CLEARMDM

clearMDM – Settings API Instructions

Author: Mark Cane (mark@clearmdm.com)
Version: v1.2
Date: 18th October 2018

Introduction

This document provides technical instructions (by example) for the deployment of application configuration settings between Salesforce orgs via the clearMDM Settings API.

The Settings API is RESTful API that supports the retrieval (GET) of configuration settings from a source Salesforce org as JSON data that can be deployed (POST) to one or more target Salesforce org.

Please Note, the Settings API is available in clearMDM product version 3.7 and above.

CLEARMDM – SETTINGS API INSTRUCTIONS 1
INTRODUCTION 1
DOCUMENT CHANGE CONTROL 2
API USE CASES 2
COMMIT MODEL 2
SETTINGS API COVERAGE 2
SETTINGS API PRE-REQUISITES 4
PERMISSIONED USER 4
CONNECTED APP 4
ACCESS TOKENS 6
SYSTEM SETTINGS 8
SYSTEM SETTINGS RETRIEVE (GET) 8
SYSTEM SETTINGS DEPLOY (POST) 10
TARGET OBJECT SETTINGS 11
TARGET OBJECT SETTINGS RETRIEVE (GET) 11
TARGET OBJECT SETTINGS DEPLOY (POST) 13
EXAMPLE API SCENARIO 14

CLEARMDM

Document Change Control

Version	Date	Author	Change
1.0	9th October 2018	Mark Cane	Initial version.
1.1	13th October 2018	Mark Cane	API Url corrections. Picture compression added to reduce the document file size.
1.2	18th October 2018	Mark Cane	MIME type confirmation Username => username correction Password => password correction

API Use Cases

The Settings API targets the following use cases.

- Build Automation. Incorporation of (clearMDM) application configuration settings into automated build processes (Continuous Integration, daily builds etc.).
- Configuration Management. Version management of (clearMDM) application configurations. For example, storage of versioned configurations in a source code repository alongside related Salesforce metadata.
- Ad-hoc Deployment. Convenient error-free deployment of configuration settings between Salesforce orgs.

Whilst the Settings API allows application configuration data to manually modified (via JSON editing) and re-deployed back to the same Salesforce org this practice is not supported.

Commit Model

Application configuration updates applied by the Settings API will either complete successfully or rollback automatically (on failure) to prevent a partial update state.

Where fields referenced (by API Name) in the Settings API JSON do not exist in the target org (or the running user has insufficient permissions) the related configuration setting will be set to blank. The API response errorText parameter will include the text “Missing Permissions” and the full list of affected fields will be recorded in the clearMDM Audit Log (accessible via the Audit Log tab in the clearMDM application).

Settings API Coverage

All application configuration settings are covered by the Settings API with the exception of workload tuning adjustments applied by clearMDM support. Please contact support@clearmdm.com to have such settings applied to nominated Salesforce orgs.

Settings API Pre-requisites

Permissioned User

The Settings API should be invoked via a Salesforce User with permissions to all fields referenced by the (clearMDM) application configuration (i.e. matching and merge rules etc.). The Salesforce User should also have the [MDM Data Steward] Permission Set assigned to ensure that all (clearMDM) packaged object and fields permissions are correctly set.

