Example Request Data

```
{
  "area": "001",
  "hash": "89a0aa9a95d2b5bc48eeab2287ad65g7",
  "mode": "getSalesPeople"
}
```

Request Return Data

The request can return one of two statuses. Each of these statuses depict the result of the request. The two statuses are as follows: 'success' and 'error'. Each status and returned data structure is outlined below.

Success Request Return Data

If the request is successful then the returned data will follow the structure below. The fields will be returned in the same order as configured in the system.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'success' when the request was successful.	string
SALES_PEOPLE	N/A	An array of sales people for the database area.	array
name	SALES_PEOPLE	The sales person's name.	string
emailAddress	SALES_PEOPLE	The sales person's email address.	string
salesArea	SALES_PEOPLE	The sales person's sales area. Primarily used for reporting purposes.	string
owner	SALES_PEOPLE	The owner code of the owner to associate with the sales person.	string
postCodeRegions	SALES_PEOPLE	An array of postcode regions configured against the sales person.	array
allowableProjects	SALES_PEOPLE	An array of any project set against the sales person.	array
gmtDifference	SALES_PEOPLE	The difference (in minutes) of the sales person's timezone when compared to GMT.	integer
country	SALES_PEOPLE	The sales person's country. "0" means that no country has been configured.	string
googleUsername	SALES_PEOPLE	The sales person's Google username/email address.	string
googleOptionsValue	SALES_PEOPLE	Indicates the configured Google Options value.	integer

Example Return Data

```
{
  "STATUS": "success",
  "SALES_PEOPLE": [
    {
      "name": "Catherine",
      "emailAddress": "catherine@my-company.com",
      "salesArea": "Welders",
      "owner": "Catherine",
      "postCodeRegions": [],
      "allowableProjects": [
        "Project A"
      ],
      "gmtDifference": 0,
      "country": "United Kingdom",
      "googleUsername": "catherine@gmail.com",
      "googleOptionsValue": 1
    },
    {
      "name": "Jared",
      "emailAddress": "",
      "salesArea": "",
      "owner": "",
      "postCodeRegions": [],
      "allowableProjects": [
        "aaaaa"
      ],
      "gmtDifference": 100,
      "country": "0",
      "googleUsername": "",
      "googleOptionsValue": 0
    }
  ]
}
```

Failed Request Return Data

If the request is unsuccessful then the returned data will follow the structure below.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'error' when the request was unsuccessful.	string
ERROR	N/A	An array of errors for the request.	array

Example Return Data

```
{
  "STATUS": "error",
  "ERROR": [
    "The specified database area does not exist"
  ]
}
```

Error Messages

Below are the specific error messages which can be returned by this API mode endpoint. Please note that any global error messages can also be returned.

Message	Description
Unauthorised	The provided API authentication hash did not match any known hash in the system.
No database area has been specified	No 'area' parameter was specified in the request.
The specified database area does not exist	The 'area' parameter specified was not a valid database area.
You do not have the necessary user permissions to access this database area.	The user profile associated with the provided API hash does not have access to the database area requested.
An internal error occurred while trying to get the sales people.	An internal error occurred.

Javascript (with jQuery)

```
$(function() {  
  $.ajax(  
    {  
      url: 'yourSystemUrl/api2.php?' +  
        'mode=' + encodeURIComponent('getSalesPeople') +  
        '&area=' + encodeURIComponent('001'),  
      data: {  
        hash: '86ab15zaf95c3431ac68e115f515g1a0'  
      },  
      type: 'POST',  
      dataType: 'json',  
      success: function(response) {  
        //Manage the returned data.  
      }  
    }  
  );  
});
```

```
<?php

$baseUrl = 'yourSystemUrl/api2.php';

$getrequestData = array(
    'mode'      => 'getSalesPeople',
    'area'      => '001'
);

$postrequestData = array(
    'hash' => '86ab15zaf95c3431ac68e115f515g1a0',
);

$url = $baseUrl . '?' . http_build_query($getrequestData);
$curl = curl_init($url);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt(
    $curl,
    CURLOPT_POSTFIELDS,
    http_build_query($postrequestData)
);
$response = curl_exec($curl);
curl_close($curl);

$decodedResponse = json_decode($response, true);
//Manage the response using $decodedResponse
```

8 Account information

8.1 Mode: importCustomerProducts

This API mode endpoint can be used to import new Customer Products into the system.

