

Create Your First Unlocked Package



Robert Davis
Solution Architect, Hike2
Twitter: @robertdavis



SOUTHEAST
DREAMIN'

GOALS

- Understand what are the types of 2GP Packages.
- The advantages and disadvantages of second generation packaging
- How to set up a Dev Hub
- Creation of an Unlocked Package
- Promotion of a Beta Package to a Release Package

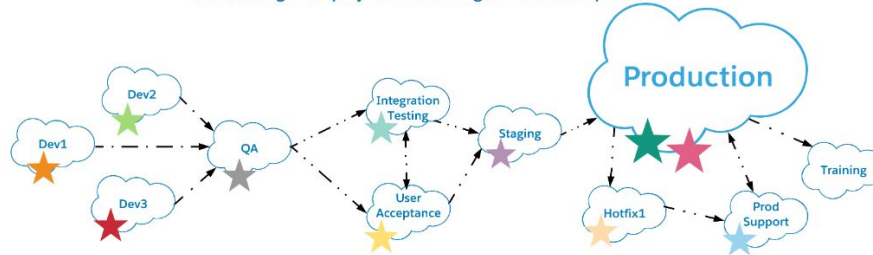
What is a Package

- Container for your metadata:
 - Flows
 - Fields
 - Objects
 - Code
 - Assignment Rules
 - Page Layouts
 - Etc...
- Types
 - Managed - First Generation and Second Generation
 - Unmanaged - not upgradable
 - Unlocked - upgradable

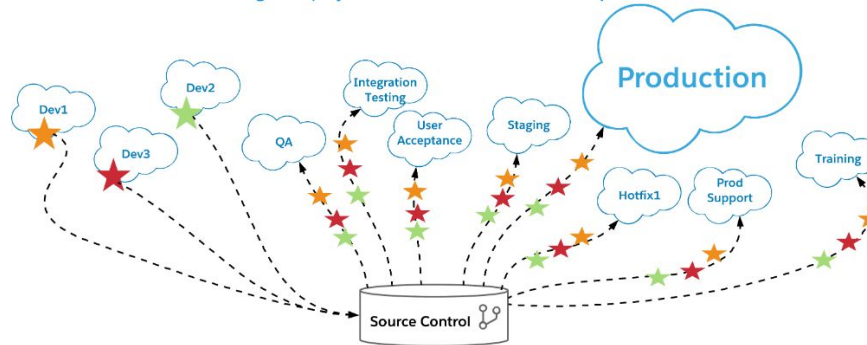
Packaging in Modular Development

★ = Group of metadata

Non-Packaged Deployments and Org-Based Development



Packaged Deployments and Source-Based Development



Salesforce Developer Blog -
Zayne Turner

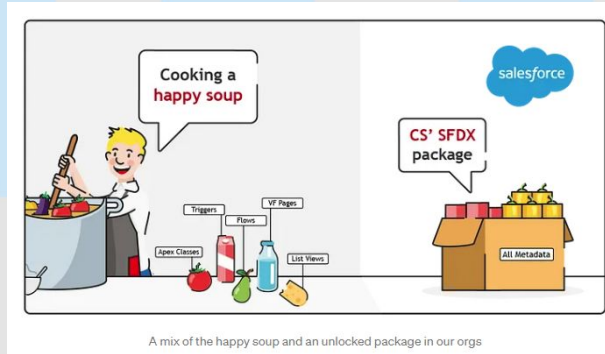
Advantages of Unlocked Packages

- Version Control Centric as opposed to Org Centric
- It helps you to add, edit, and remove metadata in your org in a trackable way. You can install unlocked package to multiple orgs, and upgrade your Salesforce apps easier and faster. (Apex Hours)
- Forces an organization to really think about their architecture and arguably their road map.
- Easier to create pipeline because command based and not configuration based
- Pipelines with things like Github actions make deployment and through put faster

Disadvantages of Unlocked Packages

Overhead and complexity to setting up the infrastructure:

- How to split up the “Happy Soup”
 - The Salesforce Org and all its metadata.



