

Making a Development Site



- [Goal](#)
 - [Prerequisite knowledge](#)
 - [Site prerequisites](#)
 - [Generic steps for making a development site](#)
 - [DDEV-specific steps for making a development site](#)
 - [Expand your understanding](#)
 - [Videos](#)
 - [Additional resources](#)
-

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/T0/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/T0/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/TO/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/T0/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](#)

📺 #4

📺 cocobook-sync 📺 2026-10-28 16:16:11 UTC

📺 cocobook-sync 📺 2026-10-29 04:15:00 UTC