
Ark Development Open Source Standards Documentation

Release 0.1

Ark Development Open Source Ninjas

February 29, 2016

1	Wiki <i>Free & Open</i>	3
1.1	Content	3
2	Zen Coding And Netbeans Templates	5
3	Git Tips and Tricks	7
3.1	Git Bare Repos	7
3.2	Git Shell Prompt	8
3.3	Git Workflow	9
4	Linux Administration	11
4.1	CUPS shared printer troubleshooting	11
4.2	Fix Files & Folders Permissions	12
4.3	Internal Network Fixer	12
4.4	Jenkins Installation & Configuration <i>CentOs</i>	13
4.5	EasyApache mod_proxy_wstunnel	16
4.6	Dev Server Virtual Machines	16
4.7	Build PHP 5.2 on CentOs 6	17
5	Drupal Gotcha	19
5.1	Conditional Fields	19
5.2	Bootstrap Checkbox Tables	19
5.3	Drupal Testing Scripts	20
5.4	Drupal 8 Profiler Storage	20
5.5	Send Multi part Mail in Drupal 7/8	21
6	Front-end Patterns	25
6.1	Sticky Footer	25
6.2	Patterns for Website Banners	26
7	Drupal 8 Reference	29
7.1	PDF Starter	29
7.2	Drupal 8: Installation Guide	29
7.3	How to Dependency Injection in drupal ?	31
7.4	Drupal 8: Menu Links in Admin Config	33
7.5	Drupal 8: Content Entities	35
7.6	Twig Debug	36
7.7	Drupal Twig Extension	37

Contents:

Wiki *Free & Open*

This wiki should be easy way to collect lesson learned and quick tip and tricks. Although all the available tools are free and open source, the mentality behind these tips and tricks should be applicable across different technologies.

Required software and code snippets can be found in the project files.

1.1 Content

- Editors and IDEs Tricks
 - Zen Coding and Netbeans Templates
- Git Tips and Tricks
 - Git Bare Repos
 - Git Shell Prompt
 - Git Workflow
 - CUPS shared printer troubleshooting
- Linux System Administration
 - Fix Files & Folders Permissions
 - Internal Network Fixer
 - Jenkins Installation & Configuration - *CentOs* Edition
 - EasyApache mod_proxy_wstunnel
 - Dev Server Virtual Machines
 - Build PHP 5.2 on CentOS 6
- Drupal Gotcha
 - Conditional Fields
 - Bootstrap Checkbox Tables
 - Drupal testing scripts
 - Drupal 8 Profiler storage
 - Drupal Mail

- [Drupal 8 Reference](#)
- [Front-end Patterns](#)
 - [Sticky Footer](#)
 - [Banner Patterns](#)
- [Node.js](#)
 - [Installation](#)
 - [Version Updates](#)

Zen Coding And Netbeans Templates

The objective is to write code quickly and efficiently. This tip focuses on HTML code, because it can be tedious to write, however, the technique can be applied to other languages using code editor plugins and different utilities.

Zen Coding

The main principle behind then coding is not to repeat yourself. It provides a short syntax to write HTML code.

For netbeans, you can find the plugin under Tools > Plugins > Available Plugins > Zen Coding. After installation and restarting Netbeans, you can start testing in any HTML context files like `index.html` or `index.xhtml` and test the following code

```
html:5
(header>ul>li>a.item-${*3}).container>.col-md-6.col-xs-12>p*2
```

Hit **Alt+Ctrl+n** shortcut to start converting the sample to actual HTML code

```
<!DOCTYPE HTML>
<html lang="en-US">
  <head>
    <meta charset="UTF-8">
    <title></title>
  </head>
  <body>

  </body>
</html>
```

And

```
<header>
  <ul>
    <li>
      <a href="" class="item-1"></a>
      <a href="" class="item-2"></a>
      <a href="" class="item-3"></a>
    </li>
  </ul>
</header>
<div class="container">
  <div class="col-md-6 col-xs-12">
    <p></p>
    <p></p>
```

```
</div>  
</div>
```

For more details check [Zen Coding cheat sheet - PDF](#)

These tools are here to help, but don't try to overload yourself with their details. Just keep it simple and enjoy coding with speed.

Git Tips and Tricks

3.1 Git Bare Repos

Git `--bare` is a special repository that doesn't contain any working directory and you cannot directly checkout branches from it. Bare git repos are most suitable for adding them as remotes to your local repositories in order to share your changes with other users.

3.1.1 Remote Server

First initialize bare git

```
git init --bare <REPO_NAME>.git
```

The `<REPO_NAME>` should be anything related to your project that contains alphanumeric characters (in both capital and small cases), underscores, dashes and dots (without and slashes, square and angular bracket, parentheses or curly braces). Adding `.git` suffix is just for convention, and recommended but not required.

This will create a new directory named `<REPO_NAME>.git`.

Important Note: All developers will access the repository through single user via ssh, so it's important to make sure that you can access the repository folder through ssh!

3.1.2 Local Users Access

Now we have an accessible remote repository, we need to add our local ssh private key in order to automatically authenticate through ssh.

```
cd ~/.ssh
ls id_rsa.pub
# If you don't find this file, use the following command
# $ ssh-keygen
# Just agree with all the defaults
cat id_rsa.pub
```

Copy the output from `~/.ssh/id_rsa.pub`, and append it to the end of file `/home/<remote_user>/.ssh/authorized_keys` in the remote server. You can download the file through FTP and update it locally and upload it again, or you can create it if not found.

To test that ssh authentication is working you can use the following command

```
ssh <remote_user>@<remote_server_ip>
```

If everything works correctly should be logged in to remote server without being asked for any password.