Request Data

Below are the available parameters which can be provided. These are in addition to the global parameters.

Parameter	Description	Data Type (length)	Parameter Type (GET/POST)	Required
area	The database area	string (3)	POST	Yes
customerProducts	The Customer Products to import.	array	POST	Yes

Each Customer Product within the customerProducts array must contain the required fields of 'account', 'product' and 'manufacturer'. Below are the available parameters that can be passed on each Customer Product.

Parameter	Description	Data Type (length)	Parameter Type (GET/POST)	Required
account	The account the Customer Product relates to	string (32)	POST	Yes
product	The product	string (50)	POST	Yes, however it can be left as empty but not null
manufacturer	The manufacturer of the product	string (25)	POST	Yes, however it can be left as empty but not null
reference	A unique reference for the Customer Product	string (150)	POST	No. If omitted, then a reference will be generated using the 'Next Product Reference' within the database area setup
contact	The contact that the product relates to	string (50)	POST	No

posn	The contact's job title	string (255)	POST	No
importance	The perceived importance of the Customer Product	string (255)	POST	No
company	The company the Customer product relates to	string (255)	POST	No
startDate	The starting date and time for the Customer Product, often used for contracts.	datetime	POST	No. If provided, the date must be in the format of YYYY-MM-DD hh:mm:ss
endDate	The ending date and time for the Customer Product, often used for contracts	datetime	POST	No. If provided, the date must be in the format of YYYY-MM-DD hh:mm:ss
planQty	The anticipated quantity of the Product that will be required	float	POST	No. If omitted will be set to 0
actQty	The actual quantity of the Product that will be required	float	POST	No. If omitted will be set to 0
planVal	The anticipated value of the Product(s)	float	POST	No. If omitted will be set to 0
actVal	The actual value of the Product(s)	float	POST	No. If omitted will be set to 0
type_	The type of the Customer Product	string (5)	POST	No
status	The status of the Customer Product	string (20)	POST	No
text	Any additional information	string (65k)	POST	No
updatedBy	The username who updated the Customer Product	string (50)	POST	No. If omitted then it will be populated by the username making the API request
updated	The date and time the Customer Product was updated	datetime	POST	No. If omitted then it will be populated by the current date and time. If provided, the date must

				be in the format of YYYY-MM-DD hh:mm:ss
updateFlag	A custom flag that can be applied to the Customer Product	string (10)	POST	No
nextDate	The next date for the Customer Product	datetime	POST	No. If provided, the date must be in the format of YYYY-MM-DD hh:mm:ss
callInterval	The frequency of the call interval for the Customer Product	integer	POST	No
probability	The probability the Customer Product will be sold/acquired	integer	POST	No. If omitted will be set to 0
created	The date and time the Customer Product was created	datetime	POST	No. If omitted then it will be populated by the current date and time. If provided, the date must be in the format of YYYY-MM-DD hh:mm:ss
addn01 -> 50	The additional fields for the Customer Product e.g. addn01, addn17	string (250)	POST	No
project	The project the Customer Product belongs to	string (50)	POST	No
cid	The contact id for the contact the Customer Product relates to	integer	POST	No. If omitted then the contact's id will be retrieved and used
totalNetValue	The total net value of the Customer Product	float	POST	No. If omitted will be set to 0
taxValue	The amount of tax for the Customer Product	float	POST	No. If omitted will be set to 0
totalValue	The total value of the Customer Product	float	POST	No. If omitted will be set to 0

discount	The discount for the Customer Product	float	POST	No. If omitted will be set to 0
prodGroup	The product group the product belongs to	string (25)	POST	No
frequency	How frequent the Customer Product will be required	string (50)	POST	No

Example Request Data

```
{
  "area": "001",
  "hash": "89a0aa9a95d2b5bc48eeab2287ad65g7",
  "mode": "importCustomerProducts",
  "customerProducts": [
    {
      "account": "1",
      "product": "Chair",
      "manufacturer": "Woodworks",
      "planQty": 5,
      "planVal": 60
    },
    {
      "account": "5",
      "product": "Table",
      "manufacturer": "Woodworks",
      "updatedBy": "Joe",
      "addn01": "Blue"
    }
  ]
}
```

Request Return Data

An importCustomerProducts request can return one of three statuses. Each of these statuses depict the result of the request. The three statuses are as follows: 'success', 'error' and 'success with errors'. Each status and returned data structure is outlined below.

