DRUPAL WEBSITES

George Ciesinski

HUMBER COLLEGE 205 Humber College Blvd., Toronto ON M9W 5L7

INTRODUCTION

This document contains information that is shared between all the Humber ITS Websites built using Drupal.

This includes information about settings, core and module updates, backing up and restoring sites, content updates, or possible bugs.

START DATE

The design & build process for the ITS websites started on April 2023.

TEAM MEMBERS

George Ciesinski – Web Developer

Michael Boadu – Marketing and Communications Coordinator

REQUIREMENTS

The ITS website serves tens of thousands of Faculty Members, Staff and Students in a College environment. The below requirements will help to provide a positive and secure experience to all website users:

1. **Security** – The website should stay up to date with major security updates, and should include authentication for all users who modify the content of the site.
2. **Optimization** – All images, videos, and other assets should be optimized before usage on the site to ensure a fast browsing experience.
3. **Accessibility** – Content should meet the accessibility requirements outlined in the AODA (Accessibility for Ontarians with Disabilities Act)

TOOLS

HUMBER WEB ACCESSIBILITY COMPLIANCE (AODA)

This page provides access to the WCAG Quick-Reference guide and AODA Compliance Reference, as well as various tools that assist in evaluating Web Accessibility.

https://humber.ca/tutorial/web-accessibility-compliance.html

GUIDELINES

HUMBER INTERIM WEB GUIDELINES

This document provides guidelines for standards Humber Websites should meet.

https://humber.ca/brand/sites/default/files/publications/interim-web-guidelines.pdf

STACK

Drupal is based on the LAMP stack: Linux, Apache, MySQL, and PHP Scripting Language.

Drupal can also run on other technology stacks. For example:

- Windows, Mac or Linux
- Apache, Nginx or IIS
- PostgreSQL, SQLite, or MySQL

HUMBER ITS STACK

The Humber ITS Stack uses:

- Linux OS
- Apache Server
- PostgreSQL
- PHP

DEVELOPMENT

BACKUPS

It is important to back up the website whenever there is a core/module update, or when significant amounts of content are added. This ensures that the site can be restored in case of any unforeseen event. Additionally, this allows us to migrate the site from one server to another if needed.

Needless to say, backups are very important and should be tested to ensure it is possible to restore the site from the backups.

GIT

Git is a kind of version control used by the Humber ITS team to back up Drupal site directories. This is not a complete back-up of the site, but instead contains changes to source code and some configurations for the site.

When using these docs on an existing site, some parts of this section can be ignored. For example, you may not need to know how to initialize a new repository for an existing site, but you should always create branches for new major features, and commit your changes regularly.

REQUIREMENTS

Some level of Git knowledge is required in order to follow this section. Version control is vital for modern web and software development and is used globally throughout the world. Git is currently the most used version control standard.

To learn more about Git, check out the Atlassian git documents: https://www.atlassian.com/git/tutorials

INITIALIZING A REPOSITORY

Todo

CREATING A NEW BRANCH

Todo

COMMITTING CHANGES

Todo

MORE INFORMATION

Some more information can be found in these links. These are meant to get a better understanding of how Git can be used on a Drupal site, as well as best practices and the Git-flow method of creating branches. It should not be used in place of the other instructions in this document.

https://www.drupal.org/docs/installing-drupal/building-a-drupal-site-with-git

https://www.drupal.org/docs/user_guide/en/extend-git.html

https://www.drupal.org/docs/installing-drupal/building-a-drupal-site-with-git

https://nvie.com/posts/a-successful-git-branching-model/

BACKUP DATABASE

CONCEPT: DATA BACKUPS

https://www.drupal.org/docs/user_guide/en/prevent-backups.html

BACKUP USING COMMAND LINE

https://www.drupal.org/docs/7/backing-up-and-migrating-a-site/back-up-your-site-using-the-command-line

PG_DUMP

https://www.postgresql.org/docs/7.1/app-pgdump.html

PG_RESTORE

https://www.postgresql.org/docs/7.1/app-pgrestore.html

DEVELOPMENT SITE

It is not recommended to carry out core and module updates on the live/production version of the site because it could introduce breaking changes that are difficult to reverse. It may also result in an unexpected outage. Instead, it is recommended to set up a local development site which is essentially a clone of the live/production server. The process of setting up a local development site can also be used to verify the integrity of the backup files.

This development site is used for updates and major changes as well as for testing. Once the testing is complete, the changes can be pushed to the live site.