3.1.3 Git Remote

To find the remote URL , use the following schema: `ssh://@:/path/to/git/bare/repo.git`

Each developer will have to add the information of the newly created repo to their local repository.

```
cd /path/to/local/git/project
git remote add <remote_server_alias> <remote_git_url>
```

For the first/lead developer, you will have to populate the newly created remote repository with the following command.

```
git push -u <remote_server_alias> master
```

If you need to add all branches you can add `--all` flag as follow:

```
git push --all -u <remote_server_alias>
```

3.1.4 Helpful Notes

- Add the to your `/etc/hosts` file, in order to use a simple domain name instead of server ip. Now use the domain name instead of the ip in all commands.
- can be any short name that contains alphanumeric and underscores character (It's not recommended to use any characters)
- Default is `origin`, but you can add as many remotes as you want.
- is defined only per developer per project, so different developers can have different

For More Information/Details You can find plenty of resources on the web.

3.2 Git Shell Prompt

Do you have several branches and not sure which is you current branch? Do you type `git branch` so many times just to keep track of your working branch?

The solution to this is to have your terminal prompt display the current branch. Windows *Git Bash* users have it why not us !Linux! users?!

Go to this [awesome GitHub project](#) for **Git Aware Prompt** and follow the installation instructions.

1. To use the new shell prompt find your bash initialization script `~/.bash_profile` or `~/.bashrc` or `~/.profile`.
2. Check your `PS1` prompt variable

```
echo $PS1
```

3. Append the following line at the end of file.

```
export PS1="\[\e]0;\u@\h: \w\a\}${debian_chroot:+($debian_chroot)}\u@\h:\w \[$ttxtylw\]\$git_branch
```

4. To test the new prompt:

```
source ~/<bash init script name>
```

Where is the name of the file you just edited.

If you find any problems you can comment the newly added line by putting # at the line start.

You can update the PS1 value according your original PS1 from step 2

3.3 Git Workflow

3.3.1 Overview

Workflow is a consistent standard ways of how we use any project git

repository and branches. | This workflow is a simplified version of the first [Git branching model](#).

How it works? Every project have two different types of branches: * Infinite Lifetime Branches * Supporting Branches

Infinite Lifetime Branches are special branches that should always

exists in any project repository at any given time. Most of the time they reflects actual deployed instances of the project on live sever. | The two main branches are: * master * develop

On origin/master with source code on HEAD always reflects

production ready state. You can always checkout the origin/master branch and deploy the code to the production with no fear of breaking live site. | This means that if any file was changed or deleted from production site by accident, you can always revert it back by checking it out of origin/master/HEAD.

On origin/develop with source code on HEAD always reflects

next release ready state. All features that are ready for client review and will be deployed on the next release exists on origin/develop/HEAD. | If the client wants to test and integration of all requested features ready for the next production deployment, you can checkout origin/develop branch and deploy it on stage site (e.g., stage.example.com).

Supporting Branches have short lifetime and most of the time they are created for certain purpose and can be safely deleted after merging into other branches or discarded completely. There are three different types of branches: * Feature Branches * Hotfix Branches * Release Branches

Feature Branches usually have a name in the format feature/...,

they represent a distinctive feature request or client change request, that is modular enough to safely revert it if requested. | There can be any number of feature branches in the project repository at any point of time.

Hotfix Branches usually have a name in the format `hotfix/...`, they represent bugs and issues found by client in the production site. Issues found in development instance by team members including internal testing team should be solved in their own *Feature Branches*.

Release Branches are very special type of branches that are needed in only large projects, where there are many activities needed before deploying any major release, like generating the documentation and release preparation.

Linux Administration

4.1 CUPS shared printer troubleshooting

Sometimes due to communication errors, the shared printer at 192.168.1/2.11 get *paused*.

This is simple issue that a simple *resume* command can bring the printer back to work.

Here are simple steps to do that. *[All using SSH] :(*

4.1.1 Using Browser

This is the hard option. First you need to do something called “Port Forwarding”.

From Shell:

```
ssh -L 6310:localhost:631 root@git-server
```

From Putty: Search Google how to configure Putty to do that. Use the local port: 6310 and remote port:631

Now keep the terminal running and go to browser and use the following URL: `http://localhost:6310`

This is the graphical interface for CUPS, the printer management server. From top navigation menu choose “Administration” *(If asked for credentials, use the root username and password.)*

Choose Manage Printers button, to access a list of available printers. Choose `shared_printer` *(This should be the only available printer)*

From Maintenance drop down, select Resume Printer. If you can’t find this option, then there is another problem with the printer. You can check that the printer is actually “On”, connected to the server via USB, with paper in the tray and no paper is jammed inside the printer.

4.1.2 Server Logs

If you already have a terminal and want to check if there are any printer errors, use the following commands:

```
tac /var/log/cups/error.log | less
```

This command will list the latest error logs from today. Hit `q` to stop this command and exit back to terminal.

For listing older logs use the following commands:

```
zcat /var/log/cups/error_log.* | tac | less
```

This will aggregate all error logs available order that today.

Using the past two commands will list logs in a reverse chronological order - latest logs first.

4.2 Fix Files & Folders Permissions

4.2.1 Introduction

Default file permissions should be 644 and default folder permissions should be 755. Sometimes while working on partitions that doesn't recognize *Linux File Permissions* such as: *NTFS*, all files and folders permissions turns to 777 which propose a *major* security risk.

4.2.2 Find Bad Perms

To find bad permissions in certain *folder path*: *sites/all* for example, use the following command:

```
find sites/all/ \( -type d -perm -o+w -exec ls -dl '{}' \; \) , \( -type f -perm -o+x -exec ls -l '{}' \;
```

This will list all files and folders with bad permissions. You can finish the command by pressing `q` key.

4.2.3 Fix Bad Perms

To fix all bad permissions of the previously found files and folders, use the following command:

```
find sites/all/ \( -type d -perm -o+w -exec chmod 755 '{}' \; \) , \( -type f -perm -o+x -exec chmod 644 '{}' \;
```

This command is silent, and you'll see no output until it finishes and return the prompt back to you.

PLEASE NOTE: The command will operate on *all* files and folders found by the first `find` command. Make sure that you review the list or be more specific with the folder path.

