

Chapter 11:

Extending and

Customizing Your

Site

- [Finding Modules](#)
- [Enabling and Disabling Maintenance Mode](#)
- [Downloading and Installing a Module from Drupal.org](#)
- [Finding Themes](#)
- [Downloading and Installing a Theme from Drupal.org](#)
- [Manually Installing Module or Theme Files](#)
- [Concept: Development Sites](#)
- [Making a Development Site](#)
- [Deploying New Site Features](#)
- [Synchronizing Configuration Versions](#)
- [Managing File and Configuration Revisions with Git](#)

Finding Modules

Goal

Find and evaluate modules on *Drupal.org*.

Prerequisite knowledge

- [Concept: Drupal as a Content Management System](#)
- [Concept: Modules](#)

Steps

1. Go to [Drupal.org](https://www.drupal.org), and navigate to *Download & Extend > Modules* (https://www.drupal.org/project/project_module).
2. Filter your search using the categories on the module search page. Fill in the fields as shown below.

Field name	Explanation	Example value
Maintenance status	How actively should the module be maintained?	Actively maintained
Development status	What kind of development should the module undergo?	Any
Module categories	The module's topic area.	Administration
Core compatibility	The version of the core software the module is compatible with.	9.x

Field name	Explanation	Example value
Status	Project status: <code>_Sandbox projects_</code> are experimental projects. <code>_Full projects_</code> have already gone through an approval process, but they can still be in development.	Full projects

S
t
a
b
i
l
i
t
y

Field name	Explanation	Example value
Has a supported stable release	Security advisory coverage	Whether or not the project maintainer has agreed to follow Drupal
Search modules	Search for <code>_Admin Toolbar_</code> , a module that will be covered in detail later. Alternatively, you can also leave the field blank if you are not sure which module to search for.	Admin Toolbar
Sort by	Order your search results by criteria like <code>_Most installed_</code> (popular modules that many sites use) or <code>_Last release_</code> (date of latest version released).	Most installed

Module categories - Any -

Core compatibility 8.x

Status Full projects

Stability All projects

Security advisory coverage All projects

Search modules

Sort by Most installed

[Search](#)

3. Click *Search*. Search results will appear.

Admin Toolbar

Posted by [eme](#) on 20 April 2015, updated 2 October 2017

What is Admin Toolbar module ?

Admin Toolbar intends to improve the default Drupal Toolbar (the administration menu at the top of your site) to transform it into a drop-down menu, providing a fast access to all administration pages.

The module works on the top of the default toolbar core module and is therefore a very light module and keeps all the toolbar functionalities (shortcut / media responsive).

How to use Admin Toolbar module ?

Just install it like any other module.

[Read more](#) · Categories: [Actively maintained](#) , [Under active development](#) , [Administration](#)

Adminimal Admin Toolbar

Posted by [energee](#) on 19 March 2016, updated 25 July 2017

This module is inspired by [Adminimal Administration Menu](#). It provides a minimalist style to [Admin Toolbar](#) for Drupal 8.

Although the "Adminimal" theme provides menu styling, there are some conflicts with Admin toolbar and styling will only be present when the admin theme is set like when editing content.

[Read more](#) · Categories: [Actively maintained](#) , [Under active development](#) , [Administration](#) , [Theme Enhancements](#)

D8 Editor File upload

Posted by [DuaelFr](#) on 5 September 2015, updated 21 March 2017

This module allows to add a button in the Drupal 8 rich text editor active toolbar to directly upload and link files into your content. Without this module, the writer would have to upload the files on the webserver via a file field or a FTP connexion then manually create the link.

[Read more](#) · Categories: [Actively maintained](#) , [Under active development](#) , [Content](#)

CKEditor Media Embed Plugin

Posted by [grndlvi](#) on 20 October 2015, updated 12 October 2017

4. To further evaluate a module, click its title in the list of search results to visit its project page.

Some aspects to pay attention to when evaluating modules:

- Project description: The description of the module on its project page should be clear and useful. You should get an idea of its features and requirements.

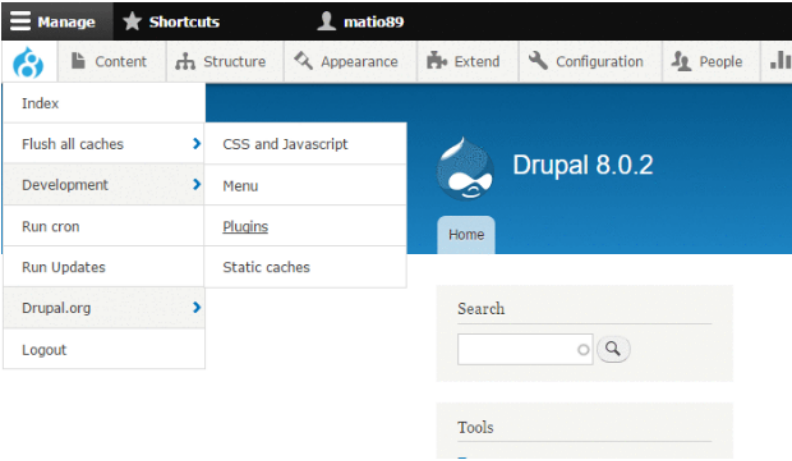
- Project information: There may be warnings in this area of the page, such as if a module is no longer being developed, or is not covered by the security advisory policy.
- Project information > Reported installs, downloads: You can see how many people have downloaded and how many sites use the module. If it's only used by a few sites, it might be a unique solution that not many people need, or it might be a warning sign that you shouldn't use it either.
- Maintainers: When was the last commit (the last time someone updated something on the module) or last release (new version)? If the project has few open issues, a long time since commits/releases might be appropriate, but if it has a lot of open issues and there are no commits/releases, that would be a clue that it might be abandoned.
- Issues: See if there are any open issues, potential problems with the module. Check the Statistics to see how regularly issues are responded to.
- Documentation, Resources: Check if the module has documentation or a README file, that can help you install, configure, explore, and test it.