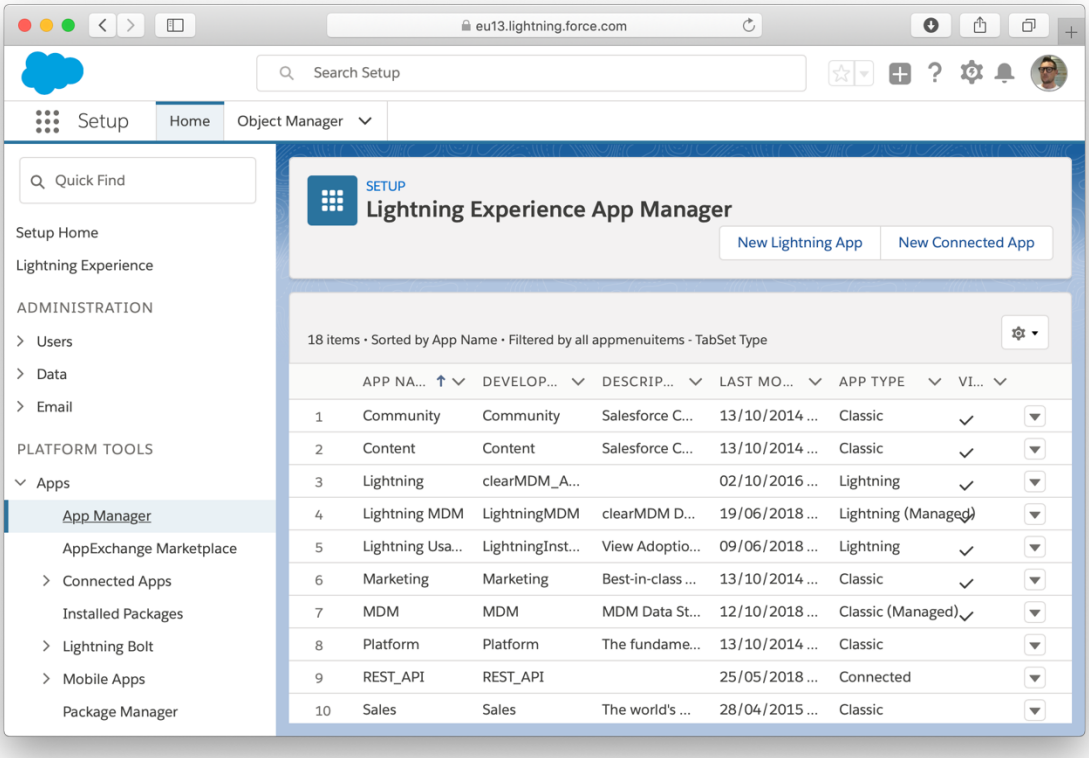
Connected App

The Settings API is a packaged API resource exposed via the standard Salesforce REST API. In order to authenticate to the Salesforce REST API (via the OAuth 2.0 protocol) a Connected App must exist in both the Source and Target Salesforce orgs. For further details in relation to Salesforce REST API authentication please refer to the link below.

https://developer.salesforce.com/docs/atlas.en-us.api_rest.meta/api_rest/intro_understanding_authentication.htm

An existing Connected App should be utilised where possible. Please follow the steps below to create a new Connected App. Note, the Salesforce help site provides additional details in respect to the creation of Connected Apps.