4.3 Internal Network Fixer

4.3.1 Challenge

Automatically change all IPs of *ArkDev* internal servers in our local `/etc/hosts` file according to current machine local network `I/2`.

4.3.2 Network Fixer Script `ark_network_fixer.sh`

This script checks the current local network, and make sure that you have entries in you `/etc/hosts` that has IPs on the wrong sub-network, and changes these entries automatically.

You can call this script manually. It has an interactive mode that show the changed `/etc/hosts` file, and can take backup files of your `/etc/hosts` files before fixing it.

To run this script manually you have to be *root* or use `sudo` command.

- Download this file from the wiki project files.

- Change file permissions to executable
- Change Owner to *root*
- Place the file in your `/usr/local/bin` folder

```
wget http://dev-server/open-source/opensourcewiki/raw/master/ark_network_fixer.sh
chmod 755 ark_network_fixer.sh
sudo chown root:root ark_network_fixer.sh
sudo mv ark_network_fixer.sh /usr/local/bin
```

4.3.3 Network Change Dispatcher 99ark_network_fixer

This simple script is used by *NetworkManager* to detect any changes in network interfaces and call the **ark_network_fixer.sh** script.

If you don't have *NetworkManager* on your system, you can consult your distribution documentation or online help and find where you need to call the **ark_network_fixer.sh** script.

This file depends on **ark_network_fixer.sh** and you must follow the previous steps to download and configure **ark_network_fixer.sh** for this file to work. But **ark_network_fixer.sh** doesn't need this file to work.

- Download this file from the wiki project files.
- Change file permissions to only executable by owner.
- Change Owner to *root*
- Place the file in `/etc/NetworkManager/dispatcher.d` folder

```
wget http://dev-server/open-source/opensourcewiki/raw/master/99ark_network_fixer
chmod 700 99ark_network_fixer
sudo chown root:root 99ark_network_fixer
sudo mv 99ark_network_fixer /etc/NetworkManager/dispatcher.d
```

4.4 Jenkins Installation & Configuration *CentOs*

Practical guide for installing Jenkins as part of setting up *Continuous Integration (CI)* server.

This guide can be applied to any Linux distribution, but tested only on Centos 6.6.

4.4.1 Steps Outline

- *Create bare git repository on server*
- *Jenkins Installation*
- *Jenkins Security Configuration*
- *Jenkins Git Configuration*

4.4.2 Git Bare Repo

Create a new user called `git` that will be used to create the bare git repo. Add `ssh authorized_keys` file to allow `ssh` authentication without password. Prevent interactive `ssh` login to `git` user account by changing login shell to the special `git-shell`.

```
adduser git
su git - # Switch to git user
cd
mkdir .ssh && chmod 700 .ssh
touch .ssh/authorized_keys && chmod 600 .ssh/authorized_keys
exit # exit git shell and return to root user
cat /root/.ssh/authorized_keys >> /home/git/.ssh/authorized_keys
which git-shell >> vi /etc/shells
chsh git -s `which git-shell`
```

Make sure that the server has Git installed, then follow the guide to create new bare git repository.

Post Git Init Checks

After creating the new Git bare repository, remember to: * You have added all allowed users passwords in the `.ssh/authorized_keys` file in git home directory.

4.4.3 Jenkins Installation

Jenkins [official documentation for installation](#) covers the basic steps and gotchas that you might face while installing Jenkins.

Post Installation Checks

- Make sure that Jenkins default port (8080) is open in the firewall configuration.
- Jenkins will try to execute binaries from `/tmp` folder. Make sure that `/tmp` folder is not mounted with option `noexec` using `mount` command. Otherwise (change Jenkins temporary folder)(#change-jenkins-temp-folder)

Change Jenkins Temp Folder

If you have problems starting Jenkins and the log say it can't execute with exception like `InvocationTargetException` "Operation not permitted", most probably you'll need to change the temporary folder that Jenkins uses.

1. Open `/etc/sysconfig/jenkins`.
2. Change the Java options line
 - *From:* `JENKINS_JAVA_OPTIONS="-Djava.awt.headless=true"`
 - *To:* `JENKINS_JAVA_OPTIONS="-Djava.awt.headless=true -Djava.io.tmpdir=$JENKINS_HOME/tmp"`
3. Find the `JENKINS_HOME` directory value from the same file (Usually: `/var/lib/jenkins`).
4. Create an empty `tmp` folder inside the `JENKINS_HOME` directory.
5. Make sure that the `tmp` folder is writable by the Jenkins user.
6. Start Jenkins: `service jenkins start`

4.4.4 Jenkins Security Configuration

By default Jenkins allow anonymous users to do everything. To tighten Jenkins security we will prevent anonymous user from doing anything. Only authenticated user will be able to do everything.

For small and medium teams, we will use Jenkin's Own User database. The [official documentation](#) explains the steps clearly.

To disallow anonymous users from accessing any information on Jenkins, we will use [Matrix-based Security](#)

- Go to Jenkins Dashboard, usually `http://server_domain:8080`.
- Select *Manage Jenkins* then *Configure Global Security*.
- Click *Enable Security*.
- Select *Jenkins' own user database*.
- In *Authorization* section, select *Logged-in users can do anything*.
- Make sure you click *Save*
- You'll be promoted to create your first user. Remember the username and password.

4.4.5 Jenkins Git Configuration

By default, Jenkins doesn't come with Git plugin. You'll need to find and install the `Git Plugin`, configure Jenkins ssh authorization to access the *bare Git repo*.

- Go To `http://_server_:8080/pluginManager/available`
- Search for `git`
- Select and Install `GIT plugin`
- Restart Jenkins
- Go to `JENKINS_HOME` directory (defaults to `/var/lib/jenkins`).
- Change user to jenkins: `su jenkins`
- Generate ssh keys: `ssh-keygen`
- Make sure that the generated key is found at `$JENKINS_HOME/.ssh/id_rsa`
- The `.ssh` folder has permission `700` and the `id_rsa` file has permission `600`
- Append file `$JENKINS_HOME/.ssh/id_rsa.pub` to git user `authorized_keys` file

```
cat /var/lib/jenkins/.ssh/id_ras.pub >> /home/git/.ssh/authorized_keys
```

- When creating a new workspace, make sure that the authorization is using jenkins private key file.