Admin Toolbar

[View](#) [Version control](#) [Automated testing](#)

Posted by [eme](#) on 20 April 2015, updated 20 November 2017

★ 131



Maintainers for Admin Toolbar

[adriancid](#) - 47 commits
last: 1 month ago, first: 1 year ago

[romainj](#) - 49 commits
last: 7 months ago, first: 2 years ago

[eme](#) - 11 commits
last: 2 years ago, first: 2 years ago

[matio89](#) - 23 commits
last: 2 years ago, first: 3 years ago

[View all committers](#)
[View commits](#)

Issues for Admin Toolbar

To avoid duplicates, please search before submitting a new issue.

[Search](#)

[Advanced search](#)

[All issues](#)

What is Admin Toolbar module?

Admin Toolbar intends to improve the default Drupal Toolbar (the administration menu at the top of your site) to transform it into a drop-down menu, providing a fast access to all administration pages.

Expand your understanding

[Downloading and Installing a Module from *Drupal.org*](#)

Videos

[Finding Modules](#)

Enabling and Disabling Maintenance Mode

Goal

Put your site in maintenance mode to allow users with the right permissions to use the site while users without this permission are presented with a message that the site is under maintenance.

Prerequisite knowledge

[Concept: Security and Regular Updates](#)

Site prerequisites

If you want to use Drush to enable or disable maintenance mode, Drush must be installed. See [Concept: Additional Tools](#).

Steps

You can use the administrative interface or Drush to enable or disable maintenance mode.

Enabling maintenance mode using the administrative interface

1. In the *Manage* administrative menu, navigate to *Configuration > Development > Maintenance mode (admin/config/development/maintenance)*. The *Maintenance mode* page appears.

2. Fill in the fields as shown below.

Field name	Explanation	Value
Put site into maintenance mode	Enable the maintenance mode	
Message to display when in maintenance mode	The information that is shown to website visitors when the mode is enabled. Variables such as @site can be used in the message	@site is currently under maintenance but should be back shortly. Thank you for your patience.

3. Click *Save configuration*.

4. Verify that the site is in maintenance mode by accessing it from another browser where you aren't logged in. If you are not able to verify, try clearing the cache. See [Clearing the Cache](#).

Anytown Farmers Market

Site under maintenance

Anytown Farmers Market is currently under maintenance. We should be back shortly. Thank you for your patience.

