

Project Phase 02

Objectives

- Use PHP headers
- Use cURL (Client URL)
- Add output buffering
- Create utility functions for get and post requests
- Create code for a new record
- Create code to edit a record

Getting Started

Read all of the instructions before starting to code.

Start this phase by using the code you created for phase01 or using the phase02 starter files in Moodle. You can copy the starter files into your **sas** folder. Git will recognize that these are new files. You will need to stage and commit them. Here is a [video on copying the files into your sas folder](#) and using Git.

First you will need to navigate to your **sas** folder. We previously moved the **sas** folder out of our **web250** folder so it can have its own version of git.

File structure

This is the file structure you will work with for this assignment.

```
├─ phase02
|   ├─ index.php
|   ├─ private
|   |   ├─ functions.php
|   |   ├─ initialize.php
|   |   └─ shared
|   |       ├─ salamander-footer.php
|   |       └─ salamander-header.php
|   └─ public
|       ├─ images
|       ├─ index.php
|       ├─ salamanders
|       |   ├─ create.php
|       |   ├─ edit.php
|       |   ├─ index.php
|       |   ├─ new.php
|       |   └─ show.php
|       └─ stylesheets
|           └─ salamanders.css
```

[Win] `sh cd c:/xampp/htdocs/sas`

[Mac]

```
cd /Applications/MAMP/htdocs/sas
```

Here is list of the git commands to use

```
git status
git add .
git commit -m"starting phase 2 of the project."
```

You can use these commands whenever you are at a point where you want to save your project using Git. Usually, I run them after I've finished a chunk of code I want to make sure I've saved.

What to Watch

We will continue using the tutorial, [PHP with MySQL: Essential Training 1.](#)

- Chapter 3: Headers and Redirects

- Chapter 4: Build Forms with PHP
- [Video describing how phase02 looks in the browser.](#)

Chapter 3: Headers and Redirects

This chapter shows how PHP uses file headers. At first this may seem a bit dry, but knowing how to redirect files and use output buffering is an essential part of PHP programming.

- Add the functions, error404 and error500.
- Create a file in your **salamanders** folder called **new.php**.
- Add the reference to **initialize.php** and place it at the top of **new.php**.
- Test your code. You should receive "No error".
- Go along with Kevin and try using cURL (Client URL) and run the `curl` command. Here is an example of my code.
- Take a screenshot of the output after you run it and save the file as `curl.png`.

```
> curl --head http://localhost:8888/web182/sas/public/salamanders/new.phpcu
HTTP/1.1 200 OKHTTP/1.1 200 OK
Date: Mon, 21 Mar 2022 15:13:12 GMTDate: Mon, 21 Mar 2022 15:13:12 GMT
Server: Apache/2.4.46Server: Apache/2.4.46 ((UnixUnix)) OpenSSL/1.0.2u PHP/
X-Powered-By: PHP/8.0.8X-Powered-By: PHP/8.0.8
Content-Type: text/htmlContent-Type: text/html;; charsetcharset=UTF-8
```

Now do the same thing for errors 404 and 500.

I could not get mine to work with 404 or 500.

redirect_to

Create the `redirect_to` function in your **functions.php** file.

Output buffering

Add output buffering to your **initialize.php** file.

Build Forms with PHP

You will need to create the files, **new.php**, **edit.php** and **create.php** from scratch.

Modify your code so you can pull up the **edit.php** file. Note, you will only be able to see the **id** in

the URL. No data is present