Jenkins Git Plugin and CentOS 6.6

Jenkins Git Plugin requires git of version `1.7.9` minimum, `1.8.x` recommended. The default git version on CentOS 6.6 is `1.7.1`.

As a workaround, you can use `JGit` a Java git implementation that is bundles with `Git Client` a Jenkins plugin dependency for *Git Plugin*.

4.5 EasyApache mod_proxy_wstunnel

Install `mod_proxy_wstunnel` on EasyApache.

4.5.1 System

- CentOS 6.7 (64 bit)
- cPanel/WHM with EasyApache4
- *root* ssh access

4.5.2 Resources

- https://features.cpanel.net/topic/add-mod_proxy_wstunnel-to-easyapache
- <https://documentation.cpanel.net/display/EA4/EasyApache+4+%28EA4%29+-+How-To's>
- https://httpd.apache.org/docs/2.4/mod/mod_proxy_wstunnel.html
- An EasyApache 3 option using Apache Extension tools *apxs* <https://forums.cpanel.net/threads/limiting-bandwidth-per-ip.136477/#post586877>

4.5.3 General Steps

- Switch to EasyApache 4 if the system is running EasyApache 3 `/scripts/migrate_ea3_to_ea4 --run`
- Create new profile by copying a file from `/etc/cpanel/ea4/profiles/cpanel` to `/etc/cpanel/ea4/profiles/cpanel` (Check [example profile in files](#))
- Install the new profile `ea_install_profile </full/path/to/profile.json> --install`
- From WHM, choose *EasyApache 4* settings, and Select **Provision** for the correct profile.
- Wait until Easy Apache configure the profile and review the logs after the profile is provisioned.

4.6 Dev Server Virtual Machines

4.6.1 Setup

Dev server has `vmware` software to run *Virtual Machines*. There

are currently two virtual machines: `| * Win12SQL * win7-Photoshop`

All Virtual Machines are located at `/home/Vm's/`. Each VM has two `vmnet` network interfaces to access both sub-networks.

4.6.2 Auto-start

It was requested that both VM's be available after booting the server. This was achieved using `/etc/rc.local` script.

A version of the `rc` script can be found on this wiki project files as `rc.local` on project root.

4.6.3 VMWare cmd

To manage vmware VM's from command line, use `vmrun` command. [Online help for the command](#)

4.7 Build PHP 5.2 on CentOS 6

4.7.1 Notice

This guide can easily destroy your CentOS system and definitely not

secure. You'll downgrade some packages and install insecure version of PHP. | **Use virtual box and only for testing purposes.**

4.7.2 Prerequisite

- CentOS 6.7
- `git`
- `root` access
- Good internet connection

4.7.3 General Steps

1. Download [phpenv](#) in your home directory using `git clone https://github.com/CHH/phpenv.git`
2. Run `sudo ~/phpenv/install.sh && mkdir ~/.phpenv/plugins`
3. Download [php-build](#) using `git clone https://github.com/php-build/php-build.git ~/.phpenv/plugins/php-build`
4. Downgrade libxml2 rpm package `sudo yum downgrade libxml2-2.7.6-1.el6.rfx`
5. Download PHP 5.2 build dependencies `bash sudo yum -y install gcc make gcc-c++ cpp kernel-headers.x86_64 libxml2-devel openssl-devel \ bzip2-devel libjpeg-devel libpng-devel freetype-devel openldap-devel postgresql-devel \ aspell-devel net-snmp-devel libxslt-devel libc-client-devel libicu-devel gmp-devel curl-devel \ libmcrypt-devel unixODBC-devel pcre-devel sqlite-devel db4-devel enchant-devel libXpm-devel \ mysql-devel readline-devel libedit-devel recode-devel libtidy-devel`
6. Build PHP 5.2 using `phpenv phpenv install 5.2.17`

Drupal Gotcha

5.1 Conditional Fields

Conditional Fields Module define dependencies between fields based on their states and values.

Module define to entities for each condition: * **Dependent**: The field that depends on other field * **Dependees**: The controlling field on which other fields depend on its state.

5.1.1 Symptom

Dependent field values not save on form submit.

5.1.2 Bug Resources:

- <https://www.drupal.org/node/2061405>
- <https://www.drupal.org/node/1542706#comment-6773474>

5.1.3 Fix:

In **Values input mode**, instead of using “*Insert value from widget...*”, choose any option of group “*Set of values*”.

5.2 Bootstrap Checkbox Tables

5.2.1 Problem

Drupal Bootstrap theme display messed up table when table only have

check-boxes. For example, try view the permission table in Bootstrap based theme. | This issue can also reproduced with `webform` module. Create a new *webform* content with at least one component of type *Grid*. Add at least 3 options and 3 questions.

Example of Bootstrap Table Issue [|Drupal\Bootstrap\checkbox\table\issue|](#)

5.2.2 Proposed Solution

The problem occurs because Drupal adds class `checkbox` to the `td` and `th` tags. Default bootstrap css rules for `.checkbox` is

```
.radio, .checkbox {
  position: relative;
  display: block;
  min-height: 20px;
  margin-top: 10px;
  margin-bottom: 10px;
}
```

The problem occurs because the two properties `position` and `display`. To solve the probme you can use the following css rule:

```
td.checkbox, th.checkbox {
  position: static;
  display: table-cell;
  text-align: center;
}
```

`text-align` property is used because usually the *checkbox* label is hidden.

Final Result **`!Drupal\Bootstrap\checkbox\table\fixed`**

5.3 Drupal Testing Scripts

Sometimes you want to test a function that needs Drupal to be fully strapped without waiting for certain user action.