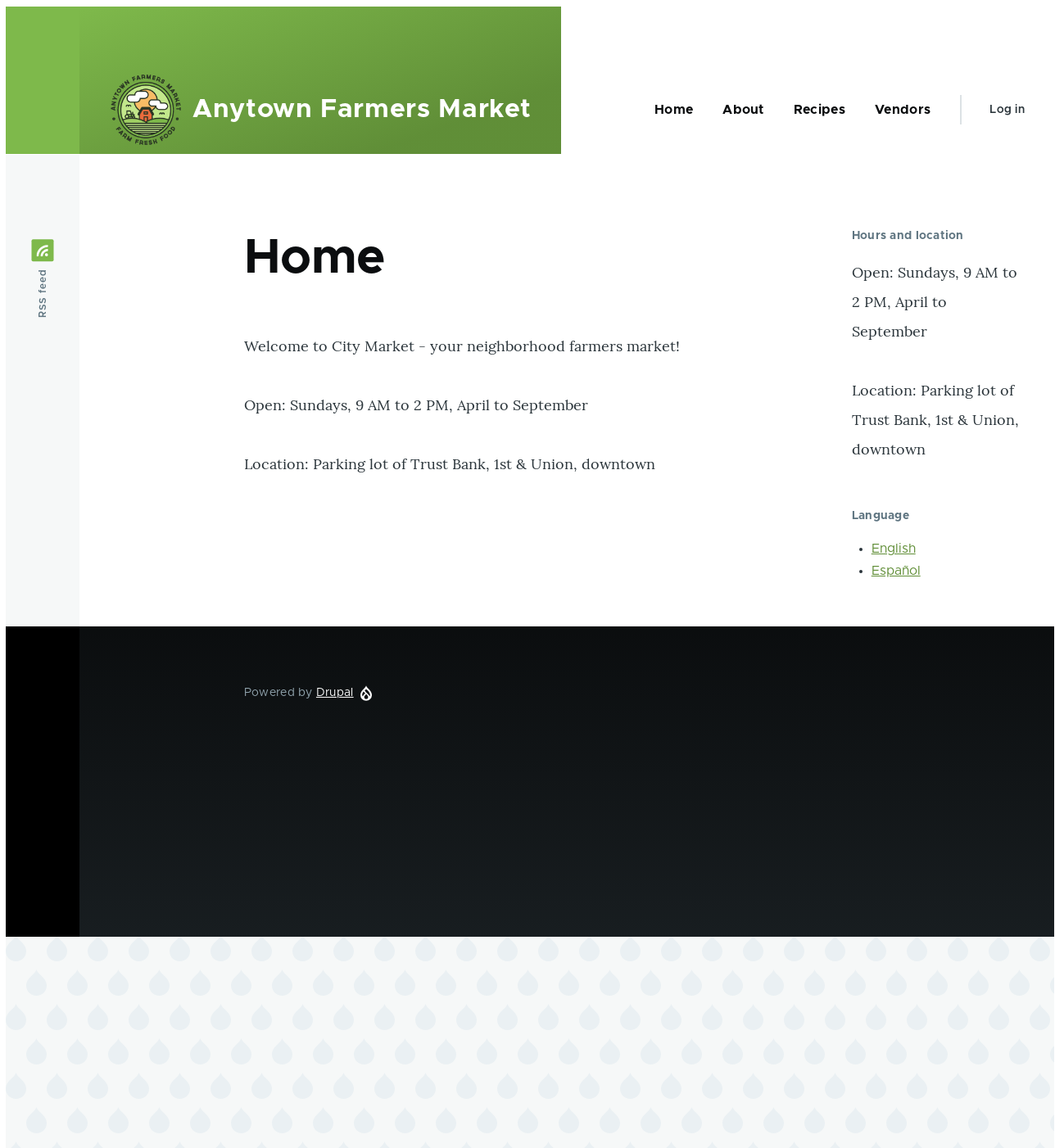


Disabling maintenance mode using the administrative interface

1. In the *Manage* administrative menu, navigate to *Configuration > Development > Maintenance mode* (*admin/config/development/maintenance*). The *Maintenance mode* page appears.
2. Fill in the fields as shown below.

Field name	Explanation	Value
Put site into maintenance mode	Disable the maintenance mode	
Message to display when in maintenance mode	No message required while disabling. You can leave the field blank.	

3. Click *Save configuration*.
4. Verify that the site is no longer in maintenance mode by accessing it from another browser where you aren't logged in. If you are not able to verify, try clearing the cache. See [Clearing the Cache](#).



Enabling or disabling maintenance mode using Drush

1. Follow the user interface steps above to edit the site maintenance message, if desired.
2. Run the following Drush commands to enable maintenance mode and clear the cache:

```
drush config:set system.maintenance message "Optional message" -y
drush state:set system.maintenance_mode 1 --input-format=integer
```

```
drush cache:rebuild
```

3. Run the following Drush commands to disable maintenance mode and clear the cache:

```
drush state:set system.maintenance_mode 0 --input-format=integer  
drush cache:rebuild
```

4. After running either set of commands, verify that your site is either in or out of maintenance mode by visiting the site in a browser where you are not logged in.

Expand your understanding

- [Updating the Core Software](#)
- [Updating a Theme](#)
- [Updating a Module](#)

Videos

[Enabling and Disabling Maintenance Mode](#)

Downloading and Installing a Module from [_Drupal.org_](https://drupal.org)

Goal

Download and install the [contributed Admin Toolbar module](#), which allows you to easily browse through the administration section of the website.

Prerequisite knowledge

- [Concept: Modules](#)
- [Finding Modules](#)
- [Concept: Additional Tools](#)

Site prerequisites

Composer must be installed to download modules. If you want to use Drush, Drush must be installed. See [Concept: Additional Tools](#).

Steps

To install a contributed module, first download the module with Composer. Then install it using either the administrative interface or Drush. If you are installing a custom module rather than a contributed module that is not available via Composer, skip the steps for downloading the module, and refer to [Manually Installing Module or Theme Files](#). Then return here and follow the steps for installing the module using either the administrative interface or Drush.

Download the contributed module with Composer

1. On the *Admin toolbar* project page on drupal.org (https://www.drupal.org/project/admin_toolbar), scroll to the *Releases* section at the bottom of the page.
2. Copy the provided Composer command for the version of the module you want to install.

Releases

3.5.0  released 2 August 2024
Works with Drupal: ^9.5 || ^10 || ^11
✓ Recommended by the project's maintainer.

Install: `composer require 'drupal/admin_toolbar:^3.5'` 

Development version: [3.x-dev](#) updated 2 Aug 2024 at 22:18 UTC

GitLab CI:  [view all pipelines](#)

3. Alternatively, type the following command (substituting the short name of the module and desired version for `admin_toolbar:^3.5`):

```
composer require 'drupal/admin_toolbar:^3.5'
```

4. At the command line, change to the root directory of your project. Paste the Composer command and execute it.
5. You should see a message about the module being successfully downloaded.

Install the module using the administrative interface

1. In the *Manage* administrative menu, navigate to *Extend* (*admin/modules*). The *Extend* page appears.
2. Locate the *Admin toolbar* module and check it.

Extend

[List](#)[Update](#)[Uninstall](#)

Add [contributed modules](#) to extend your site's functionality.

Regularly review [available updates](#) and update as required to maintain a secure and current site. Always run the [update script](#) each time you update software.

Filter

▼ **Core**

^ **Administration**

+ Show all columns

**Admin Toolbar****Admin Toolbar Extra Tools****Admin Toolbar Links Access Filter ([Deprecated](#))****Admin Toolbar Search**

▼ **Core (Experimental)**

▼ **Field types**

▼ **Migration**

3. Click *Install* to turn on the new module.

Install the module using Drush

1. Run the following Drush command, giving the project name (for example, `admin_toolbar`) as a parameter:

```
drush pm:enable admin_toolbar
```

2. Follow the instructions on the screen.

Expand your understanding

- Verify that the [contributed Admin Toolbar module](#) is working by browsing through the menu in the administration section.
- Install and configure the [contributed Pathauto module](#) so that content pages in your site get nice URLs by default. See [Concept: Paths, Aliases, and URLs](#) for more on URLs.
- If you do not see the effect of these changes in your site, you might need to clear the cache. See [Clearing the Cache](#).

Videos

[Downloading and Installing a Module from Drupal.org](#)

Additional resources

- [Drupal.org community documentation page "Installing Drupal modules"](#)
- ["Download and Extend" page on Drupal.org](#)
- [Admin Toolbar module on Drupal.org](#)

Finding Themes

Goal

Find and evaluate themes on *Drupal.org*.

