

# Prodly Moover Integration Guide

## Table of Contents

---

<b>Overview</b>	<b>2</b>
<b>Salesforce Package Application Programming Interface</b>	<b>2</b>
Deployment Web Service	2
Usage	2
Namespace	3
Ways to Reference the REST Endpoint	3
Sample URLs	3
Sample Command Line Interface (CLI) Code	3
Related Salesforce Documentation	4
Deployment Apex Service	4
Usage	4
Namespace	4
Methods	4
Parameters	5
Return Value	5
Sample Code	5
<b>Deployment Scheduling</b>	<b>6</b>
<b>Salesforce DX Integration</b>	<b>7</b>
<b>API Error Messages</b>	<b>7</b>

## Overview

---

This guide covers topics of interest to systems integrators, release engineers, and developers integrating Prodlly Moover as part of their development process. For example, for:

- Integration with a release management system
- Automatic periodic updates to refresh an existing sandbox

Prodlly supports API requests for Apex and REST to accomplish these goals. Prodlly also offers some support for Salesforce DX integration.

## Salesforce Package Application Programming Interface

---

This section describes Moover's application programming interface (API). The API helps integrate continuous delivery and release management systems to automatically perform Moover deployments. For example, use API requests to deploy a data set or deployment plan.

The Moover API is a RESTful API that allows you to integrate Moover deployments into a larger continuous integration process. Moover API requests use the Force.com REST API to invoke Moover actions. Moover also supports global Apex services for internal Salesforce automation.

The following API requests are available:

- Deployment Web Service
- Deployment Apex Service

## Deployment Web Service

Deployment Web Service is REST web service specific to Prodlly Moover that provides the ability for Moover consumers and integrated systems outside of Salesforce to invoke deployments and perform other actions through REST-based API requests exposed by the Moover Salesforce package in your Moover control organization.

## Usage

The Deployment Web Service allows you to deploy deployment plans and data sets from applications external to the Salesforce, such as a continuous delivery or release management system. For example:

- You are running CircleCI and after deploying updated code to a QA sandbox, you need to automatically deploy the updated data to the sandbox for end to end testing.
- You have a custom built solution coordinating CPQ deployments that performs automated steps to promote product catalog changes between development, QA, and UAT sandboxes.

## Namespace

PDRI

## Ways to Reference the REST Endpoint

- For deployment plans:
  - /dataset/deploy/plan/\*
  - `https://<org_instance>.salesforce.com/services/apexrest/PDRI/dataset/deploy/plan/<deployment_plan_id>?targetConnectionId=<connection_id>`
  - `https://<custom_domain>.my.salesforce.com/services/apexrest/PDRI/dataset/deploy/plan/<deployment_plan_id>?targetConnectionId=<connection_id>`
- For data sets:
  - /dataset/deploy/\*
  - `https://<org_instance>.salesforce.com/services/apexrest/PDRI/dataset/deploy/<data_set_id>?targetConnectionId=<connection_id>`
  - `https://<custom_domain>.my.salesforce.com/services/apexrest/PDRI/dataset/deploy/<data_set_id>?targetConnectionId=<connection_id>`

## Sample URLs

- `https://na8.salesforce.com/services/apexrest/PDRI/dataset/deploy/a000H00400s6WikQAE?targetConnectionId=a030H00120XIWDzQAP`
- `https://prodlytest-dev-ed.my.salesforce.com/services/apexrest/PDRI/dataset/deploy/a000H00400s6WikQAE?targetConnectionId=a030H00120XIWDzQAP`

## Sample Command Line Interface (CLI) Code

### Code Format

```
$ curl -H "Authorization: Bearer <session_id>" "<endpoint>"
```

### Code Sample

```
$ curl -H "Authorization: Bearer  
BBASDQMFfAfpRjOw4dE5clp0vzbeV98wgl460GrL3eOLfghnSpfFeoeqb.HH0XurSpXjC  
eHAjZZJioJlmagcmixMxpED6xe0n"  
"https://prodlytest-dev-ed.my.salesforce.com/services/apexrest/PDRI/d  
ataset/deploy/a000H00400s6WikQAE?targetConnectionId=a030H00120XIWDzQA  
P"
```

### JSON Response

```
{  
  "resultIds":null,"resultId":"a040H0000168iXcQAI","error":null
```

```
}
```

## Related Salesforce Documentation

[https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex\\_rest.htm](https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_rest.htm)

## Deployment Apex Service

The DeploymentService class is an Apex service specific to Prodly Moover.

### Usage

The DeploymentService class allows you to deploy deployment plans and data sets using the Apex API. Depending on the parameters you specify, the API request migrates data from your control organization or a source organization you specify to one or more destination organizations.

### Namespace

PDRI

### Methods

The DeploymentPlanService class provides the following methods for deployment plans:

- **ID deploy(deploymentPlanId, targetConnectionId)** – Deploys the given deployment plan, migrating data from the control organization to the given destination organization.
- **ID deploy(deploymentPlanId, sourceConnectionId, targetConnectionId)** – Deploys the given deployment plan, migrating data from the given source organization to the given destination organization.
- **List<ID> deploy(deploymentPlanId, targetConnectionIds)** – Deploys the given deployment plan, migrating data from the control organization to the given list of destination organizations.
- **List<ID> deploy(deploymentPlanId, sourceConnectionId, targetConnectionIds)** – Deploys the given deployment plan, migrating data from the given source organization to the given list of destination organizations.

The DeploymentService class provides the following methods for data sets:

- **ID deploy(rootDataSetId, targetConnectionId)** – Deploys the given data set, migrating data from the control organization to the given destination organization.
- **ID deploy(rootDataSetId, sourceConnectionId, targetConnectionId)** – Deploys the given data set, migrating data from the given source organization to the given destination organization.

