



CloudSponge.com Address Book API

Integration Guide

Release 1.4

April 13, 2010

Document Revision History

Publish Date	Software Release	Comments
13-Apr-10	1.4	<p><u>Production Date: 13-Apr-10</u></p> <p>Updated Begin Import to POST RESTful API method</p> <p>Added contacts owner to Contacts Response</p> <ul style="list-style-type: none"> contacts_owner
30-Mar-10	1.3.2	<p><u>Production Date: 30-Mar-10</u></p> <p>Update API domain endpoint</p> <ul style="list-style-type: none"> https://api.cloudsponge.com/
16-Mar-10	1.3.1	<p><u>Production Date: 16-Mar-10</u></p> <p>Updated API parameters</p> <ul style="list-style-type: none"> import_id domain_key domain_password user_id <p>Updated import unique identifier for asynchronous process</p> <ul style="list-style-type: none"> import_id <p>Updated API authentication</p> <ul style="list-style-type: none"> domain_key + domain_password
29-Jan-10	1.3	<p><u>Production Date: 15-Jan-10</u></p> <p>Re-organized API with 3 new Begin Import endpoints based on User Authentication methods:</p> <ul style="list-style-type: none"> User Consent Import Desktop Applet <p>Added OAuth support for following services:</p> <ul style="list-style-type: none"> Yahoo Gmail
17-Dec-09	1.2	<p><u>Production Date: 17-Dec-09</u></p> <p>Added AOL as an additional address book contact source.</p>
15-Sep-09	1.1	<p><u>Production Date: 14-Sep-09</u></p> <p>Added the Direct Database Connection feature to allow users to retrieve contacts directly from the CloudSponge.com database, rather than through an API retrieval method.</p>
01-Sep-09	1.0	<p><u>Production Date: 21-Aug-09</u></p> <p>Initial release of the CloudSponge.com Address Book API includes:</p> <ul style="list-style-type: none"> Begin Import Fetch Events Fetch Contacts <p>Supported address book sources include:</p> <ul style="list-style-type: none"> Yahoo Windows Live Gmail Plaxo Microsoft Outlook

		▪ Mac Address Book
--	--	--------------------

Table of Contents

1. Address Book API	5
1.1. Asynchronous Process	5
1.2. User Authentication	5
1.2.1. User Consent	6
1.2.2. Import	7
1.2.3. Desktop Applet	8
1.3. API Authentication	9
1.4. Retrieving Contacts	9
2. Integration Checklist	9
2.1. Getting Ready	9
2.2. Server-side Dependencies	10
2.2.1. Initiating an Import.....	10
2.2.2. Formatting the Response for User's Browser	10
2.2.3. Polling Events.....	11
2.2.4. Retrieving Contacts	13
2.3. Release into Production	13
Appendix A: HTTP Response Codes	14
Appendix B: Address Book API Reference	15
Begin Import: User Consent.....	15
Begin Import: Import	16
Begin Import: Desktop Applet	17
Events.....	18
Contacts	20
Appendix C: Direct Database Connection	22

1. Address Book API

The Cloudsponge.com Address Book API imports address book contacts from multiple sources through a single integration point. The API is compatible with the most popular web programming languages. This Integration Guide outlines the technical details required to integrate the API with your application.

1.1. Asynchronous Process

Despite there being different user authentication methods, importing contacts with the API follows a similar 3-stage asynchronous process.

- 1. Request:** Your application initiates an address book import by calling the "Begin Import" URL, specifying the user authentication method, the address book source, plus any other additional information required. The API returns a result object containing a unique identifier for subsequent calls to the API, and in some cases data needs to be displayed in the user's browser.
- 2. Notify:** Your application polls the "Events" URL and receives a result object indicating the current Import Events. These events notify you of the current progress of the import, if any errors have occurred, or when the import is complete. You can optionally display these Import Events to the user's browser.
- 3. Complete:** Once you have received a "complete" Import Event, you can retrieve the contacts payload from the API directly or from a database configured for your account.