Prerequisite knowledge

- [Concept: Drupal as a Content Management System](#)
- [Concept: Themes](#)

Steps

1. Go to <https://www.drupal.org>, and navigate to *Download & Extend > Themes* (https://www.drupal.org/project/project_theme).
2. Filter your search using the categories on the theme search page. For example, you might use these filters:

Field name	Explanation	Example value
Maintenance status	How actively is the theme maintained? If a theme is actively maintained, you can expect bug fixes and improvements on a regular basis.	Actively maintained

Field name	Explanation	Example value
Development status	What kind of development is the theme undergoing? If you select <code>_Under active development_</code> , you can expect new features to be added, and some aspects may still change. If you select <code>_Maintenance fixes only_</code> , it means that the theme is considered complete.	Any

C
o
r
e
c
o
m
p
a
t
i
b
i
l
i
t
y

Field name	Explanation	Example value
Status The e t h e r o r n o t t h e p r o j e c t m a i n t a i n e r h a s s c r e a t e d a p r o d u c t i o n - r e a d y	_Sandbox projects_ are experimental projects. _Full projects_ have already gone through an approval process, but they can still be in development.	Full projects

S
t
a
b
i
l
i
t
y

Field name	Explanation	Example value
Security advisory coverage e a r c h b y s e a r c h t e r m i n t h e t h e m e , s d e s c r i p t i o n .	Whether or not the project maintainer has agreed to follow Drupal Security Team procedures.	Has security advisory coverage
Sort by	Order your search results by criteria like <u>Most installed</u> (popular themes that many sites use) or <u>Last release</u> (date of latest version released).	Most installed

Search the themes

309 themes match your search

Maintenance status Actively maintained ▼

Development status - Any - ▼

Core compatibility 8.x ▼

Status Full projects ▼

Stability All projects ▼

Security advisory coverage All projects ▼

Search themes

Sort by Most installed ▼

Search

3. Click *Search*. Search results will appear.

Bootstrap

Posted by [wundo](#) on 18 May 2008, updated 15 December 2016

“ Sleek, intuitive, and powerful mobile first front-end framework for faster and easier web development. Bootstrap has become one of the most popular front-end frameworks and open source projects in the world.

This base theme bridges the gap between Drupal and the [Bootstrap Framework](#).

Features

- [jsDelivr CDN](#) for "out-of-the-box" styling and faster page load times.
- [Bootswatch](#) theme support, if using the CDN.
- Glyphicons support via [Icon API](#).
- Extensive integration and template/preprocessor overrides for most of the [Bootstrap Framework](#) CSS, Components and JavaScript
- Theme settings to further enhance the Drupal Bootstrap integration:
 - [Breadcrumbs](#)
 - [Navbar](#)
 - [Popovers](#)
 - [Tooltips](#)
 - [Wells](#) (per region)

Documentation

Visit the project's [official documentation site](#) or the markdown files inside the `./docs` directory.

Supported modules

Drupal 8

- [Bootstrap Layouts](#)

Drupal 7

- [Bootstrap Core](#)
- [jQuery Update](#)
- [Icon API](#)
- [Picture](#)
- [Views](#) (*partial support*)
- [Webform](#) (*partial support*)

5 Year Evolution (gource)

<https://youtu.be/Cvq6MPJp2dI>

4. To further evaluate a theme, click its title in the list of search results to visit its project page.

Some aspects to pay attention to while evaluating themes:

- Introduction: The description of the theme on its project page should be clear and useful. A screenshot of the theme helps your evaluation as well.

- Project information: There may be warnings in this area of the page, such as if a theme is no longer being developed, or is not covered by the security advisory policy.
- Project information > Reported installs, downloads: You can see how many people have downloaded and how many sites use the theme.
- Issues: See if there are any open issues, potential problems with the theme. Check the Statistics to see how regularly issues are responded to.
- Documentation, Resources: Check if the theme has documentation or a README file, that can help you install, configure, explore, and test it.

Expand your understanding

- [Downloading and Installing a Theme from *Drupal.org*](#)

Videos

[Finding Themes](#)

Downloading and Installing a Theme from `_Drupal.org_`

Goal

Download and install a theme from *Drupal.org*.

Prerequisite knowledge

- [Finding Themes](#)
- [Concept: Additional Tools](#)

Site prerequisites

Composer must be installed to download themes. If you want to use Drush, Drush must be installed. See [Concept: Additional Tools](#).

Steps

To install a contributed theme, first download the theme with Composer. Then install it using either the administrative interface or Drush. If you are installing a custom theme rather than a contributed theme that is not available via Composer, skip the steps for downloading the theme, and refer to [Manually Installing Module or Theme Files](#). Then return here and follow the steps for installing the theme using either the administrative interface or Drush.

Download the contributed theme with Composer

1. On the theme's project page on drupal.org (for example, <https://www.drupal.org/project/honey>), scroll to the *Releases* section at the bottom of the page.
2. Copy the provided Composer command for the version of the theme you want to install.

