
Drupal Entity Generator (DEG)

Release 1.0.0-beta1

Mukesh Sah

Sep 13, 2023

CONTENTS:

1	Introduction	1
1.1	Entity Types	1
2	About DEG Spreadsheet	3
2.1	Overview	3
2.2	Bundles	3
2.3	Fields	4
2.4	User roles	5
2.5	Workflows	6
2.6	Workflow states	6
2.7	Workflow transitions	7
2.8	Menus	8
2.9	Image styles	8
2.10	Image effects	9
3	Installation	11
3.1	Download using composer	11
3.2	Obtaining Google OAuth 2.0 Client Credentials	12
3.3	Enable Google Sheets APIS AND SERVICES	16
3.4	Configure DEG on Drupal	17
4	Drush Commands	19
4.1	Generate All Entities	19
4.2	Generate Content Types	19
4.3	Generate vocabularies	20
4.4	Generate Blocks Types	20
4.5	Generate Image Styles	21
4.6	Generate Media Types	21
4.7	Generate menus	22
4.8	Generate Paragraphs Types	22
4.9	Generate User Roles	23
4.10	Generate Workflows	23
5	Supported Entities	25
5.1	Bundles	25
5.2	Fields	25
5.3	Form field widgets	26
5.4	Other entities	27
6	References	29

INTRODUCTION

Drupal Entity Generator (DEG) is a tool used to build Drupal entities using the architecture defined in the Google spreadsheet. It provides drush commands to generate the following entity types:

1.1 Entity Types

- Content types
- Fields
- Vocabularies
- Block
- Menus
- Image styles
- Image effects
- Workflows
- Workflow states
- Workflow transitions
- User roles
- Media Types
- Paragraphs

By using this tool we can manage the entities of Drupal projects, based on structure defined in Google spreadsheet. It saves our time and improve the productivity of team.

ABOUT DEG SPREADSHEET

DEG Spreadsheet is a Google Spreadsheet template used for capturing structure of entities, Use [DEG sample template](#) for creation of new Google spreadsheet with same structure to generate different Drupal entities as per requirements.

2.1 Overview

This sheet is used to keep track of numbers of entities used in different sheets.

Table 1: Overview

Column	Specifications
Specification	List of different entity sheets
Done	Number of entities(rows) 'Implemented and done(x)' as per status of column X of that sheet.
Total	Number of entities(rows) defined.
%	Percentage of entities(rows) 'Implemented and done(x)'.

2.2 Bundles

A list of bundles of different entity types in Drupal.

Table 2: Bundles

Column	Creation support	Update support	Specifications
ID	No	No	An optional identifier for references in external documents.
Name	Yes	Yes	To keep name of entity types like Article, Page etc.
Machine name	Yes	No	Unique machine name for Drupal bundles.
X	Yes	Yes	Implementation status for bundles, a = Approved and ready to implement , w = Wait to implement, x = Implemented and done, - c = Changed since implemented, d = To be deleted
Dev	No	No	Developer initials, signifying that the row has been implemented as specified.
QA	No	No	Tester initials, signifying that the row has been validated.
Description	Yes	Yes	Description about the bundles.
Example(s)	No	No	One or more relevant examples, optionally hyperlinked.
Mod.	No	No	Moderated (Content moderation is enabled), for example: y = Yes, n = No
Layout	No	No	Layout-enabled, via Core Layout Builder or via Panels, for example: y = Yes, n = No
Trns.	No	No	Translatable, for example: y = Yes, n = No
Migr.	No	No	Migrated (Content will be populated via migration)
Cmnt.	No	No	Commenting is enabled, for example: y = Yes, n = No
Meta.	No	No	Meta tags are enabled, for example: y = Yes, n = No
Sched.	No	No	Schedulable (Scheduled updates are enabled), for example: y = Yes, n = No
Type	Yes	No	Entity type, for example: Content type, Vocabulary, Paragraph type, Media type, Custom block type
URL alias pattern	Yes	No	To configure URL alias pattern.
Settings/notes	Yes	No	To configure settings/notes related to bundle, for example: 'Image' for media type image

2.3 Fields

A list for fields which is required for bundles.