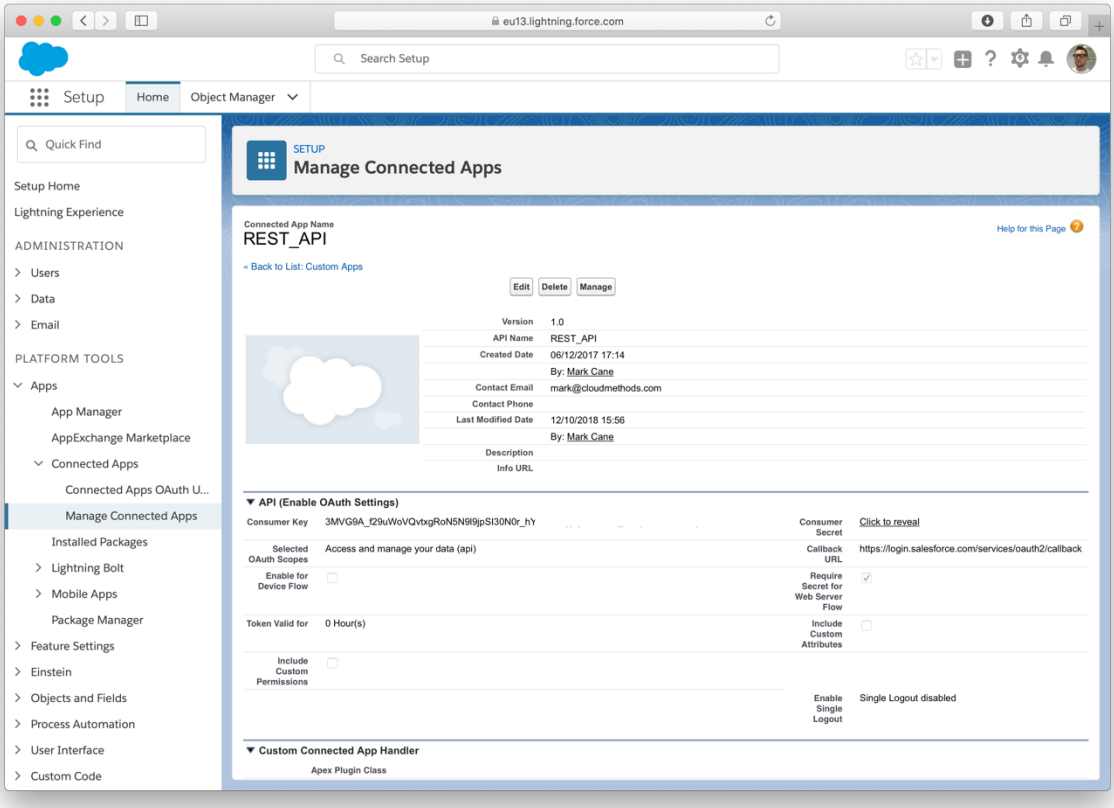
1. Open the Lightning App Manager and click the “New Connected App” button.



2. Enter the following information on the New Connected App page.

Connected App Field	Field Value
Connected App Name	REST API (*as preferred*)
API Name	REST_API (*as preferred*)
Contact Email	*as preferred*
Enable OAuth Settings	Checked
Callback URL	https://login.salesforce.com/services/oauth2/callback
Selected OAuth Scopes	Access and manage your data (api)