1.2. User Authentication

Before integrating, it is important to understand the user authentication required to retrieve contacts, as the interaction and information required by the user differs among the different address books. CloudSponge.com categorizes user authentication by the following:

- 1. User Consent:** User grants consent to retrieve specific data (i.e. contacts) from the source. An authentication token is granted by the data source, authorizing access for the application to retrieve the specific data user has granted consent.
- 2. Import:** User enters user ID and password, allowing access to retrieve contacts.
- 3. Desktop Applet:** User downloads an applet to their system and allows access for the applet to retrieve contacts directly from their system.

The following table outlines the address books supported by user authentication method.

User Consent	Import	Desktop Applet
<ul style="list-style-type: none">▪ Yahoo▪ Windows Live▪ Gmail	<ul style="list-style-type: none">▪ Gmail▪ AOL▪ Plaxo	<ul style="list-style-type: none">▪ Mac Address Book▪ Microsoft Outlook

Since the user authentication differs by contact source, the API process is described in further detail over the next 3 sections.

1.2.1. User Consent

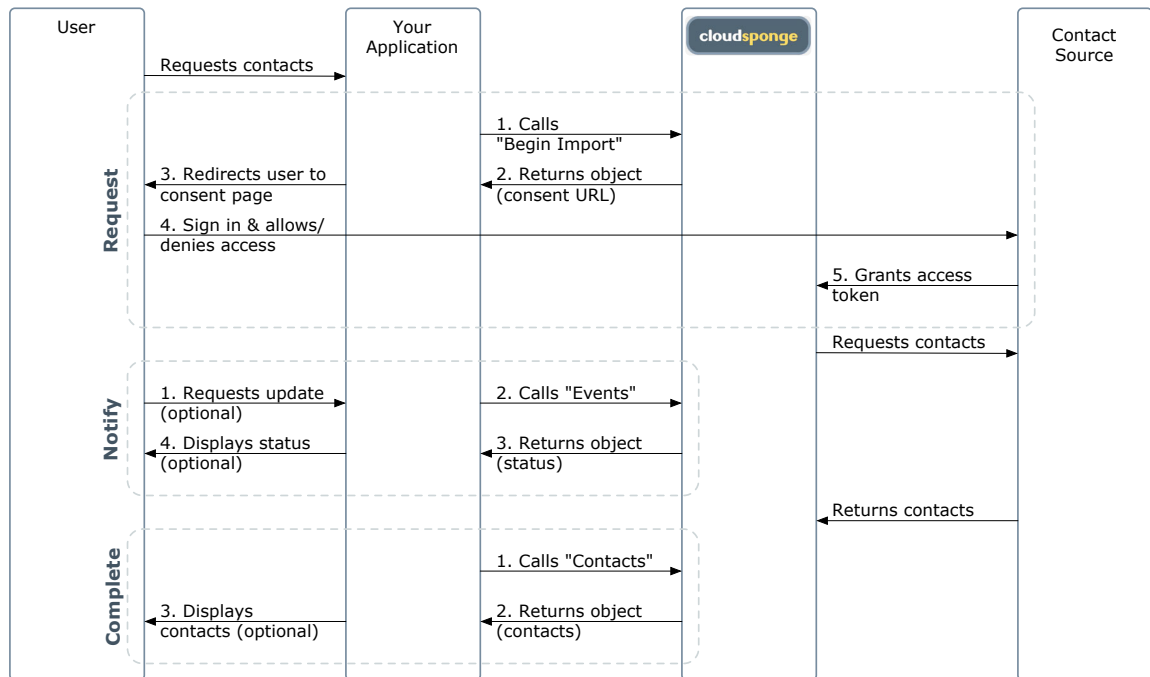


Figure 1.2.1: User Consent

Request

1. Your application calls the "Begin Import" URL, including the contact source requested.
2. CloudSponge.com returns a result object, including the `import_id` to use to fetch the contacts later and a URL for the user consent page, which needs to be displayed to the user.
3. The user is redirected to the consent page URL. (Optional: display in a new tab or window).
4. User signs in with their username and password for the respective source. User allows or denies access to the application requesting access to their contacts.
5. Contact Source grants an access token for CloudSponge.com to retrieve the user's contacts.

Notify

1. (Optional) User's browser requests a status update on the contact import.
2. Your application calls the "Events" URL to get the current Import Event.
3. CloudSponge.com returns the current Event result object (refer to section 2.2 Events and Contacts for event status codes).
4. (Optional) Your application displays a status update to the user.