Success Request Return Data

If the request is successful then the returned data will follow the structure below.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'success' when the request was successful.	string
CUSTOMER_PRODUCTS	N/A	An array of the Customer Products imported as part of the request.	array
SUCCESSFUL	CUSTOMER_PRODUCTS	An array of the Customer Products which have been successfully imported.	array
UNSUCCESSFUL	CUSTOMER_PRODUCTS	An array of the Customer Products which have been not been successfully imported. This will still be returned on a successful request, however it will be an empty array.	array
CUSTOMER_PRODUCT	SUCCESSFUL and UNSUCCESSFUL	The Customer Product details.	array

Example Return Data

```
{
  "STATUS": "success",
  "CUSTOMER_PRODUCTS": {
    "SUCCESSFUL": [
      {
        "CUSTOMER_PRODUCT": {
          "account": "1",
          "product": "Chair",
          "manufacturer": "Woodworks"
          ...
        }
      }
    ],
    "UNSUCCESSFUL": []
  }
}
```

Success with Errors Request Return Data

If the request had any Customer Products which were not imported successfully, and there were no base level errors such as 'No database area specified', then a 'success with errors' status will be returned. The returned data will follow the structure below.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'success with errors' when the request was successful, but some errors occurred.	string
CUSTOMER_PRODUCTS	N/A	An array of the Customer Products imported as part of the request.	array
SUCCESSFUL	CUSTOMER_PRODUCTS	An array of the Customer Products which have been successfully imported.	array
UNSUCCESSFUL	CUSTOMER_PRODUCTS	An array of the Customer Products which have been not been successfully imported.	array
CUSTOMER_PRODUCT	SUCCESSFUL and UNSUCCESSFUL	The Customer Product with the details that have been successfully imported or have not been successfully imported.	array
ERROR	UNSUCCESSFUL	An array of errors for the particular Customer Product they are against.	array

Example Return Data

```
{
  "STATUS": "success with errors",
  "CUSTOMER_PRODUCTS": {
    "SUCCESSFUL": [
      {
        "CUSTOMER_PRODUCT": {
          "account": "1",
          "product": "Chair",
          "manufacturer": "Woodworks"
          ...
        }
      }
    ],
    "UNSUCCESSFUL": [
      {
        "CUSTOMER_PRODUCT": {
          "account": "1",
          "product": "Chair",
          "reference": "1007"
          ...
        },
        "ERRORS": [
          "Reference (1007) already in use in area (001)",
          "Manufacturer cannot be null"
        ]
      }
    ]
  }
}
```

Failed Request Return Data

If the request is unsuccessful then the returned data will follow the structure below.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'error' when the request was unsuccessful.	string
ERROR	N/A	An array of errors for the request.	array

Example Return Data

```
{
  "STATUS": "error",
  "ERROR": [
    "The specified database area does not exist"
  ]
}
```

Error Messages

Below are the specific error messages which can be returned by this API mode endpoint. Please note that any global error messages can also be returned.

Message	Description
Unauthorised	The provided API authentication hash did not match any known hash in the system.
No database area has been specified	No 'area' parameter was specified in the request.
The specified database area does not exist	The 'area' parameter specified was not a valid database area.
Invalid Customer Products data provided.	The Customer Products provided are not in the expected structure.
No POST element (customerProducts) found as part of the request	No Customer Products are provided with the request
Invalid date provided for <i>field (date)</i> . Dates should match the following format YYYY-MM-DD hh:mm:ss (<i>date</i>)	An invalid date is provided for a date field.
<i>Field</i> cannot be null	A field which cannot be null has been set as null or omitted from the request.
<i>Field</i> cannot be empty	A field which cannot be empty has been set as empty.
Reference (<i>reference</i>) already in use in area (<i>area</i>)	The provided reference is already in use by another Customer Product.
Account (<i>account</i>) does not exist in database area (<i>area</i>)	The provided account does not exist in the database area specified.
Contact (<i>contact</i>) does not exist on account (<i>account</i>) in database area (<i>area</i>)	The provided contact does not exist on the provided account.

The combination of account (<i>account</i>), product (<i>product</i>), manufacture (manufacturer), reference (<i>reference</i>) is already in use by another Customer Product in area (<i>area</i>).	The provided combination of account, product, manufacturer, reference are already in use in the provided database area.
Field (<i>field</i>) is longer than the maximum length allowed (<i>field length</i>)	The provided field value is longer than the allowed limit for that field.
An internal error occurred while trying to import the Customer Products.	An internal error occurred.