3. Save the New Connected App via the “Save” button. The detail page should appear as below.

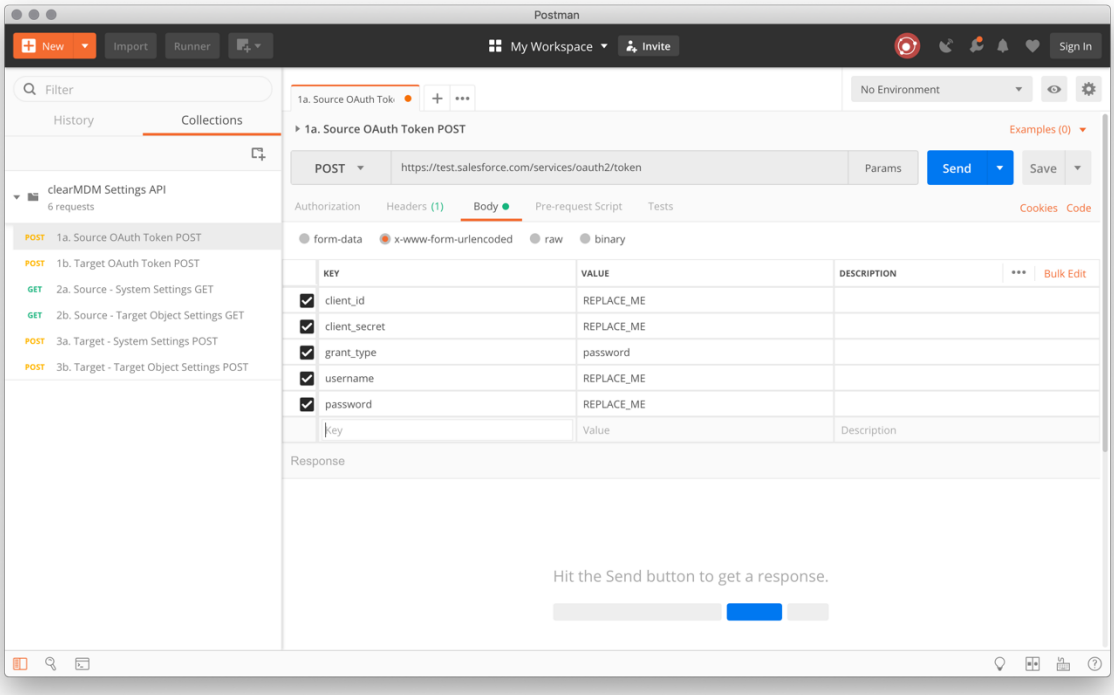


Access Tokens

Each Settings API interaction (POST or GET) is authenticated by an Access Token obtained via the OAuth 2.0 Username-password flow.

The following steps describe how an Access Token is retrieved using the Postman[1] utility application as the API client. Any REST API client can be utilised.

1. Open the Postman application and create a new Collection to group the Settings API requests within the open Workspace.



2. Create a new POST Request named OAuth Token POST.

3. Set the URL to:

   `https://test.salesforce.com/services/oauth2/token`

4. Set the Type to POST and Body format to **x-www-form-urlencoded** (as per the above screenshot).

[1] The Postman application is used for example purposes only; further details can be found at https://www.getpostman.com – all rights and trademarks respected.

5. Enter the Key Value pairs as defined below.

Key	Value
