#### Plug-in development

**Bring your soundblogChild folder down from GitHub**

* git clone (url from GitHub clone button - your soundblogChild)
* set up MAMP
* import your database

**Plugins update**

* Update your plugins (if necessary)
* git status, add, commit (if necessary)

**Create a new branch in Git**

* git branch ‘plugin’
* git checkout branch plugin
* git branch to check

**Create a plugin**

* cd into plugins folder
* touch ‘concerts-custom-post.php’
* open in VSC: add plugin info and licensing
  + <https://developer.wordpress.org/plugins/the-basics/including-a-software-license/>
  + update with your info
* git push --set-upstream origin plugin
* check out branches on GitHub
* activate the plugin in your WordPress dashboard
* Create a function and use an action hook
* <https://codex.wordpress.org/Function_Reference>

function my\_custom\_posttypes() {

}

add\_action ( 'init', 'my\_custom\_posttypes' );

* Use the register\_post\_type WordPress function inside your my\_custom\_posttypes function
* <https://codex.wordpress.org/Function_Reference/register_post_type>

$args = array(

'public' => true,

'label' => 'Concerts'

);

register\_post\_type( 'concerts', $args);

* Grab the code from Flushing Rewrite on Activation so that WordPress can create new slugs for your custom post types (this will overwrite default slugs and create new ones)
* <https://codex.wordpress.org/Function_Reference/register_post_type#Flushing_Rewrite_on_Activation>

function my\_rewrite\_flush() {  
 // First, we "add" the custom post type via the above written function.  
 // Note: "add" is written with quotes, as CPTs don't get added to the DB,  
 // They are only referenced in the post\_type column with a post entry,   
 // when you add a post of this CPT.  
 my\_cpt\_init();  
  
 // ATTENTION: This is \*only\* done during plugin activation hook in this example!  
 // You should \*NEVER EVER\* do this on every page load!!  
 flush\_rewrite\_rules();  
}  
register\_activation\_hook( \_\_FILE\_\_, 'my\_rewrite\_flush' );

* Change my\_cpt\_init(); to my\_custom\_posttypes();
* Check your Concerts post type in the dashboard, then replace your $args array with this code

$labels = array(

'name' => 'Concerts ',

'singular\_name' => 'Concert',

'menu\_name' => 'Concerts',

'name\_admin\_bar' => 'Concerts',

'add\_new' => 'Add New',

'add\_new\_item' => 'Add New Concert',

'new\_item' => 'New Concert',

'edit\_item' => 'Edit Concert',

'view\_item' => 'View Concert',

'all\_items' => 'All Concerts',

'search\_items' => 'Search Concerts',

'parent\_item\_colon' => 'Parent Concerts:',

'not\_found' => 'No Concerts found.',

'not\_found\_in\_trash' => 'No Concerts found in Trash.',

);

$args = array(

'labels' => $labels,

'public' => true,

'publicly\_queryable' => true,

'show\_ui' => true,

'show\_in\_menu' => true,

'query\_var' => true,

'rewrite' => array( 'slug' => 'concerts' ),

'capability\_type' => 'post',

'has\_archive' => true,

'hierarchical' => false,

'menu\_position' => 5,

'supports' => array( 'title', 'editor', 'thumbnail' )

);

* Check your Concerts post type in the dashboard to see the difference
* Find a dash icon: <https://developer.wordpress.org/resource/dashicons/#marker>
  + Add the code below ‘show\_in\_menu’

'menu\_icon' => 'dashicons-format-gallery',

* These arguments determine whether the post will be a post or a page:

'capability\_type' => 'post',

'has\_archive' => true,

'hierarchical' => false,

* This arguments sets what can be edited in the Editor

'supports' => array( 'title', 'editor', 'thumbnail' )

* + <https://codex.wordpress.org/Function_Reference/post_type_supports>

**Test your concerts post types**

Create four new concerts post types: (all images in IMG 210>Shared>Portfolio concert images)

* Suite: Live in Liverpool
  + Philip Jeck (UK), recipient of an Award of Distinction for "Suite: Live in Liverpool", amazes those who behold his sophisticated utilization of old record albums and turntables.

Photo showing his performance at Digital Musics & Sound Art in Concert.