Code Examples

Javascript (with jQuery)

```
$(function() {
    $.ajax(
        {
            url: 'yourSystemUrl/api2.php',
            data: {
                hash: '86ab15zaf95c3431ac68e115f515g1a0',
                mode: 'importCustomerProducts',
                area: '001',
                customerProducts: [
                    {
                        account: '1',
                        product: 'Chair',
                        manufacturer: 'Woodworking',
                        created: '2018-01-01 01:00:00',
                        planQty: 12,
                        planVal: 200,
                        addn01: 'Blue',
                        addn45: 'Large',
                    },
                    {
                        account: '1',
                        product: 'Table',
                        manufacturer: 'Woodworking'
                    }
                ]
            },
            type: 'POST',
            dataType: 'json',
            success: function(response) {
                //Manage the returned data.
            }
        }
    );
});
```

PHP

```
<?php

$url = 'yourSystemUrl/api2.php';

$postRequestData = array(
    'hash'          => '86ab15zaf95c3431ac68e115f515g1a0',
    'mode'          => 'importCustomerProducts',
    'area'          => '007',
    'customerProducts' => array(
        array(
            'account'      => '1',
            'product'     => 'Chair',
            'manufacturer' => 'Woodworking',
            'created'     => '2018-01-01 01:00:00',
            'planQty'     => 12,
            'planVal'     => 200,
            'addn01'     => 'Blue',
            'addn45'     => 'Large',
        ),
        array(
            'account'      => '1',
            'product'     => 'Table',
            'manufacturer' => 'Woodworking'
        )
    )
);

$curl = curl_init($url);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt(
    $curl,
    CURLOPT_POSTFIELDS,
    http_build_query($postRequestData)
);
$response = curl_exec($curl);
curl_close($curl);

$decodedResponse = json_decode($response, true);
//Manage the response using $decodedResponse
```

8.2 Mode - updateCustomerProducts

This API mode endpoint can be used to update pre-existing Customer Products in the system.

Request Data

Below are the available parameters which can be provided. These are in addition to the global parameters.

Parameter	Description	Data Type (length)	Parameter Type (GET/POST)	Required
area	The database area	string (3)	POST	Yes
customerProducts	The Customer Products to update.	array	POST	Yes

Each Customer Product within the customerProducts array must each contain the required fields of 'account', 'product', 'manufacturer' and 'reference'. A matching Customer Product will be found and then updated with the information provided within the customerProducts array. If a field is not provided within the request, then the pre-existing value will be kept (excluding the parameters 'updated' and 'updatedBy').

Parameter	Description	Data Type (length)	Parameter Type (GET/POST)	Required
account	The account the Customer Product relates to	string (32)	POST	Yes
product	The product	string (50)	POST	Yes
manufacturer	The manufacturer of the product	string (25)	POST	Yes
reference	A unique reference for the Customer Product	string (150)	POST	Yes
contact	The contact that the Customer Product relates to	string (50)	POST	No
posn	The contact's job title	string (255)	POST	No
importance	The perceived importance of the Customer Product	string (255)	POST	No
company	The company the Customer product relates to	string (255)	POST	No
startDate	The starting date and time for the Customer Product, often used for contracts.	datetime	POST	No. If provided, the date must be in the format of YYYY-

				MM-DD hh:mm:ss
endDate	The ending date and time for the Customer Product, often used for contracts.	datetime	POST	No. If provided, the date must be in the format of YYYY-MM-DD hh:mm:ss
planQty	The anticipated quantity of the Product that will be required	float	POST	No
actQty	The actual quantity of the Product that will be required	float	POST	No
planVal	The anticipated value of the Product(s)	float	POST	No
actVal	The actual value of the Product(s)	float	POST	No
type_	The type of the Customer Product	string (5)	POST	No
status	The status of the Customer Product	string (20)	POST	No
text	Any additional information	string (65k)	POST	No
updatedBy	The username who updated the Customer Product	string (50)	POST	No. If omitted then it will be populated by the username making the API request
updated	The date and time the Customer Product was updated	datetime	POST	No. If omitted then it will be populated by the current date and time. If provided, the date must be in the format of YYYY-MM-DD hh:mm:ss
updateFlag	A custom flag that can be	string (10)	POST	No