Table 3: Fields

Column	Creation support	Update support	Specifications
Bundle	Yes	No	Name of bundles like Article (Content type), Page (Content type) etc.
Field label	Yes	Yes	Field label of fields.
Machine name	Yes	No	Unique machine name for Drupal bundles, (use prefix + underscore: 'field + underscore' for field, for 'group + underscore' for group type fields)
X	Yes	Yes	Implementation status for bundles, a = Approved and ready to implement , w = Wait to implement, x = Implemented and done, - c = Changed since implemented, d = To be deleted
Dev	No	No	Developer initials, signifying that the row has been implemented as specified.
QA	No	No	Tester initials, signifying that the row has been validated.
Field group	Yes	Yes	Field group to which field comes
Field type	Yes	No	To define field type like Text (formatted, long), Text (plain) etc.
Ref. bundle	Yes	No	Reference of bundle, for example: Article categories (Vocabulary)
Req	Yes	Yes	Is field mandatory or not, for example: y = Yes, n = No
Vals.	Yes	Yes	Allowed number of values, Asterisk (*) means unlimited, for example: 1, 2, 3
Form widget	Yes	No	To define field Form widget, like Text area (multiple rows), Textfield
Trns.	No	No	Translatable, for example: y = Yes, n = No
Settings/notes	Yes	No	To configure settings/notes related to bundle, for example: 'Image' for media type image
Help text	Yes	Yes	Notes about the field

2.4 User roles

Defines a list of user roles in Drupal.

Table 4: User roles

Column	Creation support	Update support	Specifications
ID	No	No	An optional identifier for references in external documents.
Name	Yes	Yes	To keep name of user roles like Administrators, Content Editor, Content Manager etc.
Machine name	Yes	No	Unique machine name for Drupal bundles.
X	Yes	Yes	Implementation status for bundles, a = Approved and ready to implement , w = Wait to implement, x = Implemented and done, - c = Changed since implemented, d = To be deleted
Dev	No	No	Developer initials, signifying that the row has been implemented as specified.
QA	No	No	Tester initials, signifying that the row has been validated.
Notes	No	No	Notes about the field

2.5 Workflows

Defines a list of Drupal Workflows types.

Table 5: Workflows

Column	Creation support	Update support	Specifications
ID	No	No	An optional identifier for references in external documents.
Label	Yes	Yes	To keep name of Drupal workflow like Editorial, Administrator etc.
Machine name	Yes	No	Unique machine name for Drupal bundles.
X	Yes	Yes	Implementation status for bundles, a = Approved and ready to implement , w = Wait to implement, x = Implemented and done, - c = Changed since implemented, d = To be deleted
Dev	No	No	Developer initials, signifying that the row has been implemented as specified.
QA	No	No	Tester initials, signifying that the row has been validated.
Type	Yes	No	Workflow type Content Moderation
Notes	No	No	Notes on workflow

2.6 Workflow states

Defines a list of Workflow states for workflows.

Table 6: Workflow states

Column	Creation support	Update support	Specifications
Workflow	Yes	No	Defines name of Workflow like Editorial, Administrator etc.
Label	Yes	Yes	To keep name of Drupal workflow state like Draft, In review, Published etc.
Machine name	Yes	No	Unique machine name for Drupal bundles.
X	Yes	Yes	Implementation status for bundles, a = Approved and ready to implement , w = Wait to implement, x = Implemented and done, - c = Changed since implemented, d = To be deleted
Dev	No	No	Developer initials, signifying that the row has been implemented as specified.
QA	No	No	Tester initials, signifying that the row has been validated.
Notes	No	No	Notes on workflow states

2.7 Workflow transitions

Defines a list of Workflow transitions.