Complete

1. Once a successful "Complete" event type has been received from "Notify" above, your application calls the "Contacts" URL to retrieve the contacts, specifying the `import_id` from the initial request.
2. CloudSponge.com returns the Contacts result object in the format requested.
3. (Optional) Your application displays the contacts to the user.

1.2.2. Import

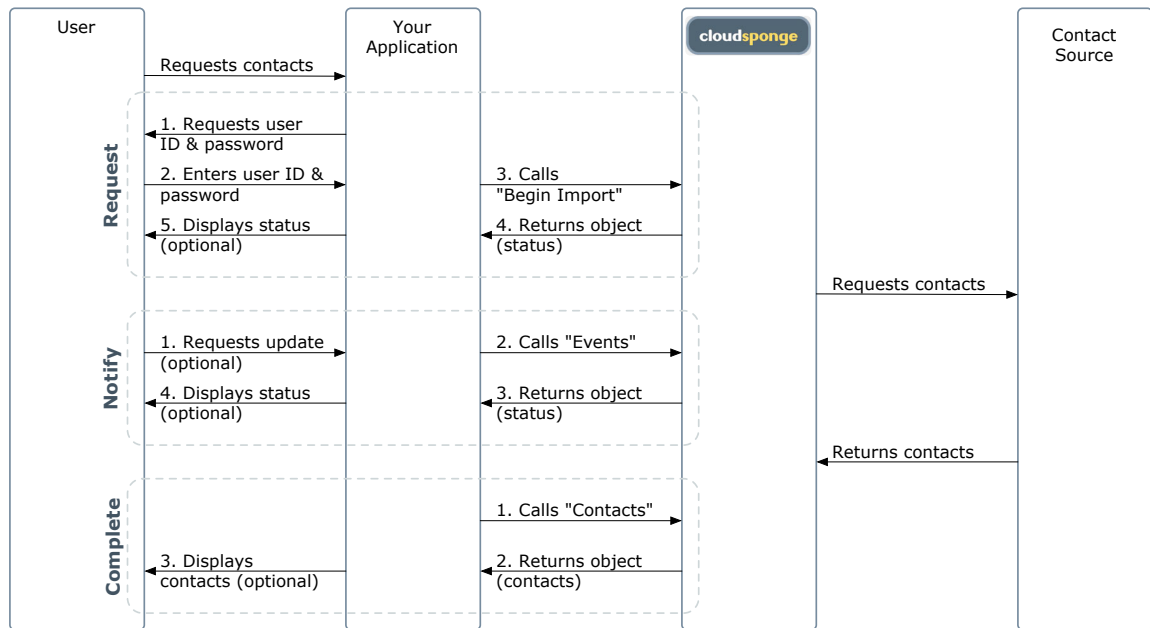


Figure 1.2.2: Import

Request

1. Your application requests the user to enter their user ID and password combination for the contact source requested.
2. User enters their user ID and password combination.
3. Your application calls the "Begin Import" URL, including the user ID/password plus the contact source requested.
4. CloudSponge.com returns a result object indicating the request has been received and specifying an identifier to use for fetching events and contacts.
5. (Optional) Your application displays a status update to the user.

Notify

1. (Optional) User's browser requests a status update on the contact import.
2. Your application calls the "Events" URL to get the current Import Events, specifying the import_id returned in the initial request.
3. CloudSponge.com returns the current Event result object (refer to section 2.2 Events and Contacts for event status codes).
4. (Optional) Your application displays a status update to the user.

Complete

1. Once a successful "Complete" event has been received from "Notify" above, your application calls the "Contacts" URL to retrieve the contacts, specifying the import_id returned in the initial request.
2. CloudSponge.com returns the Contacts result object in the format requested.
3. (Optional) Your application displays the contacts to the user.