The Salesforce Makeover Project Blog - Tachu Salamanca

- Research and architecture to divide org into vertical or horizontal packages:
 - Deciding what should be packaged
 - Vertical by functionality / application
 - Horizontally by Client Layer, Service Layer

Illustrations of Splitting an Org



By Bill Appleton in Medium blog : Salesforce DX for Admins

History

- Second Generation Packaging includes Managed and Unlocked Packages released in October 2019 (Winter 2020 Release)
- Unlocked Packages now allowed for publishing of packages that were upgradeable and with a namespace (be careful with this).
- It allowed for tracking of changes with version control
- Now could use a single namespace for multiple Managed Packages.
 - Reduced the need for Extension Packages in Manage Packages
- Made automation of CI easier with sfdx command line

Creating an Unlocked Package

- Setup a Dev Hub
- Authorize a Dev Hub in VS Code
- Create an Unlocked Package
- Create a Beta Unlocked Package
- Look at some list metadata about the package
- Create a Release Package
- Create a Scratch Org
- Install the package via command line
- View a Release Package Report

Additional Resources

Developer Salesforce Blog : [Working with Modular Development and Unlocked Packages: Part 1](#)

Trailhead Module : [Build your first Unlocked Package](#)

Developer Salesforce Help : [Workflow for unlocked Packages](#)

Salesforce Developer YouTube by Mohith Shrivastave : [Successfully Creating Unlocked Packages](#)

Salesforce DX Developer Guide : [Unlocked Packages](#)

Developer Salesforce Help : [Configure Packages](#)

Metadata API Guide : [Metadata Coverage Report](#)

[The Salesforce Makeover Project](#) by Tachu Salamanca

[Salesforce DX for Admins](#) by Bill Appleton

[FAQ on Unlocked Packages](#)

What we Did

- Set up your Dev Hub
 - Created a Package
 - Created a Version of that Package
 - Promoted Package to a Released Package
-
- Steps and Recipe to try at home at:
 - <https://github.com/robtdavis/CreateAnUnlockedPackage>
 - ReadMe

Setup Dev Hub

sfdx org login web --set-default-dev-hub --alias DevHub



Allow Access?

Salesforce CLI is asking to:

- Access the identity URL service
- Manage user data via Web browsers
- Manage user data via APIs
- Perform requests at any time

Do you want to allow access for
rdavis7408+devhub202310@gmail.com? (Not you?)

Deny

Allow

To revoke access at any time, go to your personal settings.

The screenshot shows the Salesforce Setup interface. The left sidebar has a search bar with 'Dev Hub' entered, and the 'Dev Hub' link is highlighted under the 'Development' section. The main content area shows the 'Enable Dev Hub' toggle switch, which is currently 'Disabled'. A red arrow points to this toggle. Below this, there are sections for 'Enable Source Tracking in Sandboxes' and 'Enable Einstein Features', both with 'Disabled' toggle switches. The 'Enable Source Tracking in Sandboxes' section includes a list of bullet points and a 'Notes' section. The 'Enable Einstein Features' section also includes a list of bullet points. At the bottom, there is a note about sandbox deletion.

Search Setup

Setup Home Object Manager

Q Dev Hub

Development

Dev Hub

Didn't find what you're looking for?
Try using Global Search.

2. View information about your scratch orgs
Information includes details about requested scratch orgs, including whether they are active, expired or deleted.

3. Link namespace orgs
You can link namespace orgs to the Dev Hub for using scratch orgs with namespaces.

Note: You can't disable this setting.

Enable Dev Hub

Enable Source Tracking in Sandboxes to:

- Track changes that are made in Developer and Developer Pro Sandboxes. Use the Salesforce Command Line Interface (CLI) to view, pull, and push changes in the Sandbox.

Notes:

- After you enable Source Tracking in Sandboxes, Developer and Developer Pro Sandboxes that you create or refresh have Source Tracking enabled automatically.
- You don't need to have Dev Hub enabled to enable Source Tracking in Sandboxes.