Table 7: Workflow transitions

Column	Creation support	Update support	Specifications
ID	No	No	An optional identifier for references in external documents.
Workflow	Yes	No	Name of Workflow like Editorial, Administrator etc.
Label	Yes	Yes	Name of Workflow transitions like Create New Draft, Send to review, Send to Publish etc.
Machine name	Yes	No	Unique machine name for Drupal bundles.
X	Yes	Yes	Implementation status for bundles, a = Approved and ready to implement , w = Wait to implement, x = Implemented and done, - c = Changed since implemented, d = To be deleted
From state	Yes	No	From state, example Draft, In review, Published etc.
To state	Yes	No	To state, example Draft, In review, Published etc.
Dev	No	No	Developer initials, signifying that the row has been implemented as specified.
QA	No	No	Tester initials, signifying that the row has been validated.
Notes	No	No	Notes on workflow states

2.8 Menus

Defines a list of menu types in Drupal.

Table 8: Menus

Column	Creation support	Update support	Specifications
Label	Yes	Yes	Name of Menus like Article menu, Sidebar menu etc.
Machine name	Yes	No	Unique machine name for Drupal bundles.
X	Yes	Yes	Implementation status for bundles, a = Approved and ready to implement , w = Wait to implement, x = Implemented and done, - c = Changed since implemented, d = To be deleted
Description	Yes	Yes	To keep description of menus
Dev	No	No	Developer initials, signifying that the row has been implemented as specified.
QA	No	No	Tester initials, signifying that the row has been validated.
Notes	No	No	Notes on workflow states

2.9 Image styles

Defines a list of image styles in Drupal.

Table 9: Image styles

Column	Creation support	Update support	Specifications
Style name	Yes	Yes	Name of image style name like Crop thumbnail, Thumbnail (100×100), Medium (220×220), Large (480×480) etc.
Machine name	Yes	No	Unique machine name for Drupal bundles.
X	Yes	Yes	Implementation status for bundles, a = Approved and ready to implement , w = Wait to implement, x = Implemented and done, - c = Changed since implemented, d = To be deleted
Notes	No	No	Notes on workflow states

2.10 Image effects

Defines a list of image style effects in Drupal.

Table 10: Image effects

Column	Creation support	Update support	Specifications
Image style	Yes	No	Name of image style name like Crop thumbnail, Thumbnail (100×100), Medium (220×220), Large (480×480) etc.
Effect	Yes	No	Image style effects, example- Scale, Manual crop etc.
X	Yes	No	Implementation status for bundles, a = Approved and ready to implement , w = Wait to implement, x = Implemented and done, - c = Changed since implemented, d = To be deleted
Summary	Yes	No	Summary about image style effects like width, height of image, example - width 400, uses Freeform crop type, 480×480
Notes	No	No	Notes on workflow states

INSTALLATION

Using composer is the preferred way of managing your modules and themes as composer handles dependencies automatically and there is less margin for error. You can find out more about composer and how to install it here: <https://getcomposer.org/>. It is not recommended to edit your composer.json file manually.

3.1 Download using composer

Add Github repository source in you project composer.json. Add below is snippet under `repositories` in `composer.json`:

```
"acquia-pso": {  
  "url": "https://github.com/acquia-pso/dst-entity-generate.git",  
  "type": "git"  
}
```

More information on adding repository can found at [Modifying Repositories](#)

Open up your terminal and navigate to your project root directory. Run the following command to require the module:

```
composer require acquia-pso/dst-entity-generate
```

DST Entity Generate module will install along with several module dependencies from drupal.org.