Releases

1.0.2  released 17 January 2023

Works with Drupal: ^9 || ^10

~~(Recommended by the project's maintainers.)~~

Install: `composer require 'drupal/honey:^1.0'`



Development version: **1.0.x-dev** updated 17 Jan 2023 at 12:33 UTC

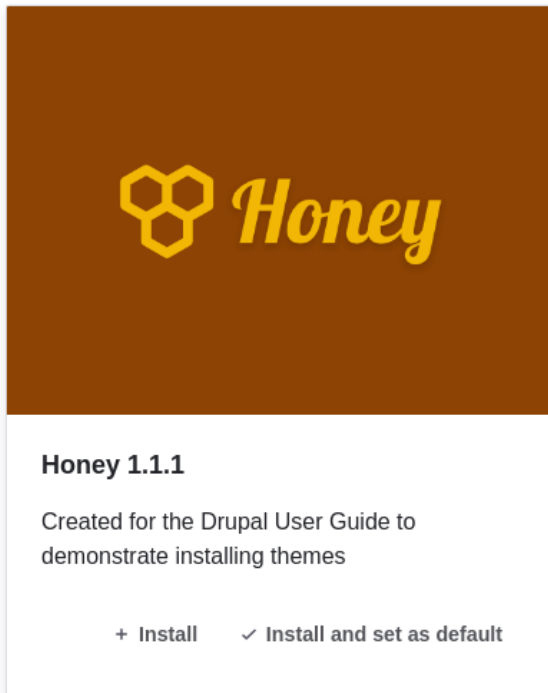
3. Alternatively, type the following command (substituting the short name of the theme and desired version for `honey:^1.0`):

```
composer require 'drupal/honey:^1.0'
```

4. At the command line, change to the root directory of your project. Paste the Composer command and execute it.
5. You should see a message about the theme being successfully downloaded.

Install the theme using the administrative interface

1. In the *Manage* administrative menu, navigate to *Appearance* (<admin/appearance>). The *Appearance* page appears.
2. Locate the new theme under *Uninstalled themes* and click *Install and set as default* to use it. All non-administrative pages on the site will now use this new theme.



Install the theme using Drush

1. To install the theme, and set it as the default, run the following Drush commands, giving the project name (for example, *honey*) as a parameter:

```
drush theme:enable honey  
drush config:set system.theme default honey
```

2. Follow the instructions on the screen.

Expand your understanding

- In the *Manage* administrative menu, navigate to *Appearance* (*admin/appearance*) and uninstall any themes that you are not using.
- [Finding Modules](#)
- [Downloading and Installing a Module from Drupal.org](#)
- If you do not see the effect of these changes in your site, you might need to clear the cache. See [Clearing the Cache](#).

Videos

[Downloading and Installing a Theme from Drupal.org](#)

Manually Installing Module or Theme Files

Goal

Custom modules and themes may not be available via Composer and their files will need to be manually installed.

Prerequisite knowledge

- [Concept: Modules](#)
- [Finding Modules](#)
- [Concept: Themes](#)
- [Finding Themes](#)

Site prerequisites

Use this approach if you created a custom module/theme or received its files from someone.

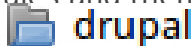
When installing or updating the core software, contributed modules, and contributed themes you should use Composer. See [Using Composer to Download and Update Files](#), [Downloading and Installing a Module from *Drupal.org*](#), and [Downloading and Installing a Theme from *Drupal.org*](#) instead of this topic.

Steps

If you are installing a module or theme from *Drupal.org*, follow the downloading instructions, and then the uploading/unpacking instructions. If you created the module or theme, skip the downloading step. Instead, create an archive file (that you know how to extract on the server) and

proceed with the steps in uploading/unpacking, using whatever method is appropriate for the way you initially created the archive file.

Uploading the files to your site and unpacking them

1. If you are adding a new module or theme, create subdirectories in your top-level *modules* and *themes* directories (if they don't already exist). Typically, people make a *contrib* subdirectory for contributed modules and themes that are downloaded from *Drupal.org*, and a *custom* subdirectory for  at they created. Your *modules*

 **modules**

 **contrib**

directory might look like this:

 **custom**

2. If you are replacing an existing module or theme with an updated version, put the site into maintenance mode. See [Enabling and Disabling Maintenance Mode](#).
3. If you are replacing an existing module or theme with an updated version, find and delete all the existing files and directories for the existing module or theme. Modules are normally located in directories under the top-level *modules* directory, and themes are normally located in directories under the top-level *themes* directory.
4. Upload the *.tar.gz* file (or whatever archive you created) to your site. Place it in either the same location from which you deleted the directory (if replacing an existing module or theme) or the appropriate subdirectory of *modules* or *themes* (if adding a new module or theme).
5. Extract the files from the *.tar.gz* archive (or whatever archive you created), making a subdirectory in the same location as the archive file. If you do not have terminal access, or your hosting server is not running Linux, your hosting control panel's file manager should provide a way to extract the files. If you have terminal access to your hosting server (running Linux) and you are using a *tar.gz* file, you can use a command like:

```
tar -xzf custom_toolbar-2.4.0.tar.gz
```

6. Delete the compressed file from the server, unless your unpacking method already deleted it.
7. Refer to [Downloading and Installing a Module from Drupal.org](#), [Downloading and Installing a Theme from Drupal.org](#), [Updating a Module](#), or [Updating a Theme](#) to complete the installation or update of the module or theme. Start at the step after the automatic download has been completed.

Expand your understanding

- If you work with multiple environments (for example, a local development site and a production site) you will have to repeat the steps on each environment, or re-clone the environment. See [Making a Development Site](#).
- If you added a new theme, navigate in the *Manage* administrative menu to *Appearance* (*admin/appearance*) and uninstall the old theme.

Videos

[Manually Downloading Module or Theme Files](#)

Additional resources

- [Drupal.org community documentation page "Updating modules"](#)
- [Drupal.org community documentation page "Installing Drupal modules"](#)
- [Drupal.org community documentation page "Installing themes"](#)

Concept: Development Sites

What are Development Sites?

Development Sites are different copies of the same site used for developing, updating, and testing a site without risking the integrity of the live site.