1.2.3. Desktop Applet

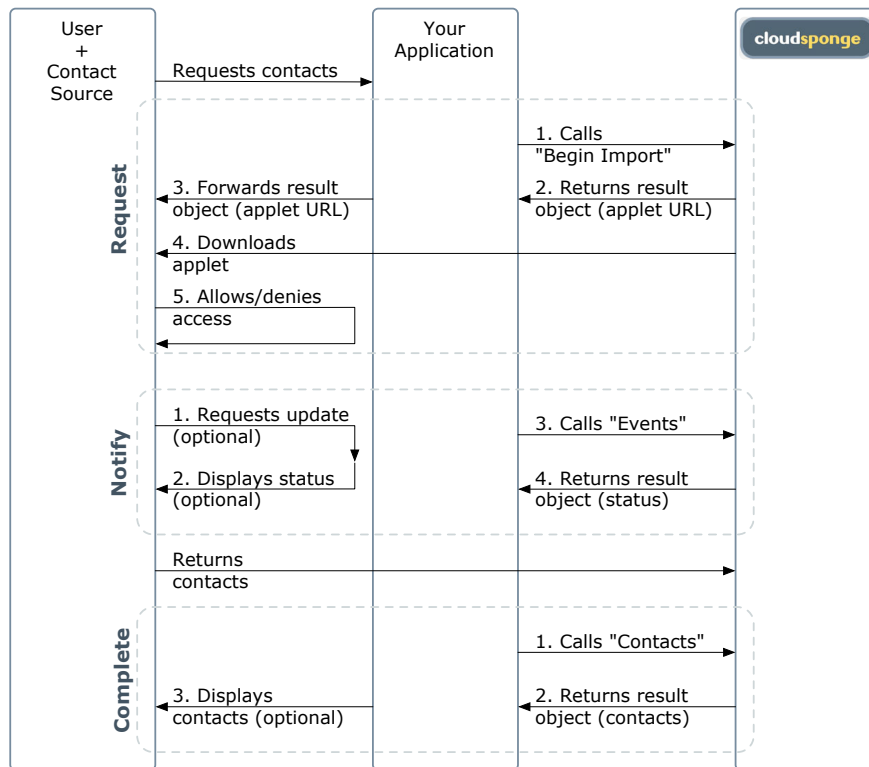


Figure 1.2.3: Desktop Applet

Request

1. Your application calls the "Begin Import" URL, including the contact source requested.
2. CloudSponge.com returns a result object, including a URL for the desktop applet, which needs to be downloaded by the user and an import_id to be used to identify the import to the applet and to fetch the events and contacts.
3. The user's system is directed to download the applet from <https://api.cloudsponge.com> (refer to section 2.2.1 Initiating an Import for details on how the applet tag is created).
4. User's browser downloads the applet to their desktop.
5. User chooses to allow/deny the applet access to their contacts.

Notify

1. (Optional) User requests a status update on the contact import.
2. (Optional) Applet displays a status update to the user.
3. Your application calls the "Events" URL to get the current Import Events, specifying the import_id returned in the initial request.
4. CloudSponge.com returns the current Event result object (refer to section 2.2 Events and Contacts for event status codes).

Complete

1. Once a successful "Complete" event has been received from "Notify" above, your application calls the "Contacts" URL to retrieve the contacts, specifying the import_id returned in the initial request.
2. CloudSponge.com returns the Contacts result object in the format requested.

3. (Optional) Your application displays the contacts to the user.

1.3. API Authentication

Access to the Address Book API is based on the domain registered at CloudSponge.com. Your domain key combined with your domain password authenticates your call to the API. These authentication parameters can be passed via HTTP basic authentication or passed as query parameters in the request.

When using HTTP basic authentication, specify the domain key as they username and the domain password as the password in the form:

```
<domain_key>:<domain_password>
```

When authenticating with query parameters, include the domain key and domain password as parameters as shown below:

```
?domain_key=<domain_key>&domain_password=<domain_password>
```

1.4. Retrieving Contacts

Once the import is complete and your application receives notice the contacts have been imported successfully, you can pull the contacts in one of two ways:

1. **Address Book API:** When retrieving contacts from the API, use the Fetch Contacts function to retrieve the contacts. Contacts are available in either JSON or XML format. Both formats are analogous and use the same names to identify contact details. Examples of each are included in the Appendix.
2. **Direct Database:** To retrieve contacts via the direct database connection, you must setup a connection to the database and setup decryption of the data contained in the Contacts table.

2. Integration Checklist

- ✓ Configure your CloudSponge.com account
 - ✓ Set up domains
 - ✓ Record domain keys
 - ✓ Record domain passwords
- ✓ Handle Events (optional)
 - ✓ On server: relay events to the browser
 - ✓ In browser: display events to the user
- ✓ Watch for completion event
- ✓ Fetch contacts from CloudSponge.com
- ✓ Complete CloudSponge.com registration and turn off Testing mode

2.1. Getting Ready

You will need to setup your CloudSponge.com account to properly authenticate with CloudSponge.com. All these actions can be configured from the My Account tab in your CloudSponge.com account.