Important:

- In installation, you may need to provide a token.
 - You need to create a token in your github account.
-

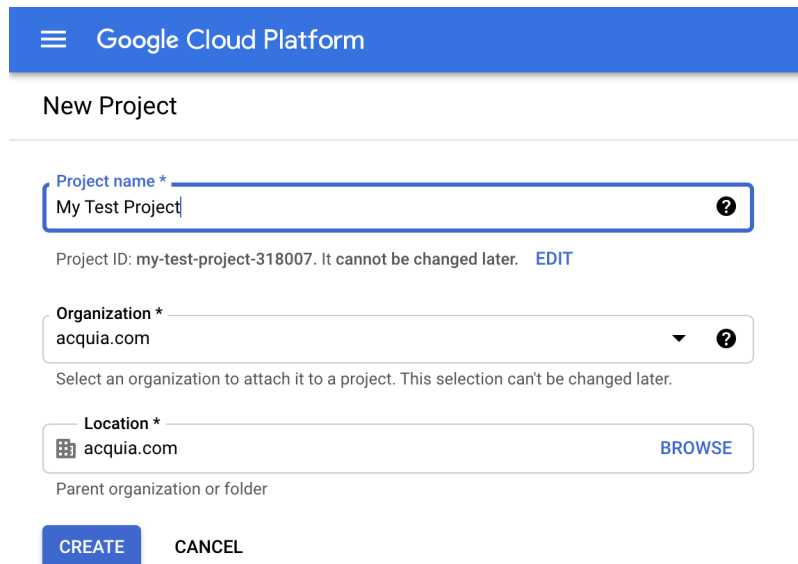
You can now enable the modules via drush with the following commands:

```
drush pm:enable dst_entity_generate -y
```

Create a Google spreadsheet for defining all drupal entities like [DEG sample template](#)

3.2 Obtaining Google OAuth 2.0 Client Credentials

- Go to Google Cloud Platform and [create new project](#)



The screenshot shows the 'New Project' form in the Google Cloud Platform console. The 'Project name' field is filled with 'My Test Project'. Below it, the 'Project ID' is shown as 'my-test-project-318007'. The 'Organization' dropdown is set to 'acquia.com'. The 'Location' dropdown is also set to 'acquia.com'. At the bottom, there are 'CREATE' and 'CANCEL' buttons.

Google Cloud Platform

New Project

Project name *
My Test Project

Project ID: my-test-project-318007. It cannot be changed later. [EDIT](#)

Organization *
acquia.com

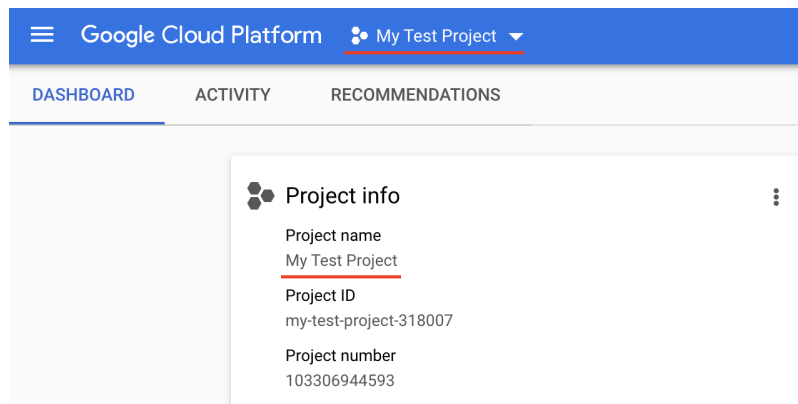
Select an organization to attach it to a project. This selection can't be changed later.

Location *
acquia.com [BROWSE](#)

Parent organization or folder

[CREATE](#) [CANCEL](#)

- After project creation, make sure your created project is selected.



The screenshot shows the Google Cloud Platform dashboard. The 'My Test Project' is selected in the top navigation bar. The 'DASHBOARD' tab is active. A 'Project info' panel is open, showing the project name 'My Test Project', project ID 'my-test-project-318007', and project number '103306944593'.