We'll use Drush ability to run any script file after bootstrapping Drupal.

```
drush scr migs_donation_test.php --script-path=`pwd`/sites/all/modules/custom/migs_donation/tests
```

5.3.1 Be Careful

- Don't modify database
- You can run any function from command line directly
- For more information about Drush `scr` command: `drush help scr`

5.4 Drupal 8 Profiler Storage

5.4.1 Requirements

- Drupal 8
- Webprofiler (*Part of Devel Module*)

5.4.2 Issue

Error message: “*Unable to store profiler information*”

5.4.3 Possible Solutions

1. Storage Backend

- Go to `/admin/config/development/devel/webprofiler`
- Change *Storage backend* to **File storage**.

5.5 Send Multi part Mail in Drupal 7/8

5.5.1 Resources

- [How to Avoid Spam Filters](#)
- [Reach More People and Improve Your Spam Score](#) .
- [Syntax of boundary](#) .
- [Multipart Content Type](#) .

5.5.2 Why Multi-Part Email is Important

- Most of Emails reported as spam and not deliver to client mail, So there was steps you must follow when you send email to limit spam score and the Preference is use `multipart/alternative` as content type then Separate your mail body in two content types first one is `text/plain` this is for Client Applications that's not Support or Cannot Parse HTML tags and the other is `text/html` for Client Applications that's support HTML but the priority for `text/html` if Client Application able to parse HTML tags .

5.5.3 Implementation in Drupal 7

```
drupal_mail('module_name', 'TEST_SMTP', 'client@example.com', 'en' ,['key' => 'value']);
```

- Implementation in drupal 7 like drupal 8 , the only difference that drupal 7 using ``drupal_mail`` function and in drupal 8 you must get mail manager service to send mails .

5.5.4 Implementation in Drupal 8

```
class DefaultController implements ContainerAwareInterface {

    /** @var MailManager */
    protected $mailManager;

    /* dependency injection */
    public function setContainer(ContainerInterface $container = null) {
        $this->mailManager = $container->get('plugin.manager.mail');
    }

    public function testSendMailAction() {
        $this->mailManager->mail('module_name', 'TEST_SMTP', 'client@example.com', 'en' ,['key' => 'value']);
        return [
            '#type' => 'markup',
            '#markup' => 'This Page In development '
        ];
    }
}
```

```
    ];  
  }  
}  
  
``class DefaultController implements ContainerAwareInterface`` \*  
Implementing ContainerAwareInterface to inject mail service into  
Controller, also you can extend ControllerBase if you need but in  
this case we don't\` need to extend BaseController class.  
  
``$this->mailManager = $container->get('plugin.manager.mail');`` \*  
this line to get mail manager service  
  
``$this->mailManager->mail('module_name', 'TEST_SMTP', 'client@example.com', 'en');``  
\* It's very simple to use, First parameter is your module name to  
call hook\_mail , Second parameter is the key that use in hook\_mail  
to specify content , Third parameter is client mail and you can add  
more than one email by using comma separator (,) between emails ,  
'en' this is the encode of email ,and last parameter for sending  
parameters as associative array that will receive in hook\_mail
```

- Now open your `module_name.module` file and implement `hook_mail` function

```
function module_name_mail($key, &$message, $params) {  
  
  $message['from'] = \Drupal::config('system.site')->get('mail');  
  $Plainttext = "Plaint Text Part";  
  $HTMLpart = "<h1>HTML Part</h1>";  
  switch($key)  
  {  
    case 'TEST_SMTP' :  
      $message['subject'] = 'Test Email from Opensource Wiki';  
      $message['headers'] = array(  
        'MIME-Version' => '1.0',  
        'Content-Type' => "multipart/alternative",  
        'Content-Transfer-Encoding' => '8Bit',  
        'X-Mailer' => 'Drupal'  
      );  
      $message['body'][] = "\r\n--\r\n";  
      $message['body'][] = 'Content-Type: text/plain; charset=utf-8;';  
      $message['body'][] = "$Plainttext";  
      $message['body'][] = "\r\n--\r\n";  
      $message['body'][] = "Content-Type: text/html; charset=utf-8;";  
      $message['body'][] = "$HTMLpart";  
      $message['body'][] = "\r\n-- --\r\n";  
      break;  
  }  
}
```

- This is the important part in your code to send mail .

```
$message['from'] = \Drupal::config('system.site')->get('mail'); * To  
specify the sender email address by this Code and we always use system site email (you can change  
this to any email as you like)
```

```
$message['subject'] = 'Test Email from Opensource Wiki'; * set subject of  
email
```

```
$message['headers'] = array( 'MIME-Version' =>
```

```
'1.0', 'Content-Type' => "multipart/alternative",
'Content-Transfer-Encoding' => '8Bit', 'X-Mailer' => 'Drupal' );
* set headers of email and make sure you are set content type as multipart/alternative

['body'][] = "\r\n--\r\n"; $message['body'][] = 'Content-Type:
text/plain; charset=utf-8;'; $message['body'][] = "$Plainttext";
$message['body'][] = "\r\n--\r\n"; $message['body'][] =
"Content-Type: text/html; charset=utf-8;"; $message['body'][] =
"$HTMLpart"; $message['body'][] = "\r\n-- --\r\n"; * You must add bound-
ary code before and after each content type but it is not required to do that if you are using
PHP_Mailer Library this part to set plain text and HTML tags (syntax is very important check
Resources)

$message['params']['attachments'][] = array( 'filecontent'
=> file_get_contents(PATH_OF_FILE), 'filename' =>
'file_name.extention', 'filemime' => 'application/type', ); * To
attach files use this code
```

NOTES:

- Some library or PHP native needs to encode your file after using `file_get_contents` function , check your library requirements.
- Email Client Applications use `base64_decode` to decode attachment as default .
- Download and enable [SMTP](#) Module then enter your SMTP configuration from `http://host/config/system/smtp`.
- Change mailing system by this command `drush cset system.mail interface.default SMTPMailSystem`.

