cloud elements

Product Dictionary

The Cloud Elements Product Dictionary

A simple vocabulary lesson is often required to get up and running as quickly as possible. Refer to this guide to get to know the Cloud Elements API Integration and Management platform. Our aim is to help developers publish, integrate, aggregate and manage their APIs through a unified platform. Beyond the Cloud Elements catalog of integrations, we offer an integration toolkit that enables developers to connect entire categories of cloud services (e.g. CRM, Documents, Finance) or synchronize data between multiple cloud services (e.g. Salesforce, Zendesk and Quickbooks).

Find more information and give our integrations a try at www.cloud-elements.com

The Cloud Elements Vocabulary

- API Hub
- Element
- Element Instance
- Authentication
- Resources
- Discovery Service
- Payloads
- Pagination
- Search
- Errors
- Events
- Documentation
- Bulk
- Cloning & Modifying
- Automated Test Scripts

API Hub

An API Hub ("Hub") provides uniform APIs to access a collection of resources enabling "One-to-Many" API access to multiple application services. For example, the CRM hub provides normalized access to multiple CRM application services including Salesforce, Microsoft Dynamics, Netsuite CRM, Sugar CRM, Zoho CRM, and Autotask (we call a connection to each service an "Element"). Your application can write to the "One" uniform API and get access to "Many" application services in that category. Within the CRM Hub, Cloud Elements has normalized the API calls to multiple resources such as Accounts, Contacts, Leads, Opportunities and more. Hubs represent the intersection of resources across the Elements within that category of application services.

Element

An 'Element' is a pre-built API integration that enables a connection into a specific cloud application endpoint (e.g., Salesforce, Quickbooks, or Marketo). All Elements start with a normalized set of features, including authentication, resources, paging, errors, events and search. At the Element level, we also seek to support the richer set of APIs that an application provides, even when not all of the Elements in that category share that resource. For example, Salesforce Sales Cloud has APIs that many other CRM services do not support. You can find these APIs that are specific to just Salesforce in the documentation for that Element.

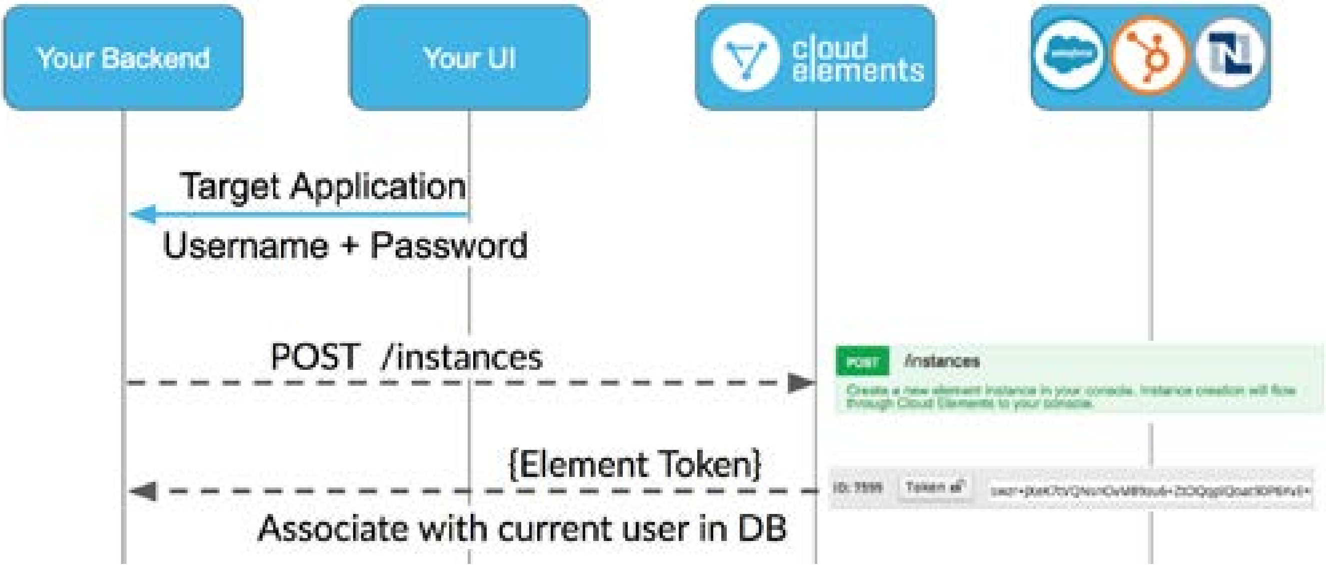
Element Instance

An "Element Instance" is an Element that is authenticated to a specific user account for the application service. An instance can access all of the objects, fields and data for that account - including custom data. An instance is created when a user successfully connects to the endpoint by providing an instance name, the required authentication credentials for that Element, and optionally add configuration for events. An Element Instance represents a connection to a single authenticated account at the target endpoint such as a Salesforce, Marketo, or Netsuite.