	applied to the Customer Product			
nextDate	The next date for the Customer Product	datetime	POST	No. If provided, the date must be in the format of YYYY-MM-DD hh:mm:ss
callInterval	The frequency of the call interval for the Customer Product	integer	POST	No
probability	The probability the Customer Product will be sold/acquired	integer	POST	No
created	The date and time the Customer Product was created	datetime	POST	No. If provided, the date must be in the format of YYYY-MM-DD hh:mm:ss
addn01 -> 50	The additional fields for the Customer Product e.g. addn01, addn17	string (250)	POST	No
project	The project the Customer Product belongs to	string (50)	POST	No
cid	The contact id for the contact the Customer Product relates to	integer	POST	No
totalNetValue	The total net value of the Customer Product	float	POST	No
taxValue	The amount of tax for the Customer Product	float	POST	No
totalValue	The total value of the Customer Product	float	POST	No
discount	The discount for the Customer Product	float	POST	No
prodGroup	The product group the product belongs to	string (25)	POST	No
frequency	How frequent the Customer Product will be required	string (50)	POST	No

Example Request Data

```
{
  "area": "001",
  "hash": "89a0aa9a95d2b5bc48eeab2287ad65g7",
  "mode": "updateCustomerProducts",
  "customerProducts": [
    {
      "account": "57A",
      "product": "Screws",
      "manufacturer": "The Metal Company",
      "reference": 1387,
      "actQty": 84,
      "actVal": 1900
    },
    {
      "account": "5",
      "product": "Table",
      "manufacturer": "Woodworks",
      "reference": 901,
      "updatedBy": "Joe",
      "addn34": "Red"
    }
  ]
}
```

Request Return Data

An updateCustomerProducts request can return one of three statuses. Each of these statuses depict the result of the request. The three statuses are as follows: 'success', 'error' and 'success with errors'. Each status and returned data structure is outlined below.

Success Request Return Data

If the request is successful then the returned data will follow the structure below.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'success' when the request was successful.	string
CUSTOMER_PRODUCTS	N/A	An array of the Customer Products that have been used to update pre-existing Customer Products.	array
SUCCESSFUL	CUSTOMER_PRODUCTS	An array of the Customer Products which have been successfully updated.	array
UNSUCCESSFUL	CUSTOMER_PRODUCTS	An array of the Customer Products which have been not been successfully updated. This will still be returned on a successful request, however it will be an empty array.	array
CUSTOMER_PRODUCT	SUCCESSFUL and UNSUCCESSFUL	The Customer Product with the updated values.	array

Example Return Data

```
{
  "STATUS": "success",
  "CUSTOMER_PRODUCTS": {
    "SUCCESSFUL": [
      {
        "CUSTOMER_PRODUCT": {
          "account": "57A",
          "product": "Screws",
          "manufacturer": "The Metal Company",
          "reference": 1387
          ...
        }
      }
    ],
    "UNSUCCESSFUL": []
  }
}
```

Success with Errors Request Return Data

If the request had any Customer Products which were not updated successfully, and there were no base level errors such as 'No database area specified', then a 'success with errors' status will be returned. The returned data will follow the structure below.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'success with errors' when the request was successful, but some errors occurred.	string
CUSTOMER_PRODUCTS	N/A	An array of the Customer Products updated as part of the request.	array
SUCCESSFUL	CUSTOMER_PRODUCTS	An array of the Customer Products which have been successfully updated.	array
UNSUCCESSFUL	CUSTOMER_PRODUCTS	An array of the Customer Products which have been not been successfully updated.	array
CUSTOMER_PRODUCT	SUCCESSFUL and UNSUCCESSFUL	The Customer Product with the details that have been successfully updated or have not been successfully updated.	array
ERROR	UNSUCCESSFUL	An array of errors for the particular Customer Product they are against.	array

Example Return Data

```
{
  "STATUS": "success with errors",
  "CUSTOMER_PRODUCTS": {
    "SUCCESSFUL": [
      {
        "CUSTOMER_PRODUCT": {
          "account": "57A",
          "product": "Screws",
          "manufacturer": "The Metal Company",
          "reference": 1387
          ...
        }
      }
    ],
    "UNSUCCESSFUL": [
      {
        "CUSTOMER_PRODUCT": {
          "account": "1",
          "product": "Chair",
          "reference": 1007
          ...
        },
        "ERRORS": [
          "No manufacturer specified on the Customer Product"
        ]
      }
    ]
  }
}
```

Failed Request Return Data

If the request is unsuccessful then the returned data will follow the structure below.

