# Tutorial 09 Learning Objectives

Using taxonomy to structure your site

Creating content types with core and contributed fields

Organizing and aggregate record data into Views

Webforms

Continue with Custom Modules – Hooks

**Tutorial Parts Index:**

Part A - Student contact Info

Part B – Create new Drupal 8 Sites on Shared Web Host or Pantheon (PaaS)  
Part C - Extending Drupal with Taxonomy+Vocabulary+Terms  
Part D - Extending Drupal 8 with Custom Content Types  
Part E - Extending Drupal 8 with Views  
Part F - Extending Drupal 8 with Webforms

Part G - Drupal 8 Custom Modules Continued – Introduction to Hooks (Hook API)  
Part H – Drupal 8 JavaScript Library Install

Part A – Student information

1. Download and save a copy of the Tutorial 09 Template document
2. Rename the template document   
   **[Your-Name or Initials]-**Tutorial-09-Template.docx

Go to the template and enter student information for Part A.

Part B – Building Drupal 8 site on shared web host or Pantheon

Build Drupal 8 sites on your preferred shared web host environment.

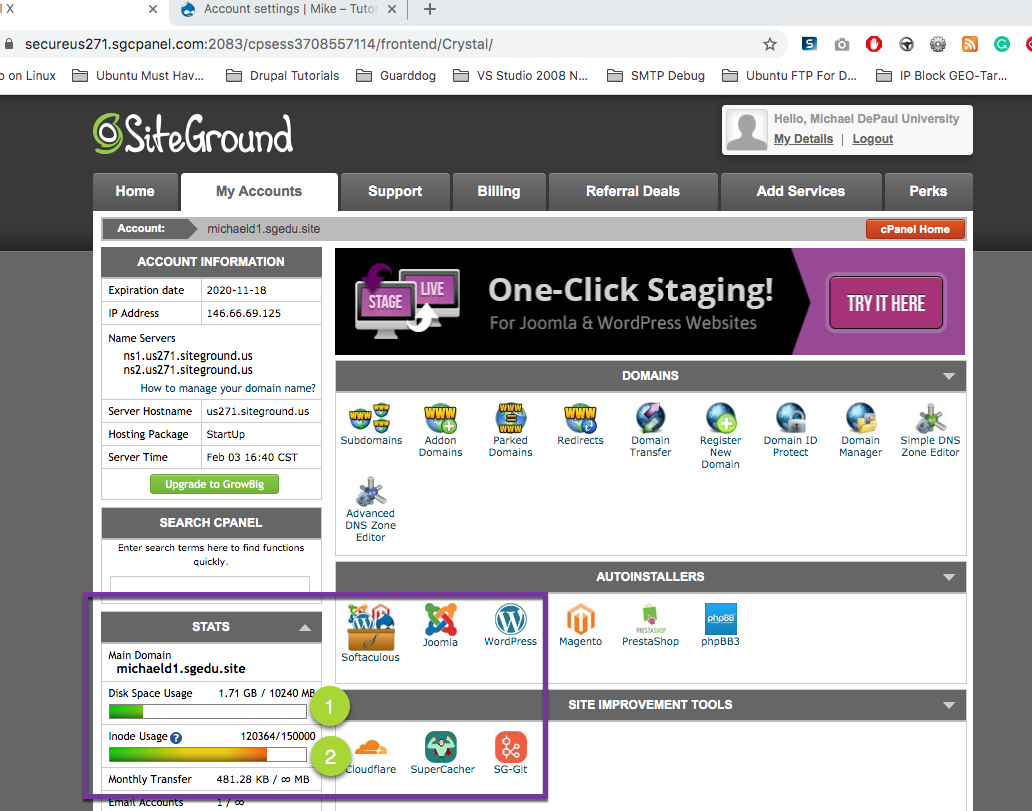
1. Building Drupal 8 sites on shared web hosts with one click cPanel install
   1. **Version:** Select the default (most recent Drupal 8 version)
   2. **In Directory:** tutorial09
   3. CRON Job: Keep default values (do not change)
   4. Site Settings:
      1. Site Name –[Your First Name] – Tutorial 09
   5. Admin Account
      1. **Admin Username:** <create your own username>
      2. **Admin Password:** <enter a string password & write it down>
      3. **Admin Email**: Use the default email
   6. Advanced Options
      1. **Database Name:** keep default – write it down
      2. **Table Prefix:** keep default – write it down
      3. **Backup Location:** keep default
   7. **After creating the D8 site, harden it.**
2. Building Drupal 8 sites on Pantheon (PaaS) – For those with Pantheon accounts
   1. **Create Your Pantheon Site:**
      1. **Name Your Pantheon Site:** [Your Initials] - Tutorial 09 PA
      2. **Choose an Organization:** None
      3. **Choose a Region for the Site:** United States
   2. Choose Your CMS:
      1. Select Deploy for Drupal 8
   3. Drupal 8 Site Configuration settings:
      1. Choose language: English
      2. Select an installation Profile: Standard
      3. Configure Site:
         1. Site Name: <Your Initials> - Tutorial 09 – PA
         2. Site Email address: <your Pantheon account email>
         3. Username: enter a username
         4. Password: enter a string password – write it down
         5. Email address: use your Pantheon account email
         6. REGION Settings:
            1. Default Country: United States
            2. Default Time Zone: Chicago
            3. UPDATE Notification:

Check for updates automatically – CHECKED

Receive email notification: CHECKED

* 1. After building the site, harden it.

1. Clean out old WordPress sites on SiteGround, if you are getting low on free space.



Go to the Tutorial 09 Template and take and record the requested screen captures and site login credentials for your Drupal 8 CMS sites for Part B.

Part C – Extending Drupal with Taxonomy+Vocabulary+Terms

Extending Drupal 8 with Taxonomy + Vocabulary + Terms

Taxonomy, a **powerful** Drupal core module, and gives our Drupal sites use of the organizational keywords “**terms**”.

**Terms** are known in other systems as **categories**, **tags**, or **metadata**.

Taxonomy allows you to **connect, relate and classify your website’s content**.

In Drupal, these “**terms”** are gathered within "***vocabularies***."

The Taxonomy module allows you to create, manage and apply those vocabularies.”

Note about adding Terms to Taxonomy-Vocabularies:   
Drupal knows that site developers usually always add more than one Term to a vocabulary. The add vocabulary page returns to itself when you submit.

To end the add process, select the Taxonomy bread-crumb (see below).

|  |
| --- |
| Structure > Taxonomy |
| Select “+Add vocabulary” |
| HD:private:var:folders:vh:lbh1qypx7zldldk1l9f6tpsc0000gn:T:2017-03-28_15-51-16.png |

1. Create Vocabulary: Direction with 4 terms: North, South, East, West
2. Create Vocabulary: Season with 4 terms: Winter, Spring, Summer, Fall
3. Create Vocabulary: Eye color with 3 terms Blue, Green, Brown
4. Create Vocabulary: Box size with 3 terms Small, Medium, Large

## Go to the Tutorial 08 Template and take the requested screen captures for Part C.

Part D - Extending Drupal 8 with Custom Content Types

*Drupal allows us to extend the content types by creating new one. The content types are templates used for data entry. We have already used Basic page and Article content type templates to add data to create content nodes.   
In this exercise we will create two new content type templates  
(1) Employee*

*(2) Product*

***We do not want any of our new content type records to display on the front page or on a menu item.****Instead we display all of our records (nodes) to be aggregated and displayed on one page using an aggregate view all each record type. In other words, we will display only those records of a specific type.*