Authentication

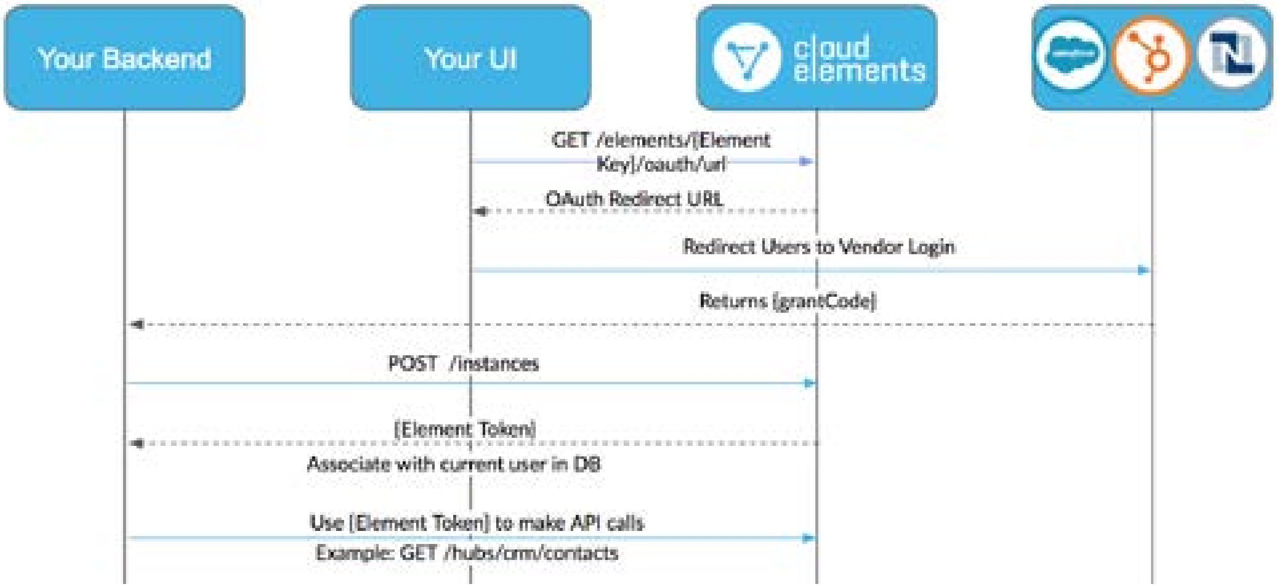
Cloud Elements has normalized authentication for each type of the authentication used by the endpoint (e.g., Basic, OAuth1, OAuth2, WS-Security, API Key, etc). There are inherent authentication workflow variations depending on the type of authentication (e.g., basic credentials vs. OAuth2) but the authentication types are implemented consistently. Cloud Elements uses a normalized API token based approach with a user and organization secret, combined with an Element token associated with the specific authenticated account of the Element. Switching endpoints within a Hub is as simple as swapping out a single element token.

Basic Authentication Handshake



Create a new element instance in your console. Instance creation will flow through Cloud Elements to your console.

OAuth Handshake


Your Backend
Your UI
cloud elements
GET /elements/{Element Key}/oauth/url
OAuth Redirect URL
Redirect Users to Vendor Login
Returns {grantCode}
POST /instances
{Element Token}
Associate with current user in DB
Use {Element Token} to make API calls
Example: GET /hubs/crm/contacts

Resources

Every resource is accessed using a consistent RESTful API with a JSON payload regardless of the protocol used at the endpoint. The Cloud Elements pre-built connectors do the work of mapping the normalized API call to each application's API endpoints, as well as transforming SOAP, XML and other API protocols to REST/JSON.

When Cloud Elements creates an Element, the resources defined for that Element are normalized to the intersection of resources that are available across the applications in that category. For example, a user can create a common object for use across hubs, which could be a Company Object. In the individual application endpoints, they may each use different naming for this resource, such as Account in Salesforce, Company in HubSpot and Organization in Quickbooks.

Cloud Elements resource normalization transforms all varieties of the Company Object to one uniform model. Additional resource normalization can also be achieved by the end user using our Transformation services. With Transformations existing resources can be renamed, normalized and "made to look" the same as other resources within a given hub (see transformation information and videos). When an endpoint doesn't support a resource that's available at the Hub level; Cloud Element provides an "API not supported" error when that resource is called.