To register the domain using the web interface:

1. Under Technical Info, add or edit a domain for your development environment.
2. Enter the Domain Name for the development environment of your application.
 - Domain: root domain and port (if not using port 80)
 - Database IP address: address from which you will retrieve contacts from the database (if using the database to retrieve contacts). This is typically your application's IP address.
 - Test Mode: The first domain you create in your account is automatically set in test mode. The domain will be taken out of test mode when the trial period expires.
3. Click "Create"
4. On the Domain page, make note of the additional parameters that are automatically created for your account:
 - Domain Key: key for use to access the API.
 - Domain Password: password used to access the API.
 - Database settings: settings used to connect to the contacts database directly to retrieve contacts.
 - Encryption Parameters: the key and initial vector values used to decrypt data in the contacts database.

2.2. Server-side Dependencies

2.2.1. Initiating an Import

An import always begins when your application calls the Begin Import URL.

User Consent	https://api.cloudsponge.com/begin_import/user_consent
Import	https://api.cloudsponge.com/begin_import/import
Desktop Applet	https://api.cloudsponge.com/begin_import/desktop_applet

In general the API returns a result object that indicates the result of the call, the identifier for subsequent calls to retrieve events and contacts, and may contain a URL to direct the user's browser to, in order to proceed with the import. The User Consent and Desktop Applet imports both return a URL value that should be handled differently, depending on the case. The details of this are described below in the Client-side Dependencies section.

2.2.2. Formatting the Response for User's Browser

User Consent

The value of the URL returned from the User Consent import is the specific URL where the user's browser should be directed to initiate the import. This URL should be loaded in a popup window, so that once the user has provided consent, the window will close.

Import

There are no additional values returned for this type of import.

Desktop Applet

The URL value returned from the Desktop Applet specifies the location of the current Java applet that the user's browser must load to facilitate the contact import. This applet should be loaded in the browser using HTML like shown below:

```
<APPLET archive="[applet_url]" code="ContactsApplet" id="Contact_Importer"
width="0" height="0">
  <PARAM name="domainKey" value="[domain_key]"/>
  <PARAM name="cookieValue" value="document.cookie"/>
  <PARAM name="importId" value="[import_id]"/>
  Your browser does not support Java which is required for this utility to
  operate correctly.
</APPLET>
```

2.2.3. Polling Events

After a contact import is initiated, your application must monitor the events as they are generated from CloudSponge.com. Use the id parameter that was returned from the call to `begin_import` to identify the import request to CloudSponge.com when fetching events. Use the following authenticated API to fetch events:

https://api.cloudsponge.com/events/<import_id>

This API only returns new events that have not been returned already. The following table summarizes the events your application should be aware of. Your application should render the result of the event calls in the user's browser if you wish to keep your user informed about the progress of the import.

Event Category	event_type	Status	Value	description
Initializing	INITIALIZING	INPROGRESS		
Initializing	INITIALIZING	COMPLETED		
Initializing	INITIALIZING	ERROR	Integer error code	Human readable error description
Gathering	GATHERING	INPROGRESS	Number of contacts currently gathered	
Gathering	GATHERING	COMPLETED		
Gathering	GATHERING	ERROR	Integer error code	Human readable error description
Import Complete	COMPLETE	COMPLETED		
Import Complete	COMPLETE	ERROR	Integer error code	Human readable error description

Various error conditions are possible and are summarized below:

Error Code	Error Condition	Error Description
0x0001	Failed domain authentication when accessing begin_import.	Could not authenticate the Domain key/password.
0x0002	Invalid parameters supplied to begin_import.	Invalid parameters supplied.
0x0100	Unexpected error occurred during webmail import.	An unexpected error occurred while importing your contacts.
0x0101	Webmail import failed.	Could not import from <service>: <reason>.
0x0200	Unknown error occurred during a delegated import.	An unexpected error occurred while importing your contacts.
0x0201	Delegated import failed because the domain is not permitted to use the service.	The domain you are attempting access this service from is not permitted to use the service.
0x0202	Delegated import failed because the user did not provide consent to access their contacts.	Consent was not given to access your contacts.
0x0300	Unexpected error occurred during applet import.	An unexpected error occurred while importing your contacts.

