

General API Functions

Epic	https://goevie.atlassian.net/browse/GS-254	
Document status	DRAFT	
Document owner	@ Michael Naughton	
QA	@ Michael Naughton	@ Chris Neave

- 1. Technical Overview
 - 1.1 Networking
 - 1.2 Software Architecture
 - 1.3 Instantiate a Query
 - 1.3.1 Creating a new derived class
 - 1.3.2 Using the derived class
 - 1.4 Functions available
 - 1.4.1 createObject()
 - 1.4.2 deleteObject()
 - 1.4.3 lookupObject()
 - 1.4.4 updateObject()
 - 1.4.5 getObjectProperties()
 - 1.4.6 getAllObjectProperties()
 - 1.4.7 bulkUpdate()
 - 1.5 Connection flow

1. Technical Overview

1.1 Networking

Endpoint	Function	URI
/jobs/heartbeatToSitetracker	Batch inserts one or many charger heartbeats into sitetracker	https://localhost:3000/jobs/heartbeatToSitetracker
/jobs/connectorStatusToSitetracker	Batch inserts one or many connector statuses into Sitetracker	https://localhost:3000/jobs/connectorStatusToSitetracker

1.2 Software Architecture

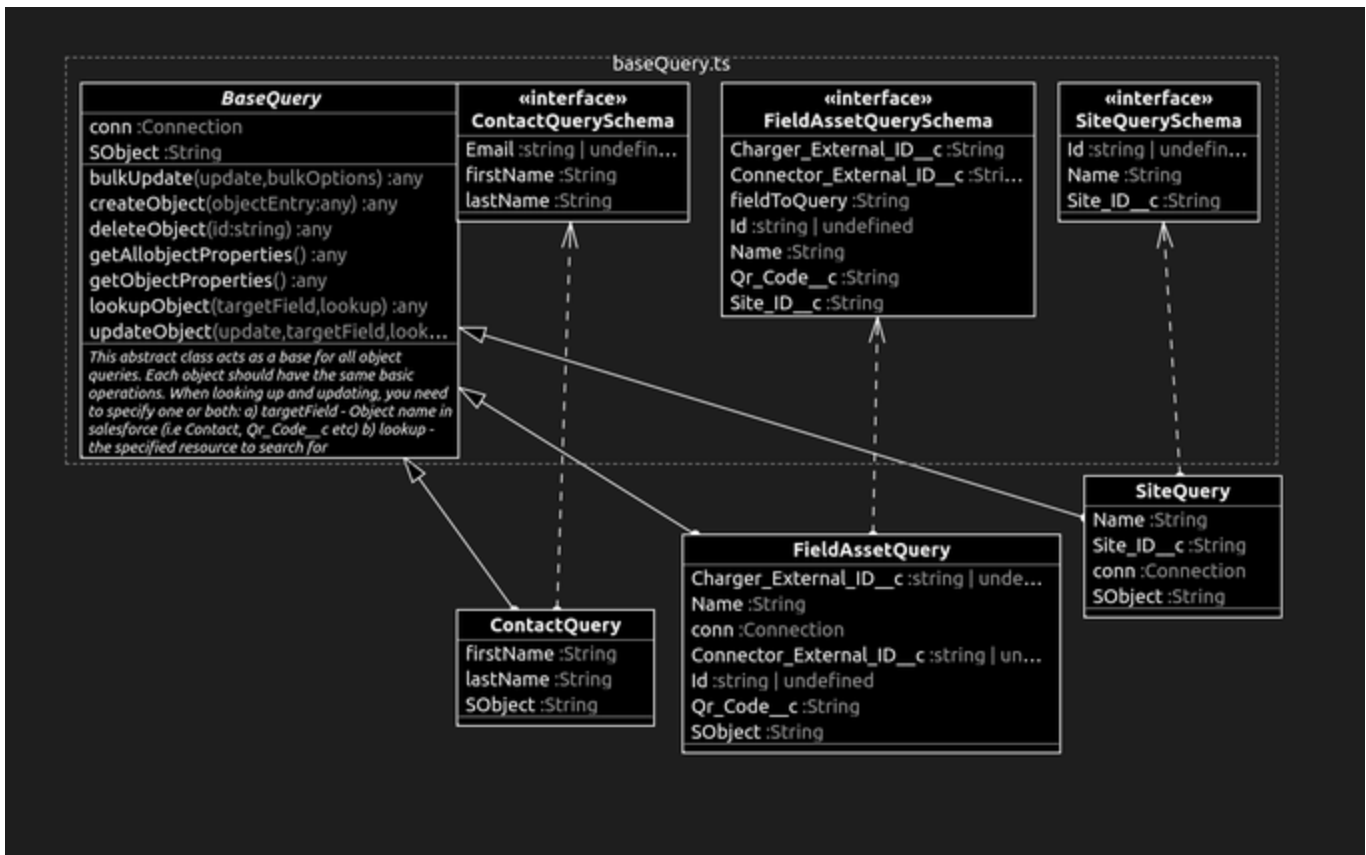
By implementing an abstract class to act as a base for most polymorphic functions - I was able to significantly reduce repeated code.