client_id	*Paste the Connected App Consumer Key*
client_secret	*Paste the Connected App Consumer Secret*
grant_type	password
username	*Salesforce user name*
password	*Salesforce user password + security token*

6. Click the Save button and then the Send button.

7. A response similar to the example below should be returned.


```
{
    "access_token": "00D5E0000000xzY!ARYAQPk.....",
    "instance_url": "https://acme--ClearMDM.cs81.my.salesforce.com",
    "id": "https://test.salesforce.com/id/00D../005..",
    "token_type": "Bearer",
    "issued_at": "1539358891688",
    "signature": "fn3KFec/v7E/1W9/..="
}
```


8. The *access_token* and *instance_url* values should be noted for use in the API interactions described in the following sections.

9. Note, the process above must be repeated for each Salesforce org accessed by the Settings API.

System Settings

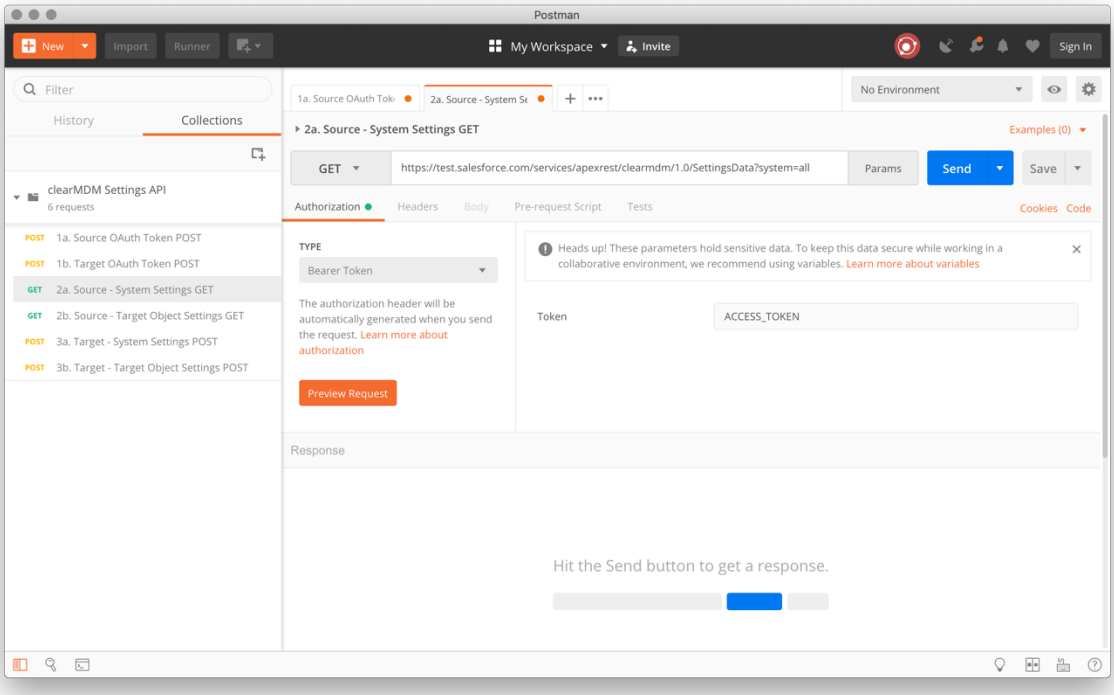
The Settings API supports 2 modes of settings retrieval; System Settings and Target Object settings. The former is covered in this section.