Enable Source Tracking in Developer and Developer Pro Sandboxes

Enable Einstein Features

- Some features like Einstein Bots require you to accept the Terms of Service for Einstein before you enable them.
- When you accept the Terms of Service for Einstein in this org, a separate acceptance is not required in the scratch orgs created from this org's Dev Hub.
- If you've previously accepted the Terms of Service for Einstein to turn on an Einstein-related feature, this setting is already enabled.

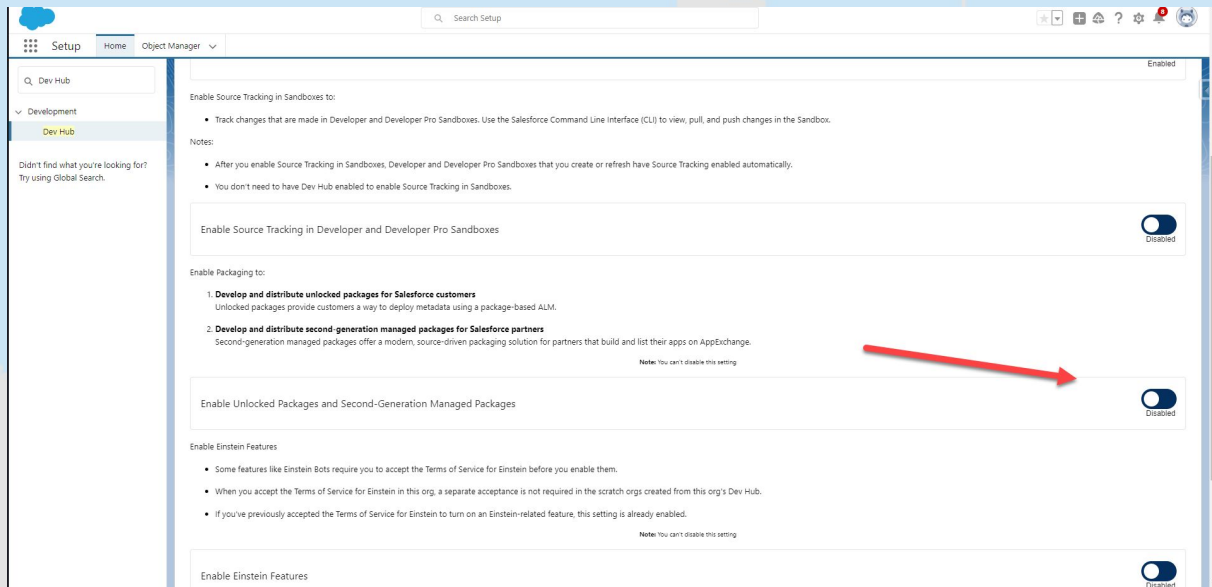
Note: You can't disable this setting.

Enable Einstein Features

Salesforce notifies sandbox users of impending deletion of sandboxes that have been inactive for 180 days or longer. Click the Enabled toggle to:

SOUTHEAST
DREAMIN'