- **List<ID> deploy(rootDataSetId, targetConnectionIds)** – Deploys the given data set, migrating data from the control organization to the given list of destination organizations.
- **List<ID> deploy(rootDataSetId, sourceConnectionId, targetConnectionIds)** – Deploys the given data set, migrating data from the given source organization to the given list of destination organizations.

## Parameters

The `deployPlan()` method accepts the following parameters:

- **ID deploymentPlanId** – The ID of the deployment plan to deploy.
- **ID sourceConnectionId** – The ID of the Moover connection to the source organization from which to migrate data.
- **ID targetConnectionId** – The ID of the Moover connection to the destination organization that receives the migrated data.
- **List<ID> targetConnectionIds** – A list of IDs of the Moover connections to up to five destination organizations that receive the migrated data.

The `deploy()` method accepts the following parameters:

- **ID rootDataSetId** – The ID of the data set to deploy.
- **ID sourceConnectionId** – The ID of the Moover connection to the source organization from which to migrate data.
- **ID targetConnectionId** – The ID of the Moover connection to the destination organization that receives the migrated data.
- **List<ID> targetConnectionIds** – A list of IDs of the Moover connections to up to five destination organizations that receive the migrated data.
- 

## Return Value

Depending on the parameters you pass in, `deployPlan()` and `deploy()` return one of the following values:

- **ID** – The ID of the result record.
- **List<ID>** – A list of result record IDs, one for each destination organization.

## Sample Code

```
ID rootDataSetId = [SELECT Id from PDRI__DataSet__c where Name =
'<data_set_name>' LIMIT 1].Id;
ID targetConnectionId = [SELECT Id from PDRI__Connection__c where
Name = '<connection_name>' LIMIT 1].Id;
ID deploymentResultId = PDRI.DeploymentService.deploy(rootDataSetId,
targetConnectionId);
```

# Deployment Scheduling

Moover provides the `class DeploymentSchedulable` Apex wrapper that you can call to schedule your deployments. Pass in the data set ID, and source and destination organization IDs. Refer to the following screenshot from **Setup > Custom Code > Apex Classes > Class Summary** for details:

The screenshot shows the Salesforce Apex Class Summary page for the global class `DeploymentSchedulable`. The page is titled "Apex Classes" and includes tabs for "Class Body", "Class Summary", "Version Settings", and "Trace Flags". The class is available in versions 1.31 through the current version. It implements the `Schedulable` interface. The class has four private properties: `rootDataSetId`, `sourceConnectionId`, `targetConnectionId`, and `targetConnectionIds`. It has four constructors and one public method named `execute` that takes a `System.SchedulableContext` object as a parameter.

Access	Name	Available in Versions
global	Schedulable	

Access	Name	Available in Versions
private	Id rootDataSetId	
private	Id sourceConnectionId	
private	Id targetConnectionId	
private	List targetConnectionIds	

Access	Signature	Available in Versions
global	DeploymentSchedulable(Id rootDataSetId, Id targetConnectionId)	
global	DeploymentSchedulable(Id rootDataSetId, List targetConnectionIds)	
global	DeploymentSchedulable(Id rootDataSetId, Id sourceConnectionId, Id targetConnectionId)	
global	DeploymentSchedulable(Id rootDataSetId, Id sourceConnectionId, List targetConnectionIds)	

Access	Signature	Available in Versions
global	void execute(System.SchedulableContext context)	1.31 - Current

The following steps provide one way to implement deployment scheduling:

1. In the upper right of your org, click your name in Classic or the gear icon in Lightning Experience, and select **Developer Console**.
2. Navigate to **Debug > Open Execute Anonymous Window**.
3. Paste the following code:

```
Id rootDataSetId = [SELECT Id from PDRI__DataSet__c where
Name = '<data set name>' LIMIT 1].Id;
Id targetConnectionId = [SELECT Id from
PDRI__Connection__c where Name = '<connection name>' LIMIT
1].Id;
DeploymentSchedulable deploymentSchedulable = new
DeploymentSchedulable(rootDataSetId, targetConnectionId);

String schedule = '0 27 11 23 3 ? 2018';
String jobId = System.schedule('Data Set Deployment Job',
schedule, deploymentSchedulable);
```

4. Modify the code to use the appropriate constructor (refer to screenshot for choices) and schedule string.
5. Modify `<data set name>` and `<connection name>` in the query strings to match your names.
6. Execute the code.

## Salesforce DX Integration

---

To migrate data to a scratch org, perform the following steps:

1. From your SFDX command line interface (CLI), generate a password to your scratch org. For example,
 

```
sfdx force:user:password:generate --targetusername
test-rvrfodezg8h9@example.com
```
2. In the Moover app, on the **Connections** tab, create a connection of **Organization Type** Sandbox to your scratch org using the org name and password login credentials.
3. In the Moover app, deploy a data set using the scratch org connection as your destination org.
4. In the Moover app, monitor the data migration on the Deployment Results tab.

## API Error Messages

---

The following table lists errors the API can return.

Error Code	Error Key	Error Message
209	INVALID_DEPLOYMENT_PLAN_ID	Invalid deployment plan id <deploymentPlanId> in API request.
210	INVALID_DATA_SET_ID	Invalid data set id <dataSetId> in API request.
211	INVALID_SOURCE_CONNECTION	Invalid source organization connection id <sourceConnectionId> in API request.
212	INVALID_DESTINATION_CONNECTION	Invalid destination organization connection id <targetConnectionId> in API request.
213	TOO_MANY_DESTINATIONS_ORG_IDS	Too many destination orgs. Please limit the number of destination orgs to five per deployment.