An example deployment workflow for site building will usually include the sites mentioned below:

- **Local environment:** The development process starts with developers working on new features, bug fixes, theming, and configuration in their local environment. The recommended tool for setting up a local environment is DDEV. See [Setting Up an Environment with DDEV](#).
- **Development site:** Developers push the changes they've been working on to the development site. For a team of more than one developer, version control is usually used. Git is a version control system that tracks your files for any changes. You can then commit those changes to a repository. Using Git allows team members to work on the same site without overriding each other's work. It also makes it possible to easily roll back to previous stages of the development.
- **Staging site:** The staging site can be used for testing, or presenting the changes to the client for approval. QA (Quality Assurance) and UAT (User Acceptance Testing) are most often carried out on the staging site. It is recommended to have live content on both the development and staging sites, so that you can test how the new features will work with the existing content.
- **Production site:** The live site on the web available to visitors. It contains new features that have been proven safe to go live.

Based on the project's size, scope, requirements, or stakeholders, stages from the above workflow can be removed, or additional stages can be added. For example, a testing site before staging can be added to separate testing and user acceptance processes.

Related topics

- [Making a Development Site](#)
- [Setting Up an Environment with DDEV](#)
- [Concept: Editorial Workflow](#)

- [Managing File and Configuration Revisions with Git](#)

Making a Development Site

Goal

Make a copy of a site that you can use to develop new features and test updates on.

Prerequisite knowledge

[Concept: Development Sites](#)

Site prerequisites

- You have a live, developed site that you would like to make a copy of for development purposes.
- If you want to use Drush for some of the steps in this task, Drush must be installed. See [Concept: Additional Tools](#).
- If you want to use DDEV for your local environment follow the steps in [Setting Up an Environment with DDEV](#). Then, follow the instructions in the DDEV-specific section below.

Generic steps for making a development site

1. Follow the steps in [Preparing to Install](#), so that you have hosting set up for your development site, you know where the web root is for your development site, and you have an empty database and database user for your development site to use.
2. The next step is to make a database dump file, containing the contents of your site's database. This file can be quite large, but there are two things you can do to reduce the size:
 - Compress the file, using gzip or another compression utility.

- Exclude the contents of the database tables for the internal caching system, by truncating these tables. Their data is temporary, and you need only the table structure and not the table contents in order to make a copy of the site. To make the database dump, try one of the following methods:
- If you use Drush, use this command, which will include only the structure and not the contents of the cache tables, and gzip-compress the output:

```
drush sql:dump --gzip --structure-tables-list="cache,cache_*" \
  --result-file='PATH/TO/BACKUPFILE.sql'
```

- If you are using MySQL and have access to the command line, use this command after truncating the cache tables (substituting in your site's database name, user name, and password):

```
mysqldump -u'USERNAME' -p'PASSWORD' DATABASENAME | \
  gzip > PATH/TO/BACKUPFILE.sql.gz
```

- If you are using MySQL as your database, and your live site's server has phpMyAdmin installed (it is available from many hosting control panels), you can truncate the cache tables by selecting them in the phpMyAdmin table structure list and choosing the *Empty* operation at the bottom of the page. Then use the *Export* tab in phpMyAdmin to export in *SQL* format, with *gzip* compression to reduce the file size.
 - Use the [contributed Backup and Migrate module](#) from within your live site. See [Downloading and Installing a Module from Drupal.org](#) for instructions on installing contributed modules. You now have a database dump stored in the file *BACKUPFILE.sql.gz*. For security reasons, avoid storing this file on your hosting server anywhere under the Drupal site root. This will prevent others from getting a copy of your database.
3. Copy all of the files from the web root of your live site to the web root of your development site. You may wish to use Git to do this; if so, see [Managing File and Configuration Revisions with Git](#).
 4. Edit the *sites/default/settings.php* file under your development site's top-level directory in a plain-text editor. Find the lines near the end that contain the database name, database username, and database password, and update them to the information about the development site database you set up. The lines look something like this (before editing):

```
$databases['default']['default'] = [
  'database' => 'live_site_database_name',
  'username' => 'live_site_database_username',
  'password' => 'live_site_database_password',
  ...
]
```

5. Check whether your *settings.php* file has the following setting; if yes, then you will need to edit this to point to your development site URL instead of your production site URL:

```
$settings['trusted_host_patterns']
```

6. Check whether your *settings.php* file has the following setting, and has it set to a random string value. If it does not, then you will need to add or edit it in order to prevent fatal errors:

```
$settings['hash_salt'] = 'any_string_value';
```

One way to produce a random string for the hash salt is the following Drush command:

```
drush php-eval 'echo  
  \Drupal\Component\Utility\Crypt::randomBytesBase64(55) . "\n";'
```

If you don't use Drush, there are numerous web sites and applications that provide random string generators; you'll want to generate a string that is about 74 characters long.

7. Import the database dump file you created, into the development site's database. Try one of the following methods:
- If you are using MySQL as your database, and your live site's server has phpMyAdmin installed (it is available from many hosting control panels), use the *Import* tab in phpMyAdmin. You may find that you have to restart the import a few times, if your database was large.
 - If you are using MySQL and have access to the command line, use this command (substituting in your site's database name, user name, and password; if you made a gzip-compressed backup file, you will also need to uncompress it first):

```
gunzip < PATH/T0/BACKUPFILE.sql.gz | \  
mysql -u'USERNAME' -p'PASSWORD' DATABASENAME
```

- If you prefer to use Drush, use this command:

```
drush sql:query --file='PATH/T0/BACKUPFILE.sql.gz'
```

8. If your development and live sites need to have different configuration, then you have to use configuration overrides in the *settings.php* file. The *\$config* variable will help you maintain override values separately from the standard configuration data. For instance, you might want the site name to be "Anytown Farmers Market" on the production site, but "Development Site for Anytown Farmers Market" on the development site. To do that, you could have the production value in the site configuration (in the database), and on the development site, in the *settings.php* file, you would need to have:

```
$config['system.site']['name'] =  
  "Development Site for Anytown Farmers Market";
```

DDEV-specific steps for making a development site

1. In the directory where you copied all of the files from your live site, run the following command to configure, and start, the DDEV environment:

```
ddev config --project-type=drupal11 --docroot=web  
ddev start
```

This assumes that you are using the recommended directory structure where the web root is in a subdirectory called *web*. If you are using a different directory structure, you will need to adjust the *--docroot* option accordingly.