MAKING A DEVELOPMENT SITE

https://www.drupal.org/docs/user_guide/en/install-dev-making.html

MAINTENANCE

MAINTENANCE MODE

Presents message that the site is under maintenance to users who do not have the right permissions. Authorized users can maintain the site and update content during this time.

https://www.drupal.org/docs/user_guide/en/extend-maintenance.html

UPDATING

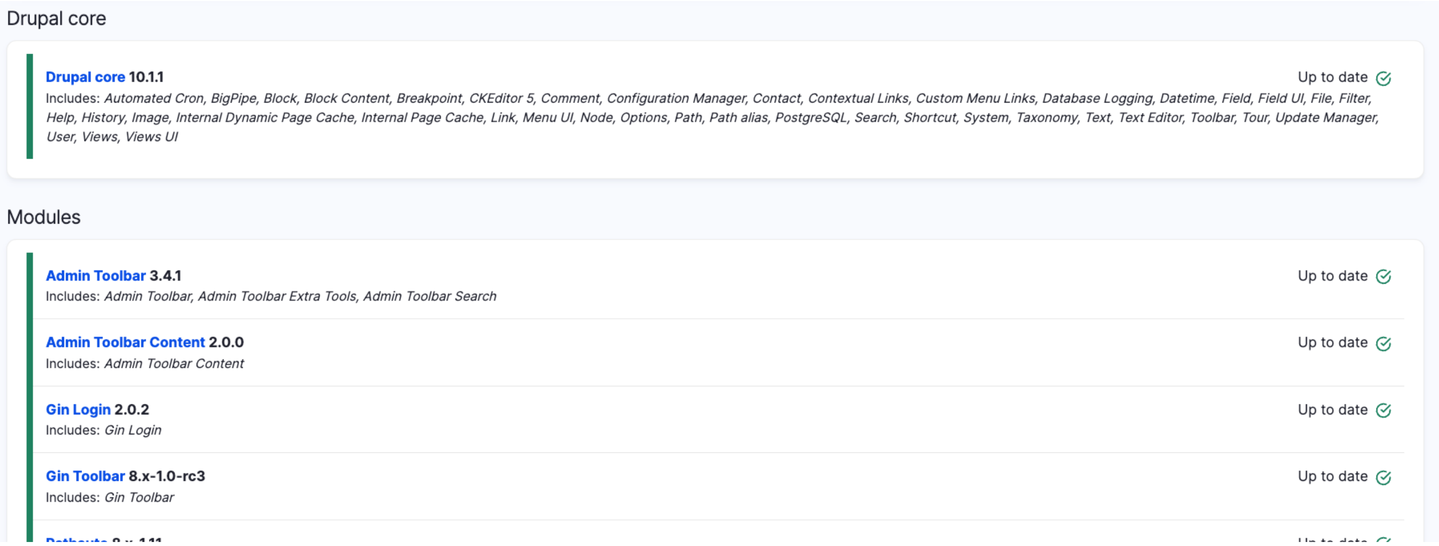
Drupal core and the various modules the site uses are frequently updated to address security issues and bugs that come up over time. In many cases, it is beneficial to update these components to ensure that the Humber site is secure and working correctly.

HOW TO TELL THERE ARE UPDATES

Updates can be checked using the Admin toolbar on the website by visiting the Available Updates section.

> *Reports > Available Updates*

This will show a screen like this which shows if Drupal core and modules are up to date:



Drupal core

Drupal core 10.1.1 — Up to date ✓
Includes: *Automated Cron, BigPipe, Block, Block Content, Breakpoint, CKEditor 5, Comment, Configuration Manager, Contact, Contextual Links, Custom Menu Links, Database Logging, Datetime, Field, Field UI, File, Filter, Help, History, Image, Internal Dynamic Page Cache, Internal Page Cache, Link, Menu UI, Node, Options, Path, Path alias, PostgreSQL, Search, Shortcut, System, Taxonomy, Text, Text Editor, Toolbar, Tour, Update Manager, User, Views, Views UI*

Modules

Admin Toolbar 3.4.1 — Up to date ✓
Includes: *Admin Toolbar, Admin Toolbar Extra Tools, Admin Toolbar Search*

Admin Toolbar Content 2.0.0 — Up to date ✓
Includes: *Admin Toolbar Content*

Gin Login 2.0.2 — Up to date ✓
Includes: *Gin Login*

Gin Toolbar 8.x-1.0-rc3 — Up to date ✓
Includes: *Gin Toolbar*

If any of these components do not show *Up to date* and a green check mark, those components may have updates.

SHOULD THEY BE UPDATED

Updates should be made if they address a security flaw or a bug that may affect the website. You can check this by visiting the pages of the Drupal core version or the module project pages and reading the release notes. Any significant changes will be listed there, and a decision can be made based on this information.