***Record Template Field Structure***

*Before you can create a content type template structure, you have to know what type of fields are on the structure, there field names, whether or not they are required, and their order on the template, and information about their field tag.  
The best way to design a template is to analyze its content.*

|  |
| --- |
| *Employee Content Record Example* |
| *HD:private:var:folders:vh:lbh1qypx7zldldk1l9f6tpsc0000gn:T:2020-02-03_18-24-36.png* |
| *The above employee example record indicates that there are at least six (6) fields in the following order: (1) Name – the employees name*  *(2) Description – A long text field that goes in its own Description field – not the Body (3) Job Title – the title of the job - a text field*  *(4) Phone – A telephone field 🡨 Drupal 8 has one in Core – that’s disabled by default (5) Eye color – We have a Taxonomy-Vocabulary-Term list of these*  *(6) Picture – an image field* |

|  |
| --- |
| *The trick with record analysis is to review many record examples to make they all have the same fields of the same field data type. Always be on the lookout for that one record that has an extra field that none of the other records has. This extra field has to be added to the template.  If it doesn’t exist, you won’t be able to enter it into the record.* |

Repeat the same process for Product.

|  |
| --- |
| *Product Record Example* |
| *HD:private:var:folders:vh:lbh1qypx7zldldk1l9f6tpsc0000gn:T:2020-02-04_09-00-13.png* |
| *The above product example record indicates that there are at least five (5) fields in the following order: (1) Name – the product name*  *(2) Description – A long text field  (3) Part Number – a text field*  *(4) Cost – A currency field (5) Box size – a reference to a taxonomy-vocabulary named Box size*  *(6) Thumbnail Image – A image field formatted/sized as Thumbnail* |

Go to the Tutorial 09 Template and take the requested screen captures and or answer the questions for Part D.

Part E– Extending Drupal 8 with Views

Drupal is a CMS. As a CMS Drupal stores all site content in a database.   
A view is a database term, for a type data aggregation.  
Drupal’s use of the term View is 100% correct.   
A Drupal view is an aggregation of content from the sites database. Drupal provides an Administrative User interface to create views. What this Admin UI does, is allow a site admin to configure the view, so that Drupal can create the correct database SQL statement to query and retrieve the correct content and its formatting options, as configured by the admin.  
  
We will create two views.   
Our first view will aggregate and display “Employee” record nodes.  
Our second view will aggregate and display “Product” record nodes.

|  |
| --- |
| Create the View Structure > Views > Add new view |
| C:\Users\mchase2\AppData\Local\Temp\s1.png |

Go to the Tutorial 09 Template and take the requested screen captures and or answer the questions for Part E.

Part F - Extending Drupal through Webforms

Before starting this part, please review the presentation

M09 -01- How to build webforms on Drupal 8.

Webform is the module for making forms and surveys in Drupal. After a submission customizable e-mails can be sent to administrators and/or submitters. Results can be exported into Excel or other spreadsheet applications. Webform also provides some basic statistical review and has an extensive API for expanding its features.

We will use the webform module to make a Contact us form.

Drupal Webform Module Project page:

<https://www.drupal.org/project/webform>

Go to the Tutorial 09 Template and take the requested screen captures and or answer the questions for Part F.

Part G– Drupal 8 Custom Modules Continued - Hooks

For Part G – Use either an existing Drupal 8 site on your Codeanywhere platform or build a new Drupal 8 site on your Codeanywhere account.

Please review the Module 09 presentation -   
M09 - 02 - Building Drupal 8 CMS Modules and Adding Hooks.

1. Review the presentation - M09 - 02 - Building Drupal 8 CMS Modules and Adding Hooks. and follow along with the videos as we introduce implementing hooks on Drupal 8 custom modules on all platforms.
2. Implement a Custom Module on SiteGround
3. Implement a Custom Module on Codeanywhere
4. Implement a Custom Module on Pantheon (PaaS)
   1. For those with Pantheon accounts.

**About Hooks:**

Hooks define functions that alter the behavior of Drupal core.

Hooks are a way modules can alter the core behavior of Drupal (or another module.

Hooks are specially-named functions that a module defines (this is known as "implementing the hook"), which are discovered and called at specific times to alter or add to the base behavior or data (this is known as "invoking the hook").