Google Cloud Platform [My Test Project](#)

[DASHBOARD](#) [ACTIVITY](#) [RECOMMENDATIONS](#)

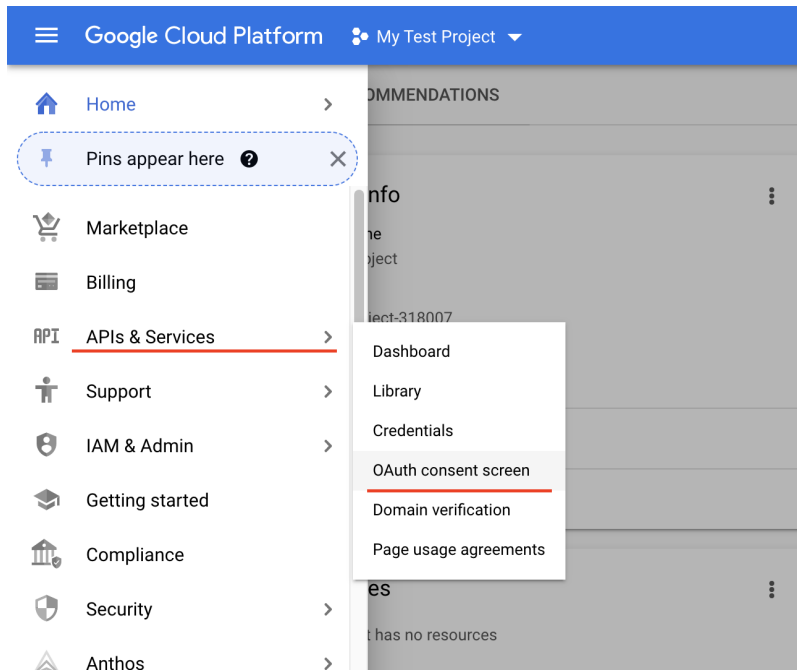
Project info

Project name
My Test Project

Project ID
my-test-project-318007

Project number
103306944593

- Then navigate to OAuth consent screen.



- Select User type to `Internal` and Click on `Create`.

APIs & Services	OAuth consent screen
<ul style="list-style-type: none"> Dashboard Library Credentials OAuth consent screen Domain verification Page usage agreements 	<p>Choose how you want to configure and register your app, including your target users. You can only associate one app with your project.</p> <p>User Type</p> <p><input checked="" type="radio"/> Internal <small>?</small></p> <p>Only available to users within your organization. You will not need to submit your app for verification. Learn more</p> <p><input type="radio"/> External <small>?</small></p> <p>Available to any test user with a Google Account. Your app will start in testing mode and will only be available to users you add to the list of test users. Once your app is ready to push to production, you may need to verify your app. Learn more</p> <p>CREATE</p>

- Provide required information on App Registration Page and click on `SAVE AND CONTINUE`.

Edit app registration

1 OAuth consent screen — 2 Scopes — 3 Summary

App information

This shows in the consent screen, and helps end users know who you are and contact you

App name *
Drupal Entity Generator

The name of the app asking for consent

User support email *
omkar.pednekar@acquia.com

For users to contact you with questions about their consent

App logo [BROWSE](#)

Upload an image, not larger than 1MB on the consent screen that will help users recognize your app. Allowed image formats are JPG, PNG, and BMP. Logos should be square and 120px by 120px for the best results.

App domain

To protect you and your users, Google only allows apps using OAuth to use Authorized Domains. The following information will be shown to your users on the consent screen.

Application home page

Provide users a link to your home page

Application privacy policy link

Provide users a link to your public privacy policy

Application terms of service link

Provide users a link to your public terms of service

Authorized domains [?](#)

When a domain is used on the consent screen or in an OAuth client's configuration, it must be pre-registered here. If your app needs to go through verification, please go to the [Google Search Console](#) to check if your domains are authorized. [Learn more](#) about the authorized domain limit.

[+ ADD DOMAIN](#)

Developer contact information

Email addresses *
omkar.pednekar@acquia.com

These email addresses are for Google to notify you about any changes to your project.

[SAVE AND CONTINUE](#)

[CANCEL](#)

- On `Scopes` step, no need to configure anything. Just click on `SAVE AND CONTINUE`.
- Now, Click on `Credentials` tab to generate OAuth Credentials.