0x0301	Applet failed to import because user did not trust the applet.	You must trust the applet to import your contacts.
0x0302	Applet failed to import because it could not find an appropriate address book to import.	No supported contact source was detected on your system.
0x0303	Applet failed to submit contacts to CloudSponge.com.	Failed posting contacts to the server.

2.2.4. Retrieving Contacts

When the Import Complete event is received with the COMPLETED status indicating a successful import, your application should retrieve the contacts from the API:

https://api.cloudsponge.com/contacts/<import_id>

This API returns the retrieved contacts. Alternatively, your application can retrieve the contacts from the direct database connection. The contacts are available from CloudSponge.com for up to 2 hours. After this point, they will be deleted from CloudSponge.com. See the Appendix for details on accessing the API with examples.

2.3. Release into Production

Finally, you are confident and ready to release this new functionality into your production environment. Your CloudSponge.com account must be updated so that the production environment is recognized by CloudSponge.com and authenticates properly.

Appendix A: HTTP Response Codes

When accessing the Address Book API, always check the HTTP response before attempting to process the response body. There are several typical responses that should be checked for.

Code	Description
200	the request processed successfully, process the body of the response for details.
400	Invalid parameters were supplied
401	Access was denied to the requested resource. Either the supplied domain_key and domain_password did not match a domain or the request attempted to access a resource that you don't have access to.
404	(fetching contacts only) the contacts are not yet available
410	(fetching contacts only) the contacts have already been retrieved and have been deleted from CloudSponge.com

Appendix B: Address Book API Reference

Begin Import: User Consent

URL: https://api.cloudsponge.com/begin_import/user_consent<.format>

Method: POST

Parameters:

- service: contact source name
 - YAHOO
 - WINDOWS LIVE
 - GMAIL
- domain_key: your account key
- domain_password: your domain password
- user_id (optional): any unique identifier you use to identify a user
- echo (optional): any customer defined string data to be returned in the response
- format: (optional) either '.json' (default) or '.xml'

Example:

https://api.cloudsponge.com/begin_import/user_consent.json?service=YAHOO&user_id=myUserId_0003&echo=echo123&domain_key=33664218758c5244136965160db455db012b1411&domain_password=Tpa01z+vVPE7MxXi

Response:

- status: success or failure
- url: User Consent URL where the user's browser should be directed
- import_id: the identifier for this import, used in subsequent calls to fetch events and contacts
- user_id: the customer defined string that was passed in as a parameter
- echo: the customer defined string that was passed in as a parameter

Example:

JSON:

```
{
  "status": "success",
  "url": "https://api.login.yahoo.com/oauth/v2/request_auth?oauth_token=d",
  "import_id": 1126,
  "user_id": "myUserId_0003",
  "echo": "echo123"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <status>success</status>
  <url>https://api.login.yahoo.com/oauth/v2/request\_auth?oauth\_token=d</url>
  <import-id>1126</import-id>
  <user-id>myUserId_0003</user-id>
  <echo>echo123</echo>
</result>
```

Begin Import: Import

URL: https://api.cloudsponge.com/begin_import/import<format>

Method: POST

Parameters:

- username: username for the contact source
- password: password for the contact source
- service: contact source name
 - GMAIL
 - AOL
 - PLAXO
- domain_key: your account key
- domain_password: your domain password
- user_id (optional): any unique identifier you use to identify a user
- echo (optional): any customer defined string data to be returned in the response
- format: (optional) either '.json' (default) or '.xml'

Example:

https://api.cloudsponge.com/begin_import/import.xml?username=abc123&password=123456&service=AOL&user_id=myUserId_0001&echo=echo123&domain_key=33664218758c5244136965160db455db012b1411&domain_password=Tpa01z+vVPE7MxXi

Response:

- status: success or failure
- import_id: the identifier for this import, used in subsequent calls to fetch events and contacts
- user_id: the customer defined string that was passed in as a parameter
- echo: the customer defined string that was passed in as a parameter

Example:

JSON:

```
{
  "status": "success",
  "import_id": 1131,
  "user_id": "myUserId_0003",
  "echo": "echo123"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <status>success</status>
  <import-id>1130</import-id>
  <user-id>myUserId_0003</user-id>
  <echo>echo123</echo>
</result>
```


Begin Import: Desktop Applet

URL: https://api.cloudsponge.com/begin_import/desktop_applet<format>

Method: POST

Parameters:

- service: contact source name
 - OUTLOOK
 - ADDRESSBOOK
- domain_key: your account key
- domain_password: your domain password
- user_id (optional): any unique identifier you use to identify a user
- echo (optional): any customer defined string data to be returned in the response
- format: (optional) either '.json' (default) or '.xml'

Example:

https://api.cloudsponge.com/begin_import/desktop_applet.json?service=ADDRESSBOOK&user_id=myUserId_0003&echo=echo123&domain_key=33664218758c5244136965160db455db012b1411&domain_password=Tpa01z+vVPE7MxXi

Response:

- status: success or failure
- url: Java applet URL to be used to process the import
- import_id: the identifier for this import, used in subsequent calls to fetch events and contacts
- user_id: the customer defined string that was passed in as a parameter
- echo: the customer defined string that was passed in as a parameter

Example:

JSON:

```
{
  "status": "success",
  "url": "https://api.cloudsponge.com/objects/ContactsApplet_signed.jar",
  "import_id": 1132,
  "user_id": "myUserId_0003",
  "echo": "echo123"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <status>success</status>
  <url>https://api.cloudsponge.com/objects/ContactsApplet\_signed.jar</url>
  <import-id>1132</import-id>
  <user-id>myUserId_0003</user-id>
  <echo>echo123</echo>
</result>
```

Events

URL: https://api.cloudsponge.com/events<format>/<import_id>

Method: GET

Parameters:

- import_id: ID returned from the call to begin_import
- domain_key: your account key
- domain_password: your domain password
- echo: any customer defined string data to be returned in the response
- format: (optional) either '.json' (default) or '.xml'

Example:

```
https://api.cloudsponge.com/events.json/1126/?domain_key=33664218758c5244136965160db455db012b1411&echo=1234&domain_password=Tpa01z+vVPE7MxXi
```

Response:

- an Event object describing the current event
- user_id: the customer defined string that was passed in when initiating the import
- echo: the customer string that was passed in
- import_id: the identifier for this import

Example:

JSON:

```
{
  "events":
  [
    {"value":0,"status":"COMPLETED","event_type":"INITIALIZING"},
    {"value":2,"status":"COMPLETED","event_type":"GATHERING"},
    {"value":0,"status":"COMPLETED","event_type":"COMPLETE"}
  ],
  "echo":null,
  "user_id":"myUserId_0003",
  "import_id":1126
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<eventsResponse>
  <events type="array">
    <event>
      <created-at type="datetime">2009-09-11T09:08:02-07:00</created-at>
      <description nil="true"></description>
      <event-type>INITIALIZING</event-type>
      <id type="integer">1834</id>
      <import-action-id type="integer">796</import-action-id>
      <status>COMPLETED</status>
      <unclaimed type="boolean">>false</unclaimed>
      <updated-at type="datetime">2009-09-11T09:08:33-07:00</updated-at>
      <value type="integer">0</value>
    </event>
    <event>
      <created-at type="datetime">2009-09-11T09:08:02-07:00</created-at>
      <description nil="true"></description>
      <event-type>GATHERING</event-type>
```

```
<id type="integer">1835</id>
<import-action-id type="integer">796</import-action-id>
<status>COMPLETED</status>
<unclaimed type="boolean">>false</unclaimed>
<updated-at type="datetime">2009-09-11T09:08:33-07:00</updated-at>
<value type="integer">2</value>
</event>
<event>
  <created-at type="datetime">2009-09-11T09:08:11-07:00</created-at>
  <description nil="true"></description>
  <event-type>COMPLETE</event-type>
  <id type="integer">1836</id>
  <import-action-id type="integer">796</import-action-id>
  <status>COMPLETED</status>
  <unclaimed type="boolean">>false</unclaimed>
  <updated-at type="datetime">2009-09-11T09:08:33-07:00</updated-at>
  <value type="integer">0</value>
</event>
</events>
<echo nil="true"></echo>
<user-id>myUserId_0003</user-id>
<import-id>1126</import-id>
</eventsResponse>
```