Setup Dev Hub

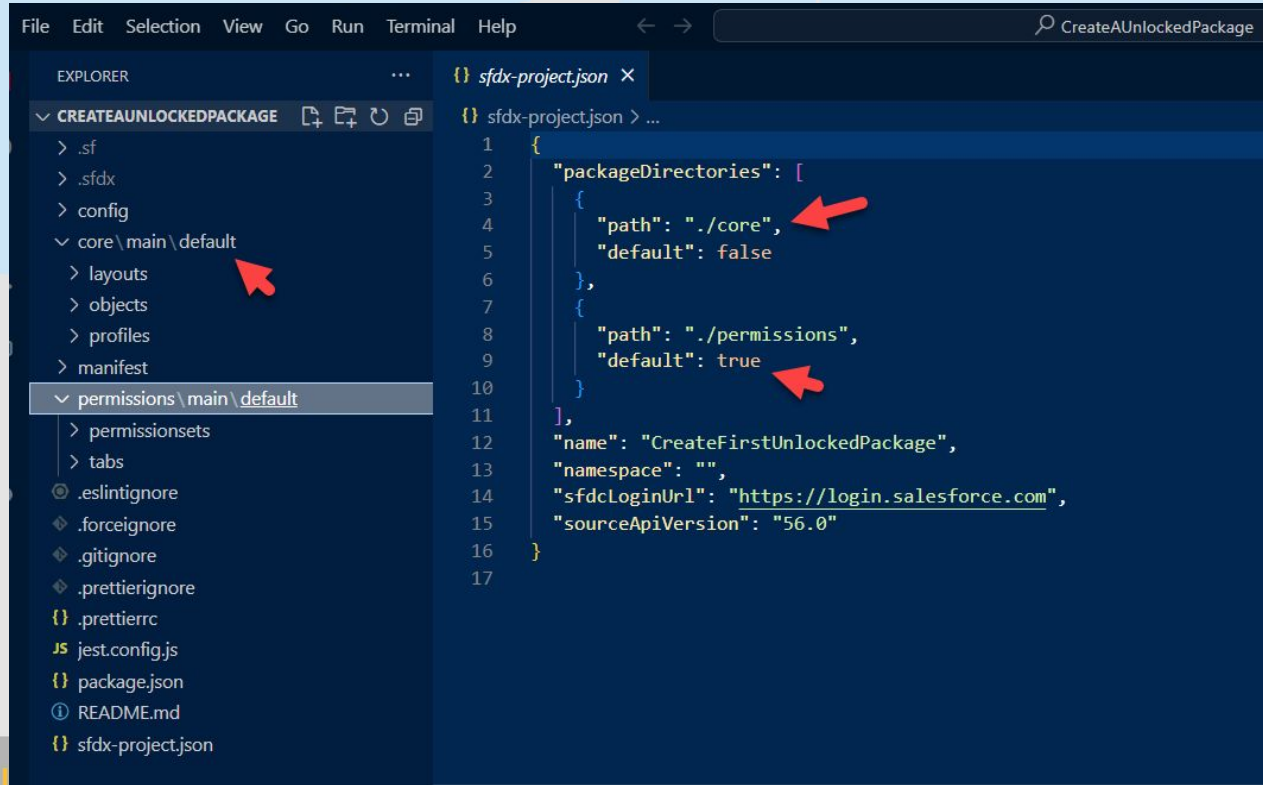


The screenshot shows the Salesforce Setup interface. The left sidebar has a search bar with 'Dev Hub' entered and a list of categories including 'Development' and 'Dev Hub'. The main content area is titled 'Dev Hub' and contains several sections:

- Enable Source Tracking in Sandboxes:** A section with a list of instructions and a toggle switch labeled 'Enabled'.
- Enable Source Tracking in Developer and Developer Pro Sandboxes:** A section with a toggle switch labeled 'Disabled'.
- Enable Packaging to:** A section with two numbered items and a toggle switch labeled 'Disabled'. A red arrow points to this toggle.
- Enable Einstein Features:** A section with a list of instructions and a toggle switch labeled 'Disabled'.

```
RobertDavis@DESKTOP-TF4N9AJ MINGW64 ~/Desktop/Not Hike2/sed/CreateAnUnlockedPackage (main)
$ sfdx org login web --set-default-dev-hub --alias DevHub
Successfully authorized rdavis7408+devhub202310@gmail.com with org ID 00DDo000001C9P1MAK
```

GitHub Repo



Create A Package

```
sfdx package:create --name "Core" --package-type Unlocked --path "./core"
--description "Core fields and Objects"
```

```
RobertDavis@DESKTOP-TF4N9AJ MINGW64 ~/Desktop/Not Hike2/sed/CreateAUnlockedPackage (main)
```

```
$ sfdx package:create --name "Core" --package-type Unlocked --path "./core" --description "Core fields and Objects"
=== Ids
```