UPDATING CORE SOFTWARE

https://www.drupal.org/docs/user_guide/en/install-composer.html

UPDATING MODULES

https://www.drupal.org/docs/user_guide/en/security-update-module.html

UPDATING THEME

https://www.drupal.org/docs/user_guide/en/security-update-theme.html

TESTING UPDATES

Updates should be made on the development website and tested thoroughly before pushing to production. This would help with catching any incompatibilities or issues that may come up from the updates.

SETTINGS.PHP

The *settings.php* file contains sensitive data about the website such as the database connection, trusted host, and configuration information.

This file should NOT be uploaded to Github. Instead, it should be backed up in a secure location.

The file is also read-only by default which is required to ensure the site is secure, however, there are many instances where the administrator may have to edit this file. The instructions to do this are below.

LOCATION OF SETTINGS.PHP

The settings file can be found in: */project-folder/web/sites/default/*

MAKING CHANGES TO SETTINGS.PHP

The *settings.php* file is read-only by default which is a security precaution to ensure the site is not able to alter it in any way. In order to make changes, you must locally change the permission of the file to give yourself write access, make the required changes, and once again harden the permissions. This ensures that the file contents are only changed in an authorized way and not by malicious actors.

1) Open the terminal and *cd* into the above directory.
2) Use the below command to make the file editable:

```
chmod a+w settings.php
```

3) Make the required changes using a text editor like vim or nano.
4) Harden the permissions after editing the file:

```
Chmod 444 settings.php (Results in permissions -r-r--r--)
```

Alternative command *(not recommended as it gives permissions -rw-r--r--)*:

```
chmod go-w settings.php
```

FREQUENTLY REQUIRED CHANGES

TRUSTED HOST SETTINGS

The trusted host settings tell the website which hosts are authorized to access the site. This is an important security measure to ensure that fraudulent hosts cannot be used to access the site or create unauthorized clones.

This setting essentially tells the site which URL is allowed to access the site, and it affects both the production/live version of the site and the local development version. In the event that the host name changes, the trusted host settings must be modified to reflect this.

The trusted host setting looks like this in *settings.php*:

```
$settings['trusted_host_patterns'] = [
  '^www\.its-cab\.test$',
  '^its-cab\.test$'
];
```

This is an array containing multiple comma-separated lines with the various host patterns. The standard is to indent each line with two spaces. The patterns themselves are defined using REGEX.

UPDATE FREE ACCESS

This setting is used during core updates.

Todo

ROLES

USERS

Users are accounts with usernames and passwords. Drupal sites come with a User 1 account and all other users must be created by the admin, or by new users requesting accounts on the site.

More information about users can be found here:

https://www.drupal.org/docs/user_guide/en/user-concept.html

CONFIGURING USER ACCOUNT SETTINGS

It is possible to configure many aspects of user accounts including whether users should be able to request/open their own accounts, or whether an admin has to manually do this step.

More information can be found here:

https://www.drupal.org/docs/user_guide/en/config-user.html

USER 1

The User 1 account is the first account created along with the site and is known as the diety account in Drupal. This is because it has significant administrative powers and can be used to create other admin accounts. It also has the power to perform any action on the site regardless of permissions.

ANONYMOUS USERS

Anonymous users are any users who visit the site without logging in. This is defined as a role and the permissions for these users can be separately configured.

AUTHENTICATED USERS

Authenticated users are any users who have accounts and can sign in.

ADDITIONAL ROLES

Various roles can be created on the site. These roles can be named anything you want, and the permissions for each role can be customized in many ways. For example, it is possible what content an account can see, edit, create and delete. There is much more to the permissions system, so to get a thorough understanding of it, it is necessary to dig into the official Drupal docs.

The Drupal User Guide Chapter 7 discusses Managing User Accounts in more detail:

https://www.drupal.org/docs/user_guide/en/user-chapter.html

CREATING A NEW ROLE

A new role can be created by following the instructions here:

https://www.drupal.org/docs/user_guide/en/user-new-role.html

ASSIGNING PERMISSIONS TO A ROLE

Once a role is created, the permissions must be defined. It can be helpful to look at other roles' permissions to see how they are set up, and create the permissions based off this. In some cases, you may need to dig deeper about which permission to add to a role in case you are trying to achieve a specific purpose.

Note: It is NOT recommended to add a permission to a role if it will only benefit a single user. This would grant that permission to every user who is assigned to that role. In that case, it would be beneficial to create a new unique role for that user.

The instructions to assign a permission to a role can be found here:

https://www.drupal.org/docs/user_guide/en/user-permissions.html

CHANGING A USER'S ROLE

Roles must be assigned to new users so they can enjoy the permissions of that role. Users with no assigned roles are simply authenticated users. In order to assign special permissions to those users, it is recommended to assign a role to them and not to modify the permissions of authenticated users.