API	APIs & Services	Edit app registration
	Dashboard	User type
	Library	Internal
	Credentials	App name
	OAuth consent screen	Drupal Entity Generator
	Domain verification	Support email
	Page usage agreements	omkar.pednekar@acquia.com
		App logo
		Not provided
		Application homepage link
		Not provided

- Click on `+ CREATE CREDENTIALS` button and select `OAuth client ID`.

API	APIs & Services	Credentials
	Dashboard	+ CREATE CREDENTIALS DELETE
	Library	<p>Create credentials to access your API</p> <p>API Keys</p> <p>Identifies your project using a simple API key to check quota and access</p> <p>OAuth client ID</p> <p>Requests user consent so your app can access the user's data</p> <p>Service account</p> <p>Enables server-to-server, app-level authentication using robot accounts</p> <p>Help me choose</p> <p>Asks a few questions to help you decide which type of credential to use</p>
	OAuth consent screen	<p>No API keys to display</p> <p>OAuth 2.0 Client IDs</p> <p>No OAuth 2.0 client IDs to display</p> <p>Service Accounts</p> <p>No service accounts to display</p>
	Domain verification	
	Page usage agreements	

- Select Application type as `Desktop app`, provide appropriate name and click on `CREATE`.

[←](#)
Create OAuth client ID

A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See [Setting up OAuth 2.0](#) for more information.

Application type *
Desktop app

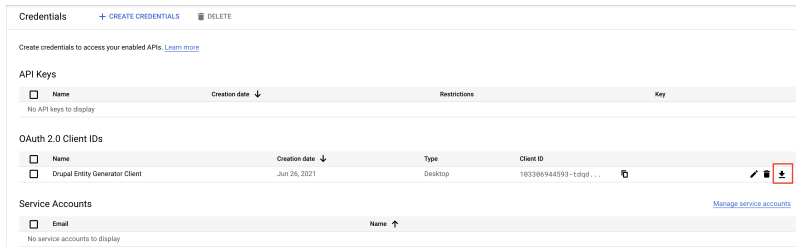
[Learn more](#) about OAuth client types

Name *
Drupal Entity Generator Client

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

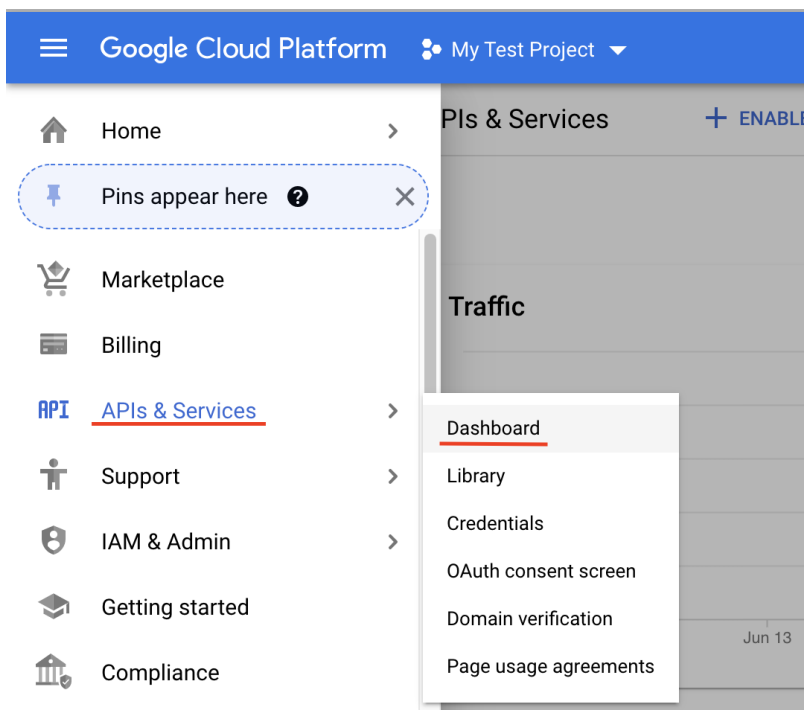
[CREATE](#)
[CANCEL](#)

- Download created credentials.