NAME	VALUE
Package Id	0HoDo000000fxzmKAA

Create A Package

Sfdx-project.json file

```
{ } sfdx-project.json M X
{ } sfdx-project.json > ...
1  {
2    "packageDirectories": [
3      {
4        "path": "./core",
5        "default": false,
6        "package": "Core",
7        "versionName": "ver 0.1",
8        "versionNumber": "0.1.0.NEXT",
9        "versionDescription": "Core fields and Objects"
10     },
11     {
12       "path": "./permissions",
13       "default": true
14     }
15   ],
16   "name": "CreateFirstUnlockedPackage",
17   "namespace": "",
18   "sfdcLoginUrl": "https://login.salesforce.com",
19   "sourceApiVersion": "56.0",
20   "packageAliases": {
21     "Core": "0HoDo000000fxzmKAA"
22   }
23 }
```

Create a Package Version

```
sfdx package:version:create --package "Core" --installation-key-bypass --wait 10  
--target-dev-hub DevHub --code-coverage
```

```
RobertDavis@DESKTOP-TF4N9AJ MINGW64 ~/Desktop/Not Hike2/sed/CreateAUnlockedPackage (main)  
$ sfdx package:version:create --package "Core" --installation-key-bypass --wait 10 --target-dev-hub DevHub --code-coverage  
Version create.... Create version status: Success  
Successfully created the package version [08cDo000000kABJIA2]. Subscriber Package Version Id: 04tDo00000058jgIAA  
Package Installation URL: https://login.salesforce.com/packaging/installPackage.apexp?p0=04tDo00000058jgIAA  
As an alternative, you can use the "sfdx package:install" command.
```

Create a Package Version

```
{  
  {} sfdx-project.json > ...  
  1 {  
  2   "packageDirectories": [  
  3     {  
  4       "path": "./core",  
  5       "default": false,  
  6       "package": "Core",  
  7       "versionName": "ver 0.1",  
  8       "versionNumber": "0.1.0.NEXT",  
  9       "versionDescription": "Core fields and Objects"  
 10    },  
 11    {  
 12       "path": "./permissions",  
 13       "default": true  
 14    }  
 15  ],  
 16  "name": "CreateFirstUnlockedPackage",  
 17  "namespace": "",  
 18  "sfdcLoginUrl": "https://login.salesforce.com",  
 19  "sourceApiVersion": "56.0",  
 20  "packageAliases": {  
 21    "Core": "0HoDo000000fxzmKAA",  
 22    "Core@0.1.0-1": "04tDo00000058jgIAA"  
 23  }  
 24 }
```

Metadata View of Package

sfdx package:version:list

RobertDavis@DESKTOP-TF4N9AJ MINGW64 ~/Desktop/Not Hike2/sed/CreateAUnlockedPackage (main)

\$ sfdx package:version:list

=== Package Versions [1]



Package Name	Namespace	Version Name	Version	Subscriber	Package Version Id	Alias	Installation Key	Released	Validation Skipped	Ancestor	Ancestor Version	Branch
Core		ver 0.1	0.1.0.1	04tDo00000058jgIAA		Core@0.1.0-1	false	false	false	N/A	N/A	

Metadata View of Package

sfdx package:create:version:list

RobertDavis@DESKTOP-TF4N9AJ MINGW64 ~/Desktop/Not Hike2/sed/CreateAUnlockedPackage (main)

\$ sfdx package:create:version:list

=== Package Version Create Requests [1]

Id	Status	Package Id	Package Version Id	Subscriber	Package Version Id	Tag	Branch	Created Date	Created By
08cDo000000kABJIA2	Success	0HoDo000000fxzmKAA	05iDo000000fxWYIAY	04tDo00000058jgIAA				2023-04-30 10:40	005Do000001zLibIAE