I declare a number of functions in `baseQuery.ts`, and they all are usable by its derived classes.

A derived class must be instantiated for every Salesforce object (sObject) type that you wish to query.

Currently, only 3 sObjects are implemented in this service:

1. Contact
2. Field Asset
3. Site



1.3 Instantiate a Query

1.3.1 Creating a new derived class

Below is the creation of a derived class for the "Field Asset" sObject.

Note that you must explicitly assign the sObject identified in the constructor, like in line 26.

```

/**
 * This derived class inherits methods from baseQuery for the sObject
 * sitetracker__Field_Asset__c, or "Field Asset"
 *
 * @export
 * @class SiteQuery
 * @extends {BaseQuery}
 */
export class FieldAssetQuery extends BaseQuery {
  constructor(conn: Connection) {
    super("sitetracker__Field_Asset__c", conn);
  }
}

```

<https://github.com/goevie/salesforce-integration/blob/main/src/query/fieldAssetQuery.ts>


1.3.2 Using the derived class

When Instantiating a Class, you must pass the fields that are designated in the derived classes constructor, in this case an object of type `fieldAssetQuerySchema` (*not used - TODO remove*) and a salesforce connection.

```
/**
 * Instantiate a derived class
 *
 * @export
 * @param {any[]} statuses
 * @param {Connection} conn
 * @return {*}
 */
export async function bulkConnectorStatusToSitetracker(
  statuses: any[],
  conn: Connection
) {
  let fieldAsset = new FieldAssetQuery(conn);

  const bulkOptions: BulkOptions = { extIdField:
    "Connector_External_ID__c" };

  return await fieldAsset.bulkUpdate(statuses, bulkOptions);
}
```

 Remove `fieldAssetQuerySchema` - we only need the `this.sObject` and `this.conn` for bulk update

1.4 Functions available

File where most of the API logic lives <https://github.com/goevie/salesforce-integration/blob/main/src/query/baseQuery.ts>

1.4.1 createObject()

```
/**
 * Creates a single object of type this.sObject
 *
 * @param {Object} objectEntry -
 * @return {*} {(Promise<SuccessResult | ErrorResult>)}
 * @memberof BaseQuery
 */
```

1.4.2 deleteObject()

```

/**
 * Deletes a single object of type this.sObject
 *
 * @param {string} id
 * @return {*} {Promise<RecordResult>}
 * @memberof BaseQuery
 */

```

1.4.3 lookupObject()

```

/**
 * Query to find all objects of type this.sObject that matches the
query object.
 * Note that the query fields need to match the sitetracker API name
of the object.
 *
 * @param {string} targetField
 * @param {string} lookup
 * @return {*} {Promise<
 *   {
 *     Id?: string;
 *     attributes?: RecordAttributes;
 *   }[]
 * >}
 * @memberof BaseQuery
 */

```

1.4.4 updateObject()

```

/**
 * Updates a single value within a single object
 *
 * @param {Object} update
 * @param {string} targetField
 * @param {string} lookup - sitetracker Id of object
 * @return {*} {Promise<RecordResult[]>}
 * @memberof BaseQuery
 */

```

1.4.5 getObjectProperties()

```
/**
 * Gets all SObjects currently in Salesforce
 *
 * @return {*} {Promise<DescribeGlobalResult>}
 * @memberof SiteQuery
 */
```

1.4.6 getAllObjectProperties

```
/**
 * Return an array of the API names of fields within this.sObject
 *
 * @return {*} {Promise<any[]>}
 * @memberof BaseQuery
 */
```

1.4.7 bulkUpdate()

```
/**
 * Updates in bulk objects of type this.sObject
 * Promisified to return a promise
 *
 * @param {any[]} update
 * @param {BulkOptions} bulkOptions
 * @return {*} {Promise<any>}
 * @memberof BaseQuery
 */
```

1.5 Connection flow

We used a jwt token flow with a singleton design pattern to limit connections to one at a time.

Read more about this at [JWT Token Flow Salesforce](#).