2. Use the following command to import the database dump file you created:

```
ddev import-db --file=PATH/TO/BACKUPFILE.sql.gz
```

Expand your understanding

- Verify that the development site is working correctly.
- Log in to the development site as an administrator, and clear the cache. See [Clearing the Cache](#).
- [Deploying New Site Features](#)
- [Managing File and Configuration Revisions with Git](#)

Videos

[Making a Development Site](#)

Additional resources

- [Installing a new Drupal application on your local machine](#)
- [Creating a Drupal demo application for evaluation purposes](#)

Deploying New Site Features

Goal

Copy a view that you have created in a local development site to the production site.

Prerequisite knowledge

- [Making a Development Site](#)
- [Synchronizing Configuration Versions](#)
- [Concept: Development Sites](#)

Site prerequisites

- The core Configuration Manager module must be installed in both the development and production sites. See [Installing a Module](#) for instructions on installing core modules.
- The Vendor content type must exist in both the development and production sites, with the same fields. See [Adding a Content Type](#).
- The Vendors view must exist in the development site but not the production site. See [Creating a Content List View](#).

Steps

1. Open the local development site.
2. In the *Manage* administrative menu, navigate to *Configuration > Development > Configuration synchronization > Export > Single item* (*admin/config/development/configuration/single/export*). The *Single export* page appears.
3. Select *View* from the *Configuration type* list.
4. Select *Vendors* from the *Configuration name* list. The configuration appears in the textarea.
5. Copy the configuration from the textarea.

Home > Administration > Configuration > Development > Synchronize

Single export ☆

Synchronize Import **Export**

Full archive **Single item**

Choose a configuration item to display its YAML structure.

Configuration type

View ▾

Configuration name

vendors (Vendors) ▾

Here is your configuration:

```
uuid: 62d8dafb-8ecd-4874-9f30-b97b91df5b3e
langcode: en
status: true
dependencies:
  config:
    - field.storage.node.body
    - field.storage.node.field_main_image
    - image.style.medium
    - node.type.vendor
    - system.menu.main
  module:
    - image
    - node
    - text
    - user
id: vendors
label: Vendors
module: views
description: ""
tag: ""
base_table: node_field_data
base_field: nid
display:
  default:
    id: default
```

Filename: views.view.vendors.yml

6. Open the production site.
7. In the *Manage* administrative menu, navigate to *Configuration > Development > Configuration synchronization > Import > Single item* (*admin/config/development/configuration*). The *Import* page appears.
8. Select *View* from the *Configuration type* list.
9. Paste the configuration in the textarea.
10. Click *Import*. The confirmation page appears.
11. Click *Confirm*.
12. Verify that the view was imported by navigating in the *Manage* administrative menu to *Structure > Views*.

Expand your understanding

The steps in this topic show how to export and import a single configuration item. However, often if you develop functionality on a development website and want to transfer it to your production site, you will need to transfer multiple configuration items. For instance, if you developed a new content type with fields, you would need to transfer several configuration items for each field, one for the content type itself, and possibly multiple view mode and form mode items, and they would have to

be transferred in the right order. Getting this right can be both tedious and difficult.

As an alternative, you can export and import the complete configuration of the site. For this, you would need a local development site that is a clone of the production site (see [Making a Development Site](#)), and then you can follow the steps in [Synchronizing Configuration Versions](#) to synchronize configuration between development and production sites.

Another alternative is to use the [contributed Features module](#), which allows exporting and importing bundled functionality (for example, all the configuration for a photo gallery).

Finally, if you do not see the effect of these changes in your site, you might need to clear the cache. See [Clearing the Cache](#).

Related concepts

- [Concept: Development Sites](#)
- [Concept: Editorial Workflow](#)

Videos

[Deploying New Site Features](#)

Synchronizing Configuration Versions

Goal

Synchronize the configuration between a development and live site.

Prerequisite knowledge

- [Concept: Types of Data](#)
- [Preparing to Install](#)
- [Concept: Development Sites](#)
- [Making a Development Site](#)

Site prerequisites

- You must have a development copy of your production site. See [Making a Development Site](#).
- The core Configuration Manager module must be installed on both the development and production sites. See [Installing a Module](#) for instructions on how to install core modules.
- You must have changed configuration on either the production or development site (the *source site*), and want to synchronize the changes to the other site (the *destination site*). As an example, you can develop a new content type, fields, and views on your development site, and when it is all working correctly, deploy the changes to the live site.
- All configuration that should not be synchronized between the source and destination sites must be stored in configuration overrides in the *settings.php* file rather than in the database. See [Making a Development Site](#).