Front-end Patterns

6.1 Sticky Footer

6.1.1 Problem Statement

How to pin footer to be fixed to bottom of browser window without min-height for content cell

6.1.2 Sample CSS

```
html {  
  position: relative;  
  min-height: 100%;  
}  
body {  
  /* Margin bottom by footer height */  
  margin-bottom: 60px;  
}  
.footer {  
  position: absolute;  
  bottom: 0;  
  width: 100%;  
  /* Set the fixed height of the footer here */  
  height: 60px;  
}
```

6.1.3 Common Issues

- Footer must be absolute from `<html>`. Avoid any parents to `.footer` with position relative.
- If you don't need to put a fixed height to your footer, you need to change "margin-bottom" at the `body` in the different media queries in your css

6.1.4 Resources

<https://getbootstrap.com/examples/sticky-footer/>

6.2 Patterns for Website Banners

6.2.1 Purpose

A common request in modern Web sites to have some sort of graphical banner, usually with some text inside. These are most important guide lines for banners. These patterns are specially important for displaying the banner image on different zoom levels or screens and always displaying the main content.

6.2.2 Assumptions

- The banner image is added as `background-image` property. Not `` tag.
- The design is flexible or responsive.

6.2.3 Banner Images

There are 3 important guide lines regarding image specifications: 1. Image Size 2. Image Aspect Ratio 3. Image CSS properties

1. Image Size

Focus on image **width**. It must match the maximum screen size

acceptable by client. | Usually *1600 px* or *1900 px* are enough.

We will use `background-size: cover` property to always make sure that the image width is 100% of the screen.

2. Image Aspect Ratio

Aspect ration affects image height. You should calculate the aspect ration for image main content to be visible.

For Example: Image size is *1600 x 900 px*. So the aspect ratio is *16:9* which is excellent for high definition images.

The banner image is usually visible through *viewport*, a fixed width

and height `<div>` with the banner as background image. | To determine the height of the *viewport*, you can calculate it based on the *viewport* width.

Height Calculation

Don't use pixels `px` for height unit.

On default zoom level, *1 screen* pixel matches *1 css* pixel. On

lower zoom level, *1 screen* pixel matches *more than 1 css* pixel. For example: (50% Zoom level, 1 screen pixel = 2 css pixel) | On higher zoom level, *1 css* pixel matches *more than 1 screen* pixel. For example: (200% Zoom level, 2 screen pixel = 1 css pixel)

Height Alternative

Instead of using `height` css property, we will use `padding` on parent *viewport*! `padding` property when given value in percentage %, it's calculated as percentage of the parent container!! (*Intereseting CSS Feature*) To calculate the required padding, use the following equation:

$$\text{Padding (Top OR Bottom)} = 100\% \times (\text{aspect/ratio})^{-1}$$

Example: * Aspect Ratio = **16:9** => padding-bottom = $100\% \times 9/16 = 56.25\%$ * Aspect Ratio = **4:3** => padding-bottom = $100\% \times 3/4 = 75\%$

3. Image CSS properties

Image is used as `background-image` property. Image is centered and stretched.

```
<div class="viewport">
  <div class="banner-img"></div>
</div>
```

```
.viewport {
  position: relative;
  padding-bottom: 56.25%; /* (100% x 9/16) */
  height: 0; /* Only height
}
.banner-img {
  position: absolute;
  left: 0;
  right: 0;
  height: 100%;
  background-position: center center;
  background-repeat: no-repeat;
  background-color: transparent;
  background-size: cover;

  /* Dont't forget background-image! */
}
```

6.2.4 Example

- See `BannerPatterns.html` in project files.

6.2.5 Special Cases

- Notice if you need to include text inside the banner image, it will change in size as you zoom in and out. And it might overflow the *viewport*.
- If you have scallable content above the *viewport*, the *viewport* will move up and down as you zoom in and out.
- For really small screens, you can move the content inside the banner image outside. (For example, remove the position absolute for inside content)

Drupal 8 Reference

Drupal 8 Render pipeline represent *Drupal 8 Life Cycle*

Attached resource represent how Drupal 8 handle requests and different

options for rendering the request response. | The graphic in the resource link can be found in PDF format.

Resource: <https://www.drupal.org/developing/api/8/render/pipeline>

7.1 PDF Starter

You can find PDF which was mentioned in this URL: [Drupal8 LifeCycle Export](#)

7.2 Drupal 8: Installation Guide

This guide will describe steps and hacks for installing Drupal 8 using Composer, Drush and standard MySQL cli client.

7.2.1 Prerequisite

- [PHP v5.3.2+](#)
- [Composer](#)
- Your PATH shell variable includes the path to Composer binaries at `$HOME/.composer/vendor/bin`
- [MySQL 5.1+](#)

7.2.2 Resources

- [Drush 8 installation using Composer](#)
- [Composer timeout](#)
- [MySQL Collation](#)

7.2.3 Goal

Install latest [Drupal 8](#) instance using Drush installed through Composer. Drupal will use MySQL database.

7.2.4 Installation Steps

Each installation step comes with its own command lines examples and notes for more help.

1. Install Drush

To bootstrap Drupal 8, we need Drush 8, which isn't stable by the time of writing this guide.