The System Settings mode covers settings displayed on the Application Settings tab of the clearMDM Settings page. Note, retrieving System Settings includes additional internal system settings that should not be modified without consultation with clearMDM support.

System Settings Retrieve (GET)

The following steps describe how System Settings are retrieved using the Postman utility application as the API client.

1. Create a new GET Request named System Settings GET.



2. Set the Type to GET and the URL to the *instance_url* response value retrieved in the Access Tokens section (point 8) with the path defined as below.

   `{instance_url}/services/apexrest/clearmdm/1.0/SettingsData`

3. Set the Authorisation Token to the *access_token* response value retrieved in the Access Tokens section (point 8).

4. Enter the Key Value pair as defined below. Or add a URL parameter as per the prior screenshot.

Key	Value
system	all

5. Click the Save button and then the Send button.

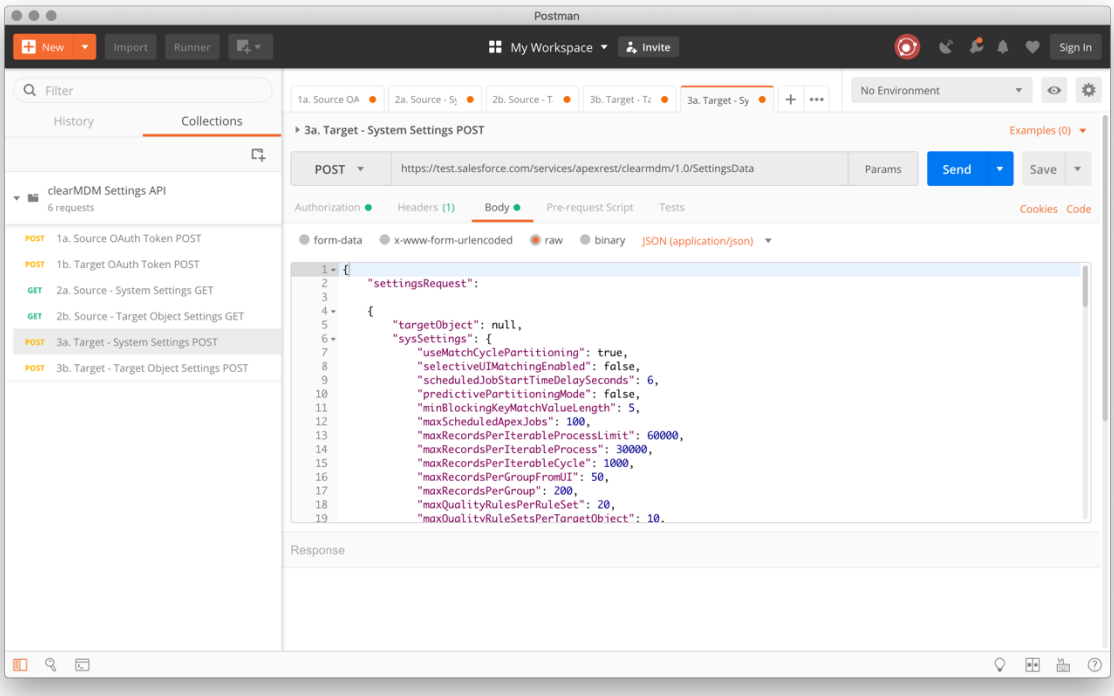
6. A response similar to the example below should be returned.


```
{
    "targetObject": null,
    "sysSettings": {
        "useMatchCyclePartitioning": false,
        "selectiveUIMatchingEnabled": false,
        "scheduledJobStartTimeDelaySeconds": 5,
        "predictivePartitioningMode": false,
        "minBlockingKeyMatchValueLength": 5,
        "maxScheduledApexJobs": 100,
        "maxRecordsPerIterableProcessLimit": 60000,
        "maxRecordsPerIterableProcess": 30000,
        "maxRecordsPerIterableCycle": 500,
        "maxRecordsPerGroupFromUI": 100,
        "maxRecordsPerGroup": 500,
        "maxQualityRulesPerRuleSet": 20,
        "maxQualityRuleSetsPerTargetObject": 10,
        "maxQualityReferenceRulesPerRuleSet": 5,
        "maxQualityActionsPerRuleSet": 10,
        "maxPredicatesPerNonIterableProcess": 200,
        "maxNormalisationRulesPerTargetObject": 10,
        "maxNormalisationRefSettingRecords": 1000,
        "maxMRPForManualMerge": 10,
        "maxMRGMergedPerJob": 10000,
        "maxMatchCyclesPerIterableCycle": 25000, ..truncated from here
    }
}
```


System Settings Deploy (POST)

The following steps describe how System Settings are deployed using the Postman utility application as the API client.

1. Create a new POST Request named System Settings POST.



2. Set the Type to POST and the URL to the *instance_url* response value retrieved in the Access Tokens section (point 8) with the path defined as below.

   `{instance_url}/services/apexrest/clearmdm/1.0/SettingsData`

3. Set the Authorisation Token to the *access_token* response value retrieved in the Access Tokens section (point 8).

4. Set the Body format type to raw and the content type to *application/json*, add the text below to the Body input and paste the settings JSON (retrieved in the Settings GET section) to overwrite the placeholder text *[paste JSON here]*. The result should be comparable to the screenshot above in structure.

```
{
    "settingsRequest":
      [paste JSON here]
 }
```

5. Click the Save button and then the Send button.

6. A response which includes the parameter [isSuccess=True] should be returned. If this is not the case the clearMDM Audit Log should be referenced for further information.

Target Object Settings

The Settings API supports 2 modes of settings retrieval; System Settings and Target Object settings. The latter is covered in this section.

Target Object Settings cover all settings related to a given Target Object, namely:

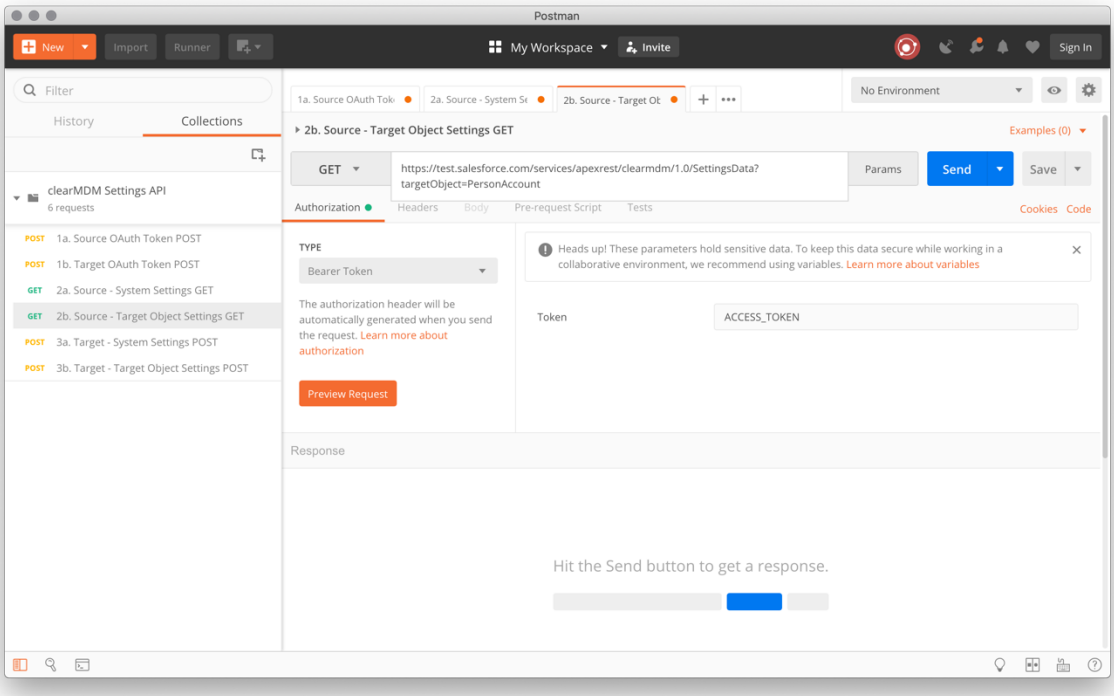
- Normalisation, Matching, Synchronisation, Merge and Re-parenting Settings
- Data Sources
- Attribute Groups
- Data Quality Rulesets
- Data Services

All Target Objects must be retrieved and deployed individually to complete the application configuration deployment.

Target Object Settings Retrieve (GET)

The following steps describe how Target Object Settings are retrieved using the Postman utility application as the API client.

1. Create a new GET Request named Target Object Settings GET.



2. Set the Type to GET and the URL to the *instance_url* response value retrieved in the Access Tokens section (point 8) with the path defined as below.

```
{instance_url}/services/apexrest/clearmdm/1.0/SettingsData
```