Field	Parent	Description	Data Type
STATUS	N/A	Whether or not the request was successful. 'error' when the request was unsuccessful.	string
ERROR	N/A	An array of errors for the request.	array

Example Return Data

```
{
  "STATUS": "error",
  "ERROR": [
    "The specified database area does not exist"
  ]
}
```

Error Messages

Below are the specific error messages which can be returned by this API mode endpoint. Please note that any global error messages can also be returned.

Message	Description
Unauthorised	The provided API authentication hash did not match any known hash in the system.
No database area has been specified	No 'area' parameter was specified in the request.
The specified database area does not exist	The 'area' parameter specified was not a valid database area.
Invalid Customer Products data provided.	The Customer Products provided are not in the expected structure.
No POST element (customerProducts) found as part of the request	No Customer Products are provided with the request
No <i>field</i> specified on the Customer Product	When one of the required fields is not passed within a Customer Product.
No matching Customer Product could be found for the following details: account (<i>account</i>), product (<i>product</i>), manufacturer (<i>manufacturer</i>), reference (<i>reference</i>) in area (<i>area</i>).	The provided key fields could not be matched to an existing Customer Product.

No pre-existing Customer Product found with the reference (<i>reference</i>) in area (<i>area</i>)	No matching Customer Product could be found with the same reference.
Invalid date provided for <i>field</i> (<i>date</i>). Dates should match the following format YYYY-MM-DD hh:mm:ss (<i>date</i>)	An invalid date is provided for a date field.
<i>Field</i> cannot be null	A field which cannot be null has been set as null or omitted from the request.
<i>Field</i> cannot be empty	A field which cannot be empty has been set as empty.
Field (<i>field</i>) is longer than the maximum length allowed (<i>field length</i>)	The provided field value is longer than the allowed limit for that field.
Account (<i>account</i>) does not exist in database area (<i>area</i>)	The provided account does not exist in the database area specified.
Contact (<i>contact</i>) does not exist on account (<i>account</i>) in database area (<i>area</i>)	The provided contact does not exist on the provided account.
An internal error occurred while trying to import the Customer Products.	An internal error occurred.

Code Examples

Javascript (with jQuery)

```
$(function() {
  $.ajax(
    {
      url: 'yourSystemUrl/api2.php',
      data: {
        hash: '86ab15zaf95c3431ac68e115f515g1a0',
        mode: 'updateCustomerProducts',
        area: '001',
        customerProducts: [
          {
            account: '1',
            product: 'Chair',
            manufacturer: 'Woodworking',
            reference: '1007',
            planQty: 12,
            planVal: 200,
            addn01: 'Blue',
            addn45: 'Large'
          },
          {
            account: '1',
            product: 'Table',
            manufacturer: 'Woodworking',
            reference: '1009',
            updatedBy: 'John',
            contact: 'Chad Buckley'
          }
        ]
      },
      type: 'POST',
      dataType: 'json',
      success: function(response) {
        //Manage the returned data.
      }
    }
  );
});
```

```
<?php

$url = 'yourSystemUrl/api2.php';

$postRequestData = array(
    'hash'          => '86ab15zaf95c3431ac68e115f515g1a0',
    'mode'          => 'updateCustomerProducts',
    'area'          => '007',
    'customerProducts' => array(
        array(
            'account'      => '1',
            'product'      => 'Chair',
            'manufacturer' => 'Woodworking',
            'reference'    => '1007',
            'planQty'      => 12,
            'planVal'      => 200,
            'addn01'       => 'Blue',
            'addn45'       => 'Large',
        ),
        array(
            'account'      => '1',
            'product'      => 'Table',
            'manufacturer' => 'Woodworking',
            'reference'    => '1009',
            'updatedBy'    => 'John',
            'contact'      => 'Chad Buckley'
        )
    )
);

$curl = curl_init($url);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_POST, true);
curl_setopt(
    $curl,
    CURLOPT_POSTFIELDS,
    http_build_query($postRequestData)
);
$response = curl_exec($curl);
curl_close($curl);

$decodedResponse = json_decode($response, true);
//Manage the response using $decodedResponse
```