Promote Beta Package to Release Package

```
sfdx package:version:promote --package "Core@0.1.0-1"
```

```
RobertDavis@DESKTOP-TF4N9AJ MINGW64 ~/Desktop/Not Hike2/sed/CreateAnUnlockedPackage (main)
```

```
$ sfdx package:version:promote --package "Core@0.1.0-1"
```

```
? Are you sure you want to release package version Core@0.1.0-1? You can't undo this action. Release package (y/n)? Yes
```

```
Successfully promoted the package version, ID: 04tDo00000058jgIAA, to released. Starting in Winter '21, only unlocked package versions that have met the minimum 75 % code coverage requirement can be promoted. Code coverage minimums aren't enforced on org-dependent unlocked packages.
```

Package Report

```
sfdx package:version:report --package "Core@0.1.0-1"
```

```
RobertDavis@DESKTOP-TF4N9AJ MINGW64 ~/Desktop/Not Hike2/sed/CreateAUnlockedPackage (main)
```

```
$ sfdx package:version:report --package "Core@0.1.0-1"
```

```
=== Package Version
```

Name	Value
Name	ver 0.1
Subscriber Package Version Id	04tDo000000058jgIAA
Package Id	0HoDo0000000fxzmKAA
Version	0.1.0.1
Description	Core fields and Objects
Branch	
Tag	
Released	true
Validation Skipped	false
Ancestor	N/A
Ancestor Version	N/A
Code Coverage	100.00%
Code Coverage Met	true
Org-Dependent Unlocked Package	No
Release Version	57.0
Build Duration in Seconds	57.0
Managed Metadata Removed	N/A
Created By	005Do000001zLibIAE

Create Scratch Org

```
sfdx force:org:create -s -f config/project-scratch-def.json
```

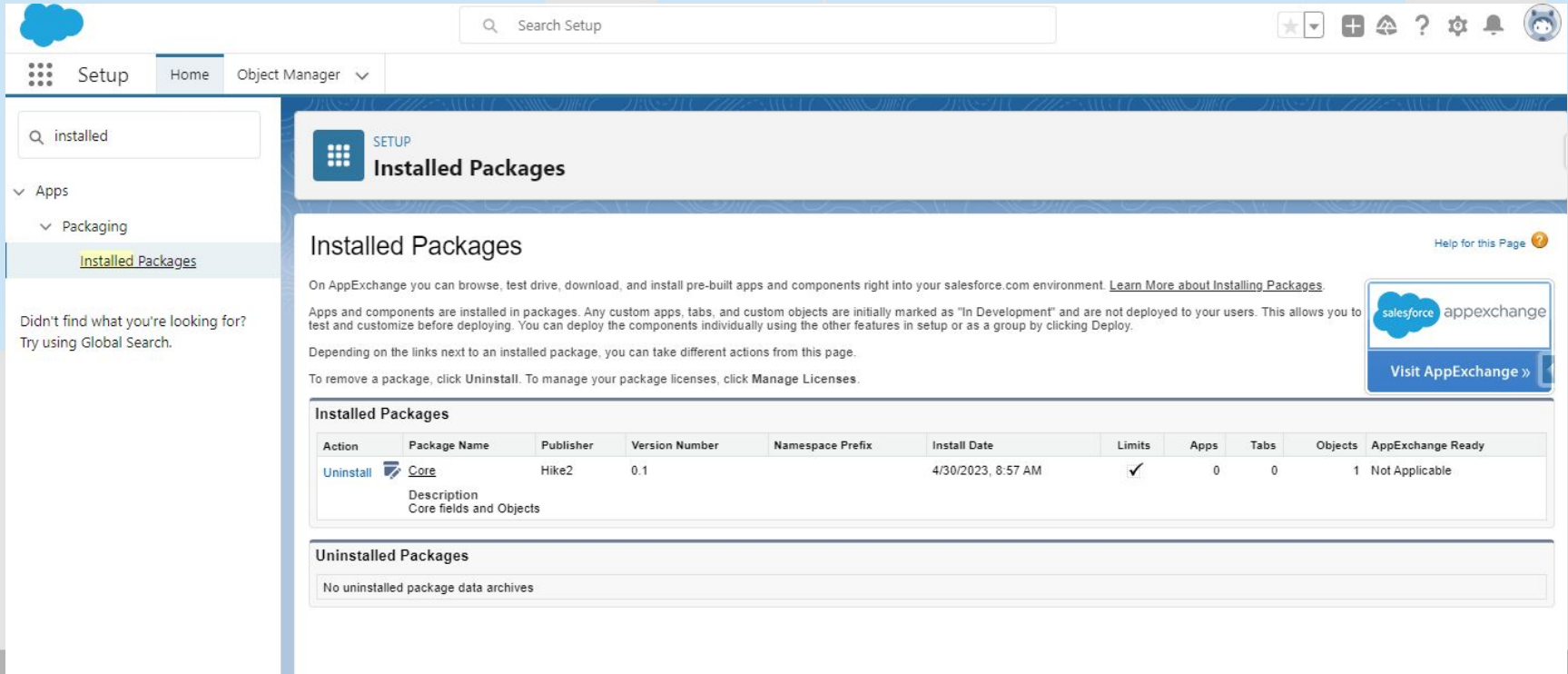
```
RobertDavis@DESKTOP-TF4N9AJ MINGW64 ~/Desktop/Not Hike2/sed/CreateAUnlockedPackage (main)
$ sfdx force:org:create -s -f config/project-scratch-def.json
Successfully created scratch org: 00D6t0000005C0yEAM, username: test-wyurp7jwavjw@example.com.
```