If you are installing Drush for the first time, use the following command:

```
export COMPOSER_PROCESS_TIMEOUT=600 # default 300
sudo composer self-update # Make sure that composer is updated
composer global update # Update all global packages to prevent conflict in dependency versions
composer global require --prefer-dist drush/drush:dev-master
```

NOTES: * The export command will prevent composer from running into timeout error while install *psy/psysh* package. * We are using composer `--prefer-dist` flag to prevent composer from downloading the source code of every required package. * We are not using `drush/drush:8.*` package because drush 8 is not stable by the time of writing this guide. * Composer will not install any package without stable version by default according to: https://groups.google.com/d/topic/composer-dev/_g3ASelFrc/discussion

If you already has drush installed, or you want to change the installed version use the following command:

```
composer global require drush/drush:7.*
# OR
composer gloabl require drush/drush:dev-master
```

To test drush version:

```
drush --version
# output should be something like:
# Drush Version : 8.0.0-rc3
```

NOTES: * Drush 8 requires special xdebug configuration, if you don't have *xdebug* php extension installed, ignore the following commands:

```
sudo vi /etc/php5/cli/php.ini # Or find the used PHP configuration file by running: `drush status` c
# You are now in `vi` command mode, type the following sequence of characters to move in `vi`
# Go
# You are now in `vi` edit mode, just add the followin lines to the end of file
[xdebug]
xdebug.max_nesting_level=256
# Hit [Esc] button and type
# :x
```

2. Download Drupal

We will use drush to download Drupal 8 instance. By default, drush will download Drupal to a folder in the current directory with schema `drupal-x-x-x`

```
drush dl drupal-8 --select --all --drupal-project-rename
```

NOTES: * We use `--drupal-project-rename` to rename the project

folder name, which defaults to “*drupal*”. You can rename the folder any name you like by providing the new name after the `--drupal-project-rename` switch. | For more help: `drush help dl` * We use `--select --all` because Drupal 8 doesn’t have a stable version by the time of writing this guide.

Now, if used the previous command as is, you have a new folder called “*drupal*” in the current directory. Go ahead and check the downloaded files.

3. Install Drupal

We will use MySQL database for Drupal installation. For this we will connect to mysql server using mysql cli client with the root username and password.

For demonstration will assume the following mysql setup: * ‘root’ password: root * mysql hostname: localhost * mysql port: *DEFAULT* * Drupal mysql user: drupal (*User already exists in database*) * Drupal mysql pass: drupalpass

```
mysql -u root -proot -e 'CREATE DATABASE drupal8 COLLATE = "utf8_general_ci"; GRANT ALL ON drupal8.*
```

NOTES: * If you want to create the mysql user “drupal” add the following code before the last semicolon: `IDENTIFIED BY PASSWORD "drupalpass"`

Now go to the Drupal folder and use `drush si` command.

For demonstration will assume the following setup: * Current Working Directory: . * Drupal files root: `./drupal` * Drupal Super Admin name: `sadmin` * Drupal Super Admin pass: `sadminpass` * Drupal Super Admin mail: `sadmin@example.com`

```
cd drupal
drush si minimal --account-name=sadmin --account-pass=sadminpass --account-mail=sadmin@example.com --
```

NOTES: * You can add additional switches to the `site-install` command like `--site-mail` and `--site-name`. For more help: `drush help si`

4. Run Drupal

You have two options to run your Drupal site now: * HTTP Server (Apache Virtual Host/Nginx) * Drush `runserver`

There are plenty of resources online to help you setup HTTP Server. It requires few configurations but this is usually how your site will run in production environment. For development environment the second option requires no setup because it uses PHP’s built-in http server.

```
drush rs
```

NOTES: * The output will indicate the exact URL to access your Drupal site. (*Defaults to: http://127.0.0.1:8888*) * The `minimal` profile uses *Stark* theme, so you’ll properly need to change to more visual theme.

7.3 How to Dependency Injection in drupal ?

we are going to explain how dependency injection with twig as example .

7.3.1 Resources

- [Late Static Binding](#) .
- [ClassResolver](#) .
- [getInstanceFromDefinition](#) .

Dependency Injection

In Dependency Injection we have over three solutions to load and render TWIG as example

1. By implement `ContainerInjectionInterface` interface in your controller

implement this interface you must add create method like this

```
class CustomController implements ContainerInjectionInterface {  
  
    protected $twig ;  
  
    public function __construct(\Twig_Environment $twig)  
    {  
        $this->twig = $twig ;  
    }  
  
    public static function create(ContainerInterface $container)  
    {  
        return new static(  
            $container->get('twig')  
        );  
    }  
  
}  
  
``public static function create(ContainerInterface $container)``
```

- make sure your function is public and static because it called without instance of your controller

```
return new static();
```

- this new concept called (late static binding LSP) start with php 5.3+ you will find info [here](#)
- more info [self vs static](#)

```
$container->get('twig')
```

- this load twig container and send to constructor as parameter .

```
protected $twig ;
```

- declare a variable , you can avoid this step and use magic function as you like .

```
public function __construct(\Twig_Environment $twig)
```

- in constructor must receive twig as parameter and assign to protected twig variable .

NOTES:

- The responsible class about call your create function is [ClassResolver](#) and method is [getInstanceFromDefinition](#)

2. By implement ContainerAwareInterface interface in your controller

- After implement this interface you must add (setContainer) method like this .

```
class CustomController implements ContainerAwareInterface {  
  
    protected $twig ;  
  
    public function setContainer(ContainerInterface $container = null)  
    {  
        $this->twig = $container->get('twig') ;  
    }  
  
}
```

`$this->twig = $container->get('twig') ;` to assign twig container to twig variable

NOTES:

- make sure you are implement (ContainerAwareInterface)[#]
- The responsible class about call your create function is [ClassResolver](#) and method is [getInstanceFromDefinition](#)

3. By get container directly (*not recommended*)

```
$twig = \Drupal::getContainer()->get('twig') ;
```

7.3.2 Common Questions

- You should make sure that your logging and errors is enable “ `yoursite/admin/config/development/logging` ” .
-

7.4 Drupal 8: Menu Links in Admin Config

Drupal 8 `admin/config` is the place to manage all site

configuration. | Adding custom links under existing sections is easy. | Adding custom sections is not so easy for beginners.

7.4.1 Challenge

Add new custom section in `admin/config` and add several links in this new section.

7.4.2 Live Example

This example is taken from Drupal 8 installed instance. Because we have the source code of Drupal 8 in plain text, we can examine how Drupal used its own system to include sections on links in the `admin/config` page. You can examine how *Media* section is implemented in the following files: `* core/modules/system/system.router.yml *`

`core/modules/system/system.links.menu.yml` You'll find other sections in the same files. And even the implementation of `admin/config` route itself.