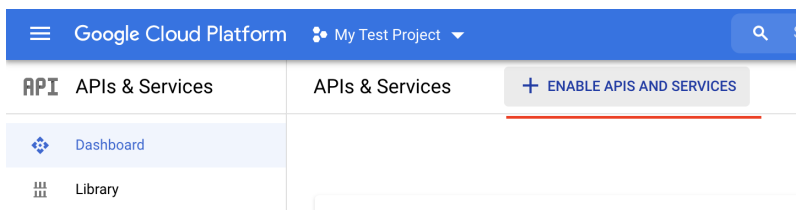


3.3 Enable Google Sheets APIS AND SERVICES

- Navigate to `APIs & Services` Dashboard.



- Click on `+ ENABLE APIS AND SERVICES` button.



- Select “Google Sheets API” and “Enable” it.

3.4 Configure DEG on Drupal

Important:

- **Private file system** should be configured before configuring DEG.
- To configure **Private file system**, refer **Private file path** section in *settings.php* file to configure it.

- Login as Administrator on site
- Go to menu Configuration -> Development -> Drupal Spec tool: Entity Generate -> Google Sheet API (/admin/config/dst_entity_generate/settings/google_sheet_api)
- Enter Application Name. This name can be anything. Not necessary it should same as we created in Google Cloud platform in previous step.
- Enter Google Sheets ID.
- Upload Client Secret file that we downloaded in earlier steps.

Home » Administration » Configuration » General

Step - 1

Application Name *

 Give Application Name like "Google Sheets API Application".

Google Sheets ID *

 Copy the spreadsheet unique id from google sheet url and paste in this field. It looks like 1xJOEelqTAC-Au02PEwPVS1zLLnwhsYaqQYPsbF8fv30.

Client Secret File *
 [Remove](#)

To get Client Secret file, follow instructions provided on [Obtaining Google OAuth 2.0 Client Credentials](#)

[Next](#)

- Click on Next.
- To complete step two, you need to valid your email account so that, it will be able to fetch data from Google Sheets.
- Click on *Click here* link.
- Select or login with Acquia email ID.
- Allow permission to *View your Google Spreadsheets*.
- Copy provided code and paste it in Step two form and submit.
- It will redirect to General Settings (/admin/config/dst_entity_generate/settings) where we can enable the entity types.

Note:

- See **Recent log messages** of Drupal if any errors occur.

DRUSH COMMANDS

A list of Drush commands used to generate the drupal entities in Drupal using Drupal Entity Generator (DEG) Spreadsheet ,

4.1 Generate All Entities

4.1.1 Description

Runs all DEG drush commands together.

4.1.2 Command

`deg:generate`

4.1.3 Aliases

`deg:generate:all`, `deg:ga`

4.1.4 Options

`--update` Update existing entity types with fields and creates new if not present.

4.2 Generate Content Types

4.2.1 Description

This command is used to generate content types with fields.

4.2.2 Command

```
deg:generate:content_types
```

4.2.3 Aliases

```
deg:content_types, deg:ct
```

4.2.4 Options

--update Update existing entity types with fields and creates new if not present.

4.3 Generate vocabularies

4.3.1 Description

This command is used to generate vocabularies with fields in Drupal.

4.3.2 Command

```
deg:generate:vocabs
```

4.3.3 Alias

```
deg:v
```

4.3.4 Options

--update Update existing Vocabulary types with fields and creates new if not present.

4.4 Generate Blocks Types

4.4.1 Description

This command is used to generate custom block types with fields in Drupal.

4.4.2 Command

```
deg:generate:custom_block_type
```

4.4.3 Alias

```
deg:cbt
```

4.4.4 Options

--update Update existing entity types with fields and creates new if not present.