Install Package

```
sfdx package:install --package "Core@0.1.0-1" --target-org  
test-wyurp7jwavjw@example.com --wait 10 --publish-wait 10
```

```
RobertDavis@DESKTOP-TF4N9AJ MINGW64 ~/Desktop/Not Hike2/sed/CreateAUnlockedPackage (main)  
$ sfdx package:install --package "Core@0.1.0-1" --target-org test-wyurp7jwavjw@example.com --wait 10 --publish-wait 10  
10 minutes remaining until timeout. Publish status: Querying Status... done  
Waiting 10 minutes for package install to complete.... done  
Successfully installed package [Core@0.1.0-1]
```

Installed Package



The screenshot shows the Salesforce Setup interface. The top navigation bar includes the Setup icon, a search bar labeled "Search Setup", and various utility icons. The left sidebar contains a search bar with "installed" and a list of navigation items: "Apps", "Packaging", and "Installed Packages" (which is highlighted). Below the sidebar, the main content area is titled "Installed Packages" and includes a "Help for this Page" link. The content area contains a paragraph about AppExchange, a "Visit AppExchange" button, and a table of installed packages. The table has columns for Action, Package Name, Publisher, Version Number, Namespace Prefix, Install Date, Limits, Apps, Tabs, Objects, and AppExchange Ready. One package, "Core" by Hike2, is listed with version 0.1, installed on 4/30/2023 at 8:57 AM, and marked as AppExchange Ready. Below the table, there is a section for "Uninstalled Packages" which states "No uninstalled package data archives".

Setup

Home Object Manager

installed

Apps

Packaging

Installed Packages

Didn't find what you're looking for?
Try using Global Search.

Installed Packages

Help for this Page

On AppExchange you can browse, test drive, download, and install pre-built apps and components right into your salesforce.com environment. [Learn More about Installing Packages.](#)

Apps and components are installed in packages. Any custom apps, tabs, and custom objects are initially marked as "In Development" and are not deployed to your users. This allows you to test and customize before deploying. You can deploy the components individually using the other features in setup or as a group by clicking Deploy.

Depending on the links next to an installed package, you can take different actions from this page.

To remove a package, click Uninstall. To manage your package licenses, click [Manage Licenses](#).

Visit AppExchange »

Action	Package Name	Publisher	Version Number	Namespace Prefix	Install Date	Limits	Apps	Tabs	Objects	AppExchange Ready
Uninstall	Core	Hike2	0.1		4/30/2023, 8:57 AM	✓	0	0	1	Not Applicable
Description Core fields and Objects										

Uninstalled Packages

No uninstalled package data archives