Contacts

URL: https://api.cloudsponge.com/contacts<format>/<import_id>

Method: GET

Parameters:

- import_id: the ID returned from the call to begin_import
- domain_key: your account key
- domain_password: your domain password
- echo: any customer defined string data to be returned in the response
- format: (optional) either '.json' (default) or '.xml'

Example:

https://api.cloudsponge.com/contacts.json/1126/?domain_key=33664218758c5244136965160db455db012b1411&echo=1234&domain_password=Tpa01z+vVPE7MxXi

Response:

- A result structure indicating the result of the operation
- A list of contacts retrieved from the source, or an error if the contacts are not available
- echo: the customer string that was passed in
- user_id: the value passed in to the call the begin_import for this import
- import_id: the identifier for this import
- contacts_owner: contact information for the owner of the contacts

Example:

JSON Format

```
{
  "echo": null,
  "user_id": "myUserId_0003",
  "import_id": 1126,
  "contacts_owner": {
    "first_name": "Joe",
    "last_name": "Smith",
    "email": [ { "address": "joe@example.com" } ]
  },
  "contacts": [
    {
      "first_name": "John",
      "last_name": "Doe",
      "phone": [
        { "number": "555-1234", "type": "Home" },
        { "number": "555-2468", "type": "Work" }
      ],
      "email": [
        { "address": "johndoe@nowhere.com", "type": "Email 1" },
        { "address": "second@email.com", "type": "Email 2" }
      ]
    },
    {
      "first_name": "Jane",
      "last_name": "Smith",
      "phone": [ { "number": "555.5678", "type": "Home" } ],
      "email": [ { "address": "janesmith@nowhere.com", "type": "Email 1" } ]
    }
  ]
}
```

```
    ]
}
```

XML Format

```
<?xml version="1.0" encoding="UTF-8"?>
<contactsResponse>
  <echo nil="true"></echo>
  <user-id>myUserId_0003</user-id>
  <import-id>1126</import-id>
  <contacts-owner>
    <first-name>Joe</first-name>
    <last-name>Smith</last-name>
    <email type="array">
      <email>
        <address>joe@example.com</address>
      </email>
    </email>
  </contacts-owner>
  <contacts type="array">
    <contact>
      <first-name>John</first-name>
      <last-name>Doe</last-name>
      <phone type="array">
        <phone>
          <number>555-1234</number>
          <type>Home</type>
        </phone>
        <phone>
          <number>555-2468</number>
          <type>Work</type>
        </phone>
      </phone>
      <email type="array">
        <email>
          <address>johndoe@nowhere.com</address>
          <type>Email 1</type>
        </email>
        <email>
          <address>second@email.com</address>
          <type>Email 2</type>
        </email>
      </email>
    </contact>
    <contact>
      <first-name>Jane</first-name>
      <last-name>Smith</last-name>
      <phone type="array">
        <phone>
          <number>555.5678</number>
          <type>Home</type>
        </phone>
      </phone>
      <email type="array">
        <email>
          <address>janesmith@nowhere.com</address>
          <type>Email 1</type>
        </email>
      </email>
    </contact>
  </contacts>
</contactsResponse>
```

Appendix C: Direct Database Connection

Configure a connection to the CloudSponge.com MySQL database with the following parameters:

- Host: cloudsponge.com
- Port: 3306
- Database: database name from your domain settings
- Username: database username from your domain settings
- Password: database password from your domain settings

The database contains a single table with all your available contacts in it. These contacts can be retrieved by the `import_id` value that was returned when initiating the import. Below is a summary of the table structure:

- `import_id` – the id value returned when initiating the import
- `user_id` – the value you specified with initiating the import
- `first_name` – (encrypted) the first name of the contact
- `last_name` – (encrypted) the last name of the contact
- `emails` – (encrypted) a semi-colon separated list of email addresses for the contact
- `phones` – (encrypted) a semi-colon separated list of phone numbers for the contact

Once the database connection has been established, data fields can be decrypted using an AES 256 CBC cipher configured with the key and initial vector for the domain. These values can be found in base-64 format on the details page for the domain. Be sure to decode the values into a binary string when setting up the cipher.