Each hook has a name (example: hook\_batch\_alter()), a defined set of parameters, and a defined return value.

Your modules can implement hooks that are defined by Drupal core or other modules that they interact with.

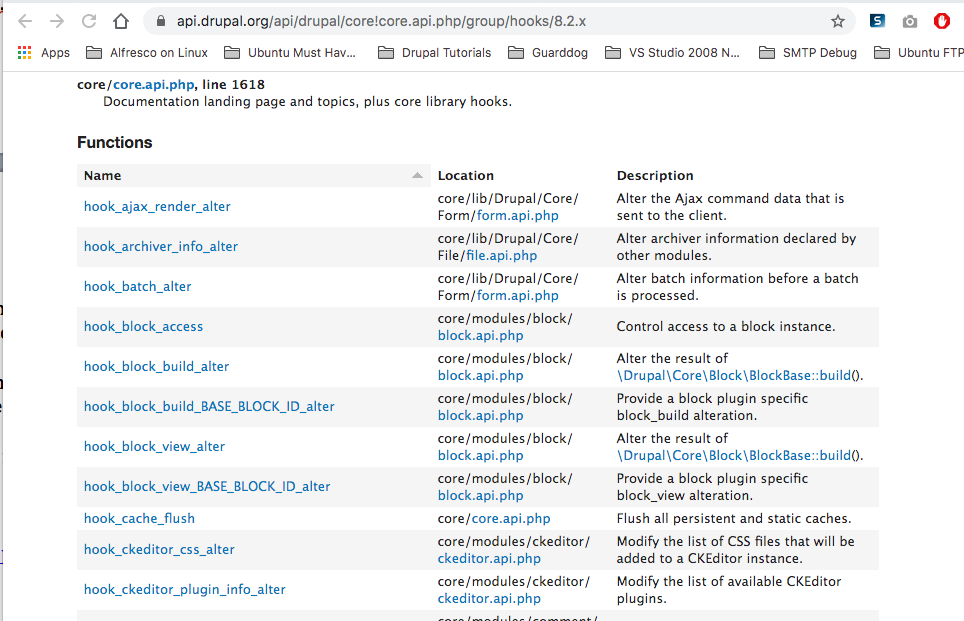
Your modules can also define their own hooks, in order to let other modules interact with them.

To implement a hook:

* Locate the documentation for the hook. Hooks are documented in \*.api.php files, by defining functions whose name starts with "hook\_" (these files and their functions are never loaded by Drupal -- they exist solely for documentation). The function should have a documentation header, as well as a sample function body. For example, in the core file system.api.php, you can find hooks such as hook\_batch\_alter(). Also, if you are viewing this documentation on an API reference site, the Core hooks will be listed in this topic.
* Copy the function to your module's .module file.
* Change the name of the function, substituting your module's short name (name of the module's directory, and .info.yml file without the extension) for the "hook" part of the sample function name. For instance, to implement hook\_batch\_alter(), you would rename it to my\_module\_batch\_alter().
* Edit the documentation for the function (normally, your implementation should just have one line saying "Implements hook\_batch\_alter().").
* Edit the body of the function, substituting in what you need your module to do.

Hooks List:

<https://api.drupal.org/api/drupal/core!core.api.php/group/hooks/8.2.x>



In this part of the tutorial, we will walk through hook implementation by adding two hook functions (HOOK HELP and HOOK CRON) into a Drupal 8 custom module.

The code for these two hook implementations are included in the tutorial assets zip file.

The assets zip file contains the Drupal 8 custom module named “Hello World”.

This is the same Drupal 8 custom plugin with two new files added to it.   
  