Discovery Service

Cloud Elements includes a comprehensive data discovery service that provides normalized metadata, such as the list of field names and types. Additional information, if available from an endpoint, may also be obtained such as: display name, read-only, etc. If an endpoint doesn't provide discovery service APIs, Cloud Elements will still provide a minimum set of metadata about the given resource (e.g., name and type). Cloud Elements also allows you to discover custom fields (as long as the values are not empty), by supplying an object Id when a native discovery service is not available. The Discovery Service is used along with the Transformation Service to normalize the responses across endpoints.

Payloads

By default Cloud Elements returns the entire payload from the vendor, so you have access to the entire response data. However, with our object and data transformation service, you can define your own normalization of the payload using API calls or via a simple drag and drop user interface. Through this approach, payloads can be normalized based on your definition of a common object model.

Pagination

When a request returns multiple pages of a response, Cloud Elements normalizes the pagination information from the endpoint, allowing the developer to implement normalized pagination code. Cloud Elements' pagination accepts an input parameter of "page" (defaults to 1) and "page size" and returns a "next page" token that links to the next page of results. If there are no more results to return, an empty result set will be returned (i.e., no data left to return). Typically the number of records per page is dictated by the endpoint. The pageSize parameter provided to Cloud Elements by an application can never be larger than that allowed by the endpoint. As a best Practice, send a page size of a value < 20 to ensure that it works consistently across the Hub.

Search

We provide a standard SQL based query language (Cloud Elements' Query Language, CEQL) that is consistent across each Element. Using CEQL, developers can pass the same query structure to all element endpoints. An example is the "where" query parameter of lastName='Jones'. Cloud Elements does the work of transforming the CEQL to the native search language at an endpoint. It is important to note that search capabilities vary significantly across endpoints so searches may be limited if the endpoint does not allow searching of certain fields. However, our goal is to provide CEQL support for all of the fields that are searchable at the endpoint and are aligned with the CEQL structure. Nested subqueries and other complex queries are not supported at this point as it can not be normalized readily across endpoints.

Errors

Errors are returned in a normalized JSON structure providing consistent error codes across all endpoints. In addition, Cloud Elements also provide the complete endpoint specific error message in order to assist developers in debugging issues. Check out our list of standardized error codes: https://developers.cloud-elements.com/docs/overview/error-codes.html

Events

Cloud Elements can be configured to listen for and notify users of actions/changes (i.e., Creates, Updates) to endpoint resources in a normalized way. We leverage webhook events when available from an endpoint, or if webhooks are not supported we provided a polling service that polls for new or modified data at the endpoint. Whether we use webhooks or polling we provide a consistent JSON payload of the action that occurred to a resource along with the object name and the object id. For completeness we also include the vendor's raw payload of the data associated with that event. Note that in order to provide a polling event capability, an endpoint must provide the ability to search for resource updates. Polling also has an inherent limitation that cannot support deleted events.

Documentation

Cloud Elements provides a uniform set of interactive API documentation that developers use to view or access the Hub and Element resources. You can access the interactive API documentation by either logging in to the UI where you can invoke our RESTful APIs by providing the required parameters; or in the Developer Portal (no sign-up required) which provides documentation about each Hub and Element API along with other materials to get started. Element specific or other unique aspects are documented for each Element, such as application setup, authentication, how to transform data, and the OpenApi Specification (formerly Swagger 2.0) for the supported resources and methods.

Bulk

Cloud Elements provides the ability to upload and download data in bulk from an endpoint in a normalized way. Cloud Elements will leverage the provider bulk endpoints whenever that is available. When there is no bulk available from the provider, Cloud Elements will provide a pseudo bulk service for uploading and downloading data from the endpoint. For uploads, we will accept a file and then create objects at the endpoint on a record by record basis. For downloads, we will execute a search API against the endpoint, and loop through all results until we have retrieved all the data. Cloud Elements will store these files (encrypted) in our platform for a maximum of three days. We're currently rolling out a Bulk service on an Element by Element basis. Let us know if you need Bulk APIs for a given Element by contacting us at support@cloud-elements.com.

Cloning & Modifying an Element

Cloud Elements allows the user to clone (copy) an existing Element instance. This allows the user to modify the cloned Element without affecting the original. Note that not all Elements are cloneable at this time but we are migrating Elements to this new framework. If there is a "gear" in the upper right of the API Manager Console for that instance, it is cloneable. Examples of enhancing existing elements can be found in the Developer Portal.

Automated Test Scripts

Normalized Element-specific automated tests are created for each endpoint API and sub-resource API. Test scripts are run on a regular basis as part of the Cloud Elements continuous integration process.