3. Set the Authorisation Token to the *access_token* response value retrieved in the Access Tokens section (point 8).

4. Enter the Key Value pair as defined below. Or add a URL parameter as per the preceding screenshot.

Key	Value
targetObject	The Target Object Name e.g. PersonAccount

5. Click the Save button and then the Send button.

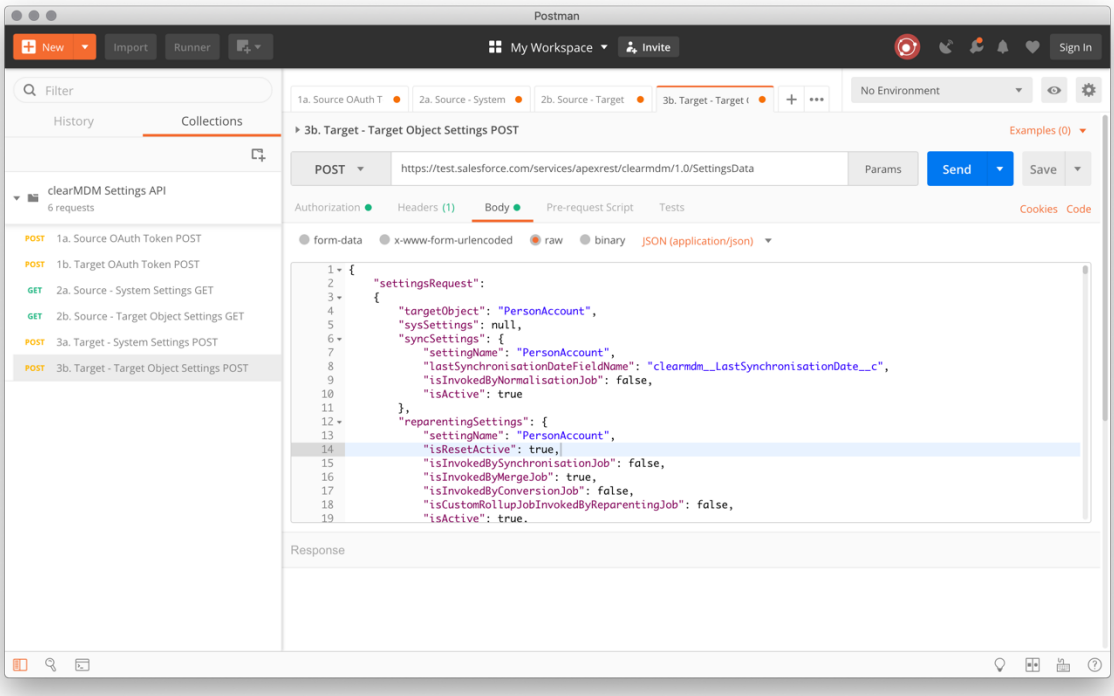
6. A response similar to the example below should be returned.


```
{
    "targetObject": "PersonAccount",
    "sysSettings": null,
    "syncSettings": {
        "settingName": "PersonAccount",
        "lastSynchronisationDateFieldName": "clearmdm__LastSynchronisationDate__c",
        "isInvokedByNormalisationJob": false,
        "isActive": true
    },
    "reparentingSettings": {
        "settingName": "PersonAccount",
        "isResetActive": true,
        "isInvokedBySynchronisationJob": false,
        "isInvokedByMergeJob": true,
        "isInvokedByConversionJob": false,
        "isCustomRollupJobInvokedByReparentingJob": false,
        "isActive": true,
        "customRollupsActive": false
    },
    "n11nSettings": {
        "settingName": "PersonAccount",
        ..truncated from here
    }
}
```


Target Object Settings Deploy (POST)

The following steps describe how Target Object Settings are deployed using the Postman utility application as the API client.

1. Create a new POST Request named Target Object Settings POST.



2. Set the Type to POST and the URL to the *instance_url* response value retrieved in the Access Tokens section (point 8) with the path defined as below.

   `{instance_url}/services/apexrest/clearmdm/1.0/SettingsData`

3. Set the Authorisation Token to the *access_token* response value retrieved in the Access Tokens section (point 8).

4. Set the Body format type to raw and the content type to *application/json*, add the text below to the Body input and paste the settings JSON (retrieved in the Settings GET section) to overwrite the placeholder text *[paste JSON here]*. The result should be comparable to the screenshot above in structure.

```
{
    "settingsRequest":
      [paste JSON here]
 }
```

5. Click the Save button and then the Send button.

6. A response which includes the parameter [isSuccess=True] should be returned. If this is not the case the clearMDM Audit Log should be referenced for further information.

Example API Scenario

DEV sandbox to SIT sandbox deployment

clearMDM Application Configuration:

- PersonAccount Target Object (all settings)
- SAP Customers Target Object (normalisation and data quality settings only)

Steps:

1. GET System Settings from DEV.
2. POST System Settings to SIT.
3. GET Target Object Settings (SAP Customers) from DEV.
4. POST Object Settings (SAP Customers) to SIT.
5. GET Target Object Settings (PersonAccount) from DEV.
6. POST Object Settings (PersonAccount) to SIT.

Between each step the clearMDM Audit Log should be checked for missing fields where success indicator (isSuccess) is true but the response error text is set to “Missing Permissions”. If the success indicator is false the Audit Log will describe the error condition that has caused the API interaction to fail.