The two additional files are a .module file and a permisisons.yml file:

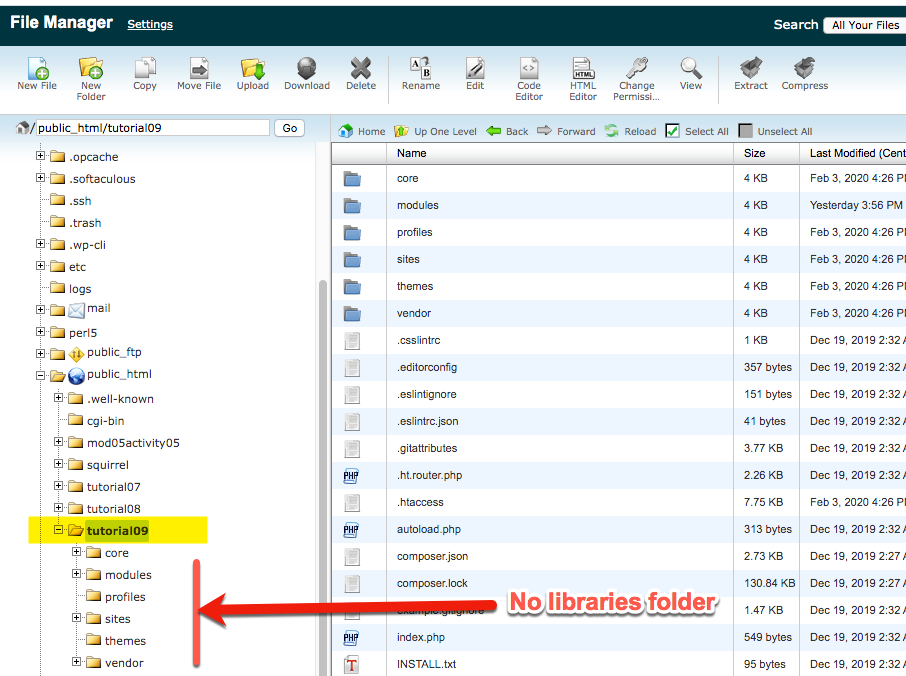
1. hello\_world.module – with Hook Help and Hook CRON implementations
2. hello\_world.permissions.yml – to demonstrate adding permissions to a module

**Follow along with the video tutorial** as we add this custom module to our Codeanywhere Drupal 8 site and then review Drupal 8 Hook API fundamentals and add the permission YML file.

Go to the Tutorial 09 Template and take the requested screen captures and or answer the questions for Part G.

Part H– Drupal 8 Java Script Libraries

Drupal 8 can be extended using JavaScript libraries.   
The only issue is that the Drupal 8 installation build script does not create a libraries folder during the installation process.  
In order for us to install JavaScript (JS) Libraries we must create the folder manually.



Please access and login to your shared web host (or Pantheon) and Codeanywhere Drupal 8 sites as we learn how to install a JS Library on Drupal 8.

The JavaScript library we will install is named WooCommerce/FlexSlider.  
It can be downloaded from GitHub at the following URL:  
<https://github.com/woocommerce/FlexSlider>

1. SiteGround
2. Pantheon (Paas)
3. Codeanyhwere

Rules for creating the JS libraries folder:

1. The folder must be named libraries (all lower case)
2. The folder must be created directly below the Drupal 8 root folder
3. On most shared web hosts and cloud dev platforms, you can upload zip or tar.gz compress JS Library files and expand them in place within the libraries folder
4. Most JS libraries expanded folders must be renamed. Refer to the JS library readme and installation instructions for renaming and any additional folder or file manipulation or edits.

Additional Rules for installing the JS Libraries on Pantheon:

1. Pantheon does not have a cPanel, and the FileZilla can be used to create and copy up files to the newly created libraries folder.
2. Do not copy zip or tar.gz files to your Pantheon libraries folder. You will be unable to expand them on the Pantheon file system.
3. Pantheon prohibits us from installing scripts that would allow us to expand compressed files on the Pantheon file system.
4. We must expand zip files on our local computers and use FileZilla to copy up the JS libraries expanded folders and files.

After installing the JS Library on you 2 platforms, go to the Tutorial 09 Template and take the requested screen captures and or answer the questions for Part H