Steps

1. In the source site, in the *Manage* administrative menu, navigate to *Configuration > Development > Configuration synchronization > Export* (*admin/config/development/configuration/full/export*).
2. Click *Export*. Your site will generate an archive of the full site configuration. Save the file on your local computer.
3. In the destination site, in the *Manage* administrative menu, navigate to *Configuration > Development > Configuration synchronization > Import* (*admin/config/development/configuration/full/import*).
4. Browse to find the downloaded configuration archive, and click *Upload*. Your configuration archive will be uploaded to the destination site, and you will be redirected to the configuration *Synchronize* page (*admin/config/development/configuration*) with a message saying your files were uploaded.
5. Verify that the differences shown on the page are what you expect. You may see configuration items that have been added, deleted, or changed; for changed items, you can click *View differences* to see what the changes are.
6. When you are satisfied, click *Import all* to import the configuration changes.

Expand your understanding

- If the changes you have made involve only one configuration item (such as one view), you can use the single configuration export/import feature to deploy the change between sites. See [Deploying New Site Features](#).
- After the step where you export the full configuration from the source site, you might also want to unpack the archive and commit it to a version control system, such as Git, to track changes in your site configuration. See [Managing File and Configuration Revisions with Git](#).

Videos

[Synchronizing Configuration Versions](#)

Managing File and Configuration Revisions with Git

Goal

Use the Git revision control tool to manage revisions to files and configuration on your site.

Prerequisite knowledge

- [Concept: Additional Tools](#)
- [Concept: Methods for Downloading and Installing the Core Software](#)
- [Concept: Development Sites](#)
- How to set up a Git repository and find its *clone* URL. For example, if you want to use GitLab to host your repository, see [GitLab "Create a project" page](#) and [GitLab "Command Line basic commands" page](#). And if you want to use GitHub to host your repository, see [GitHub "Create a repo" page](#) and [GitHub "Which remote URL should I use" page](#).
- How to open and use a command terminal window and a plain-text editor.
- To manage configuration, how to unpack and pack archive files (such as *.zip* and *.tar.gz*).

Site prerequisites

- You must have downloaded the software for your site, using one of the methods in [Concept: Methods for Downloading and Installing the Core Software](#). If you want to manage configuration, you must have installed the software and have a running site.
- Git client software must be installed on your site's server. See [Git](#) for instructions.
- You must have a new repository created and know its Git clone URL.

Steps

Initializing the repository

Do these steps once, after creating a Git repository, to connect your local directory to the repository and add the initial files to it.

1. Open a command terminal window, and change to the directory where your site's files are located. This is your "top" directory.
2. Determine where your web root is. If the *core*, *modules*, and *themes* directories are located directly in this directory, then you are in your web root. If you have used Composer to download the software, then these files are located inside the *web* subdirectory (which is your web root).
3. In a plain text editor, create a new file called *.gitignore* in the top directory (or edit it if it already exists). This file contains a list of files and directories that Git should ignore (not add to the repository). For example, the *settings.php* file for your site should not be added to Git, because it contains your database account information, and the media files uploaded to your site (usually in *sites/default/files*) should not be in Git either -- the objective is to have the software in the repository, not the site content.
4. Make sure the following lines are in the *.gitignore* file. If your web root is not your top directory, check each of these to see if they need a prefix. For example, *sites* may need to be replaced with *web/sites*.

```
sites/*/settings*.php
sites/*/files
config
```

5. Enter the following commands:

```
git init
git add -A
```

6. Optionally, verify the list of files you will be adding to your Git repository by entering this command and scrolling through the (very long) list:

```
git status
```

7. Enter the following commands. You can substitute your own *commit message* for "Initial file add" if you wish, and you will need to substitute your own Git clone URL for the URL in the second command:

```
git commit -m "Initial file add"
git remote add origin https://gitlab.com/example-name/example-repo.git
```

```
git push -u origin main
```

8. If you are using GitLab, GitHub, or another host with online access, you can now go to your repository page and see that the files are there.

Updating files in the repository

Use these steps when you have updated, added, or deleted one or more files in your site, in order to send the changes (*push*) to your repository.

1. Open a command terminal window, and change to the directory where your site's files are located.
2. Check the list of files that have been added, changed, or deleted:

```
git status
```

3. Optionally, for text files that have been changed (not images), look at the differences between the new and old versions of the file:

```
git diff path/to/file.txt
```

4. Stage all the changes for the next commit, and verify that they are staged:

```
git add -A  
git status
```

5. You can omit a particular file from the commit that you have already staged, or add another file to the ones you have already staged. If a particular file or directory keeps getting added by mistake, consider adding it to the *.gitignore* file so that it will be ignored by Git. Omit/add commands:

```
git reset HEAD path/to/file.txt  
git add path/to/file.txt
```

6. Commit and push your changes. Substitute something meaningful for the commit message:

```
git commit -m "commit message here"  
git push
```

7. If you have other copies of your repository, update them by opening a command window in the directory of each copy and running the following command:

```
git pull
```

Making a copy of the files in your repository

Follow these steps if you want to copy all the files in your repository to a new location. For example, you might have both a local development copy of your site and a production site, or several team members might all have local copies of the site.

1. Open a command window in the location where you want the files to be.
2. Enter the following command, substituting your repository clone URL for the URL, and the name of the subdirectory you want them in for *dirname*:

```
git clone https://gitlab.com/example-name/example-repo.git dirname
```

Managing configuration in the repository

1. Follow the instructions on [Synchronizing Configuration Versions](#) to export a complete archive of your site's configuration.
2. If you have not already initialized configuration in the repository, unpack the configuration archive into a new directory, preferably above the web root directory, and follow the instructions above to add these files to your repository.
3. After initializing, whenever your site configuration changes, export and unpack the configuration archive in the same location. Follow the instructions above to update these files in your repository.
4. To import updated configuration to another site, make an archive of the configuration directory from your repository. Then follow the instructions on [Synchronizing Configuration Versions](#) to upload and import this archive into the site.

Related concepts

[Concept: Development Sites](#)