To change a user's role, follow the instructions here:

https://www.drupal.org/docs/user_guide/en/user-roles.html

MODULES

Modules extend functionality and add features. Installing the feature adds functionality, while uninstalling it removes it. Modules increase the time needed to generate a page, so it is important to only keep modules you use installed and remove the unused ones.

In general, modules will not be added to sites frequently. They will only be added to create a new feature that may require a contributed module in order to function correctly.

Most actions we perform on modules will be maintenance as new versions are released.

INSTALLATION & MAINTENANCE

INSTALLING MODULES

Installing modules is well documented in the Drupal User Guide:

https://www.drupal.org/docs/extending-drupal/installing-modules

USING COMPOSER TO MANAGE DEPENDENCIES

All the Humber ITS Websites are builit with Composer. This makes adding and updating modules very simple. You can read more about it in the Drupal User Guide:

https://www.drupal.org/docs/develop/using-composer/manage-dependencies#adding-modules

CONSIDERATIONS BEFORE UPDATING

The website might flag some modules that are outdated, but care should be taken before updating any module. It is important to go over the various modules in use on the site and check the module Drupal pages to see if there are any special instructions for carrying out updates.

Furthermore, it is important to make sure we are using the correct module for the core version. If using Drupal 10, only modules compatible with Drupal 10 should be used. If a new version of a module is built for Drupal 11, the core files should be updated first.

I will go into further detail about maintenance further in this document.

SHARED MODULES

Shared Modules are any modules that are used across most or all of the Humber ITS Drupal sites.

ADMIN TOOLBAR

Improves the default toolbar and changes it into a drop-down menu with fast access to each administrative page.

In the Extend menu, the following options are also enabled:

- Admin Toolbar Content
- Admin Toolbar Extra Tools
- Admin Toolbar Search

https://www.drupal.org/project/admin_toolbar

R4032 LOGIN

This module is used to force users to Login when they attempt to access pages or content that they do not have the permission to see. This is mainly used to force Anonymous users to sign in.

https://www.drupal.org/project/r4032login

PATHAUTO

This module is used to automatically generate unique URL's for new content. Default Drupal behavior results in URL snippets like *node/1*, but PathAuto can be used to create templates so that URLs might instead look like *photo/2* or *change-request/143*.

https://www.drupal.org/project/pathauto

https://ostraining.com/blog/drupal/pathauto-2/ (Instructions)

LDAP / ACTIVE DIRECTORY INTEGRATION

This module adds Active Directory integration to sites so that this can be used as an alternative sign-in method.

https://www.drupal.org/project/ldap_auth

& Instructions: https://www.drupal.org/docs/contributed-modules/ldap-integration/ldap-user-role-mapping

GIN THEME

A number of Gin theme modules are also installed.

GIN ADMIN THEME

https://www.drupal.org/project/gin

Beautiful and sleep admin theme.

GIN LOGIN

https://www.drupal.org/project/gin_login

Gin Login provides a beautiful login page with a splashdown image.

GIN TOOLBAR

https://www.drupal.org/project/gin_toolbar

Helper module that brings Gin theme to the admin toolbar.

USEFUL MODULES

Useful Modules are any Modules that provide useful features but are not yet used on the Humber site. These can be considered in future updates if it is deemed that they are necessary for a prospective feature.

TODO: Add modules from Building a Drupal site with Git | Installing Drupal | Drupal Wiki guide on Drupal.org

ALBUM PHOTOS

Needs to be compared to core photos module to see if this is necessary or not.

https://www.drupal.org/project/photos

VIDEO

Allows you to host and play videos on the platform.

https://www.drupal.org/project/video

FIELD GROUP

Allows you to group fields so that they are easier to organize.

https://www.drupal.org/project/field_group

FIELD PERMISSIONS

Provides field level permissions for different roles.

https://www.drupal.org/project/field_permissions

DELETE UNUSED FILES

This module allows unused files to be safely deleted. Default Drupal behavior archives the files so that a copy always exists on the site.

https://www.drupal.org/project/delete_unused_files

ADMINISTRATION

ONE TIME LOGIN

Note: drush is required in order to follow these steps.

If a user is unable to login for some reason, it is possible to send them a one-time login using drush uli:

```
drush uli --name=username  --uri=website-url
```

If successful, it will output a URL that the user can use to login and change their password.

ADDITIONAL INFO

DOCUMENTATION

https://www.drupal.org/documentation

GLOSSARY

https://www.drupal.org/docs/user_guide/en/glossary.html

FURTHER READING

REACT FOR DRUPAL

https://reactfordrupal.com/tutorials/react-for-drupal-developers