4.5 Generate Image Styles

4.5.1 Description

This command is used to generate image styles and image effects in Drupal.

4.5.2 Command

```
deg:generate:imagestyle
```

4.5.3 Alias

```
deg:is
```

4.5.4 Options

--update Update existing image styles and creates new if not present.

4.6 Generate Media Types

4.6.1 Description

This command is used to generate media types with fields in Drupal.

4.6.2 Command

```
deg:generate:media
```

4.6.3 Alias

```
deg:media
```

4.6.4 Options

--update Update existing entity types with fields and creates new if not present.

4.7 Generate menus

4.7.1 Description

This command is used to generate menu types in Drupal.

4.7.2 Command

```
deg:generate:menus
```

4.7.3 Alias

```
deg:m
```

4.7.4 Options

--update Update existing Menus and creates new if not present.

4.8 Generate Paragraphs Types

4.8.1 Description

This command is used to generate paragraph types with fields in Drupal.

4.8.2 Command

`deg:generate:paragraphs`

4.8.3 Aliases

`deg:para`, `deg:p`

4.8.4 Options

`--update` Update existing entity types with fields and creates new if not present.

4.9 Generate User Roles

4.9.1 Description

This command is used to generate user roles in Drupal.

4.9.2 Command

`deg:generate:user-roles`

4.9.3 Alias

`deg:ur`

4.9.4 Options

`--update` Update existing User roles and creates new if not present.

4.10 Generate Workflows

4.10.1 Description

This command is used to generate workflows, states and workflow transitions in Drupal.

4.10.2 Command

```
deg:generate:workflow
```

4.10.3 Alias

```
deg:w
```

4.10.4 Options

--update Update existing Workflow types and creates new if not present.

Note: The following messages can come after execution of the command.

- **[OK]: means entity has been created successfully.**
Example - [OK] Node Type article is successfully created..
 - **[notice]: means field is created successfully.**
Example - [notice] Field storage created for field_teaser_title.
 - **[WARNING]: means there is some minor error comes on execution of command.**
Example - [WARNING] Alias for article is already present, skipping.
 - **[ERROR]: means there is some major error which is halting the execution of command.**
-

SUPPORTED ENTITIES

DEG tool supports basic and most required entities generation that is required in mostly all projects. Refer below list of supported entity generation.

5.1 Bundles

A list of bundles that DEG can generate.

Table 1: Bundles

Block types
Content types
Media types
Paragraph types
Vocabularies

5.2 Fields

A list for fields that DEG can generate.

Table 2: Fields

Boolean
Color
Date
Date range
Email
Link
Accordion
Accordion Item
Details
Fieldset
HTML element
Tab
Tabs
List (float)
List (integer)
Number (decimal)
Number (float)
Number (integer)
Telephone number
Entity reference
File
Image
Text (plain)
Text (plain, long)
Text (formatted)
Text (formatted, long)
Text (formatted, long, with summary)
List (text)

5.3 Form field widgets

A list of field widgets that DEG can generate.

Table 3: Form field widgets

Textfield
Text area with a summary
Text area (multiple rows)
Select list
Date and time range
Date and time
Color boxes
Color spectrum
Color HTML5
Color default
Color grid
Email
Telephone number
Autocomplete
Autocomplete (Tags style)
Image
Single on/off checkbox
Number field
Check boxes/radio buttons
File
Site Studio layout canvas
Link

5.4 Other entities

A list of Drupal entity types that DEG can generate.

Table 4: Other entities

User roles
Workflows
Workflow states
Workflow transitions
Menu
Image styles
Image effects

REFERENCES

We have used **Drupal Spec Tool** as reference for creation of DEG Google spreadsheet template.

- [Drupal Spec Tool](#)
- [Drupal.org](#)
- [drushcommands.com](#)

CONTRIBUTORS

- Mukesh Sah
- Ashutosh Gupta
- Praveen Singh
- Sharique Farooqui
- Omkar Pednekar
- Gaurav Goyal
- Amit Vyas