* + Image: Suite
* Energy Field
  + Jana Winderen (NO), honored with the Golden Nica for "Energy Field", obtains her sound material in the wild. She recorded all of it in the depths of the Barents Sea and amidst the fjords and glacial landscapes of northern Norway and Russia.

Photo showing her performance at Digital Musics & Sound Art in Concert.

Image: Energy-field

* Mucilaginous Omniverse
  + "Mucilaginous Omniverse" by Dimitry Gelfand (RU) and Evelina Domnitch (BY) works acoustically with a standing wave that forms out of oil droplets above a water surface being radiated with sound. They got an Honorary Mention at the Prix Ars Electronica 2011 for their work.
  + Image: Omniverse
* Open Reel Ensemble
  + For this performance worth seeing and hearing, Japan's Open Reel Ensemble (JP) has converted an analog recording device into a sound-producing instrument, and plays on (or with) magnetic tape machines salvaged from technological retirement.
  + Image: Reel-ensemble

**Create a page template for your concerts post type**

* In your Twenty-seventeen child theme, copy single.php and rename it single-concerts.php
* Test to see if it works
  + under get\_header(); type ‘test test test’
  + go to a concerts posts, refresh the browser and see if the text appears
  + check a normal post (like World Soundscape Project) and notice that it does not have the ‘test test test’ text
  + remove the ‘test test test’ text and save
* Copy the content.php from the Twentyseventeen parent theme
  + Rename it content-concerts.php
  + Add the ‘test test test’ text directly beneath <header class=”entry-header”>
* In single-concerts.php change the get\_template\_part arguments to

get\_template\_part( 'content', 'concerts', get\_post\_format() );

* Test to see if it works
  + go to a concert post, refresh the browser and see if the text appears
  + remove the ‘test test test’ text

**Add a class to style your new template page**

* In content-concerts.php add the class “concerts” to the “entry-content” class

<div class="entry-content concerts">

* In style.css

.concerts p {color: blue;}

* Test to see if it works - the paragraph text on a concerts post should be blue

**Remove the next and previous buttons on a concerts post**

* In single-concerts.php comment out the\_post\_navigation code
* Test to see if it works - there should no longer be next and previous links at the bottom of a concerts post

**Create an index page for your concerts posts**

* By default, WordPress will use archive.php as your index page for your concerts posts
  + localhost:8888/concerts - notice the template file is archive.php
* Copy archive.php from Twentyseventeen parent theme into your child theme and rename it archive-concerts.php
* Comment out the\_archive\_title and add an h1 title:

<?php

//the\_archive\_title( '<h1 class="page-title">', '</h1>' );

//the\_archive\_description( '<div class="taxonomy-description">', '</div>' );

?>

<h1 class="page-title">Concerts:</h1>

**Including custom post types on the site**

* Menus>Top Menu>Custom Links
  + URL: <http://localhost:8888/concerts/> (you will need to update this after migrating)
  + Link Text: Concerts

**Merging your git branches**

<https://www.atlassian.com/git/tutorials/using-branches/git-merge>

* make sure all of your changes to branch plugin have been committed and pushed
* git checkout master
  + note that your newly made files are gone and that on the dashboard of your soundblog site there is no longer an option for Concerts posts
* git checkout plugin
  + notice that the files and Concert posts have returned
* git checkout master
* git merge plugin (this will merge plugin with master\_
* if you have any merge conflicts, resolve them: <https://css-tricks.com/deal-merge-conflicts-git/>
* git push --delete origin plugin (removes the plugin branch on GitHub)
* check on GitHub that the plugin branch is no longer there

**Stepping back in Git**

* In styles.css, change the concerts p colour to red
* Do a git commit
* git log
  + copy the first 9 numbers of your penultimate (second to last) commit - make sure you are choosing the correct SHA! It will be the second one at the top
  + ‘q’ to get out of git log
* git checkout (copied numbers)
  + notice that your concerts p colour is blue again
  + git branch - notice that your HEAD is detached from the master branch
* git checkout master
* git reset --hard (9 numbers ie. 071a70d4f)
* git log
  + notice that HEAD is now on 071a70d4f SHA and there is no newer commit
* git push --force origin (to force your HEAD reset to GitHub)

Export your database and store the localhost.sql file in your soundblogChild root folder.

* You will have to remove the old one and do a commit before adding the new one

Push to GitHub.