7.4.3 Assumptions

- All configurations are part of a custom module *myModule* .
- You have custom route, that you want to add under new section.
- Route path: `/admin/config/myModule/config`
- Section
 - Title: `myModule Configuration`
 - path: `/admin/config/myModule`

7.4.4 Abstract Guideline:

1. Create at least 2 routes, one for the section and any other route to be added as links in this section.
2. Create at least 2 links, one for the section and any other menu link to be added to this section.

7.4.5 Solution

Keep in mind the following information about your module. * **Module Path:** `modules/custom/myModule`
* **Router File:** `modules/custom/myModule/myModule.router.yml` * **Links File :** `modules/custom/myModule/myModule.links.menu.yml` Here is the sample code to for the routes.

Router Files

Notes

* Route names are arbitrary; they can be anything. *

myModule.settings route can be any route with **any** configuration, there are no mandatory properties. * *myModule.adminconfig_* route can have any title, use other properties as is. You can test other path values but at least not `_controller` property!

Links Menu Files

Notes * Menu links names use the same router name for consistency. Is it a must? * You can include any other configuration link under the new section if you use: `parent: myModule.admin_config`

Cache Rebuild

Rebuild your cache for the new section and links to be visible.

7.4.6 Common Problems

The section might not appear after clearing the cache, check the following points: * Spelling of the all router names and menu link names * The section must have at least on link for it to appear in `admin/config` page * In `links.menu.yml` file, use link names in parent

7.4.7 To Do

1. Test other properties that may affect the section menu link like `weight` and `position`

7.5 Drupal 8: Content Entities

Content entities are used to store information in database that can be translated and revisioned.

7.5.1 Getting Started

NOTE: Make sure you are working on local git repo, to revert back any unwanted changes, or examining generated codes

1. Use *drupal console* to generate some boilerplate code to get things started.

```
$ drupal generate:entity:content
```

- The *console* will ask several questions about the new content entity, like: the module this entity belongs to, Label and machine_name.
 - Use *console help* to find what arguments this command accepts.
 - Revert back all changed files, and remove added one, and repeat the `generate:entity:content` command using `--learning` option.
 - Examin generated files and changes and try to guess what they do.
2. Until *drupal v0.9.8* the generated code have some issues solve them:
 - Forms: Find the generated form files and fix them; rename and move them according to their content
 - Routes: Find the two routes that have the same path, change on of them. Make sure that changed path has no other dependency paths.
 - Links: Use [Drupal 8 Config Menu Links wiki](#) to create a new configuration links block at `admin/config` page, and the links to main entity routes there.
 3. Fields There are two ways to add fields to entities, use both of them to add different fields:
 - (a) `ContentEntityBase::baseFieldDefinitions()`
 - (b) Using Field UI
 4. Field Validation Use [Entity Validation API in Drupal 8](#) to add constraint to field item properties. Try adding the following validations:
 - (a) *Email*
 - (b) *Unique*
 - (c) *Not Blank!*
 5. Interfaces Use the *intreface* class to add setters and getters for this content type.

6. Install your entity Schema If your module was already installed, clear the cache. If it wasn't installed yet, install it now to install the entity database schema. Examin the create database table.
7. Entity Listing: Try adding few entity instances using the *create route* to find the form and examin the listing, there is no entity view link in the listing table. Find the listing class and add columns to show one or two extra fields and make the name column links to entity view URL

7.5.2 Entity Challenges

Working with content entity is not a smooth joureny, there are some challenges that you need to be aware of.

- In Drupal 8 there is no module disable. You can either install or uninstall a modules. You can not uninstall a module with content entity, if the content entity table is not empty, or doesn't exist. Try it!
- Content Entity database schema updates implemented at `ContentEntityBase::baseFieldDefinitions()` are applied on only two events (for now):
 - *Cache Rebuild* if the module was enabled and you created the entity class
 - *Module Install* will apply the updates, which means:
 - If there was a problem in the `ContentEntityBase::baseFieldDefinitions()` method that prevented the entity table from being created, you'll have to create it first by hand and then you can uninstall your module! Use simple command like `$ echo 'CREATE TABLE entity_machine_name (id int);' | drush sqlc`
- Try to add any validation to fields created throught *Field UI* using only Drupal core, it is not that simple if not impossible for now!
- Uninstall and re-install your module, where are the attached entities?! Use configuration export to persist them.

7.6 Twig Debug

7.6.1 Purpose

During Twig development, we need to enable Twig debugging to see what are the template suggestions and which files to override.

7.6.2 Outline

- Enable local settings
- Use development service yml file
- Override Twig parameters

7.6.3 Resources

- [Debugging compiled Twig templates](#)

7.6.4 Guide

1. Copy `sites/example.settings.local.php` to `sites/default/settings.local.php`
2. Edit `sites/default/settings.local.php` line 92 to change `DRUPAL_ROOT . '/sites/development.services.yml` to `__DIR__ . '/../development.services.yml`
3. Uncomment the files that include the `settings.local.php` in `settings.php` php if `(file_exists(__DIR__ . '/settings.local.php')) { include __DIR__ . '/settings.local.php'; }`
4. Edit `sites/development.service.yml` to override Twig parameters. Add the following lines with spaces to the end of the file. `yml` parameters: `twig.config: debug: true`

7.7 Drupal Twig Extension

Drupal provide an extension class for Twig to expose Drupal services and functionality as Twig functions and filters.

7.7.1 Resources

- `class TwigExtension`
- `TwigExtension::getFilters()`
- `TwigExtension::getFunctions()`

7.7.2 Highlights

- *Filters*
- `clean_id`: Make sure that the html ID is unique.
- `t` or `trans`: Translate string using Drupal `t` function.

Goal

Review Drupal 8 components and structure. Run over most important Drupal 8 concepts and components needed to bootstrap our department knowledge.

Perquisite

- PHP OOP

Plan

You can find the plan sheet at [Drupal 8 Plan](#)

Indices and tables

- `genindex`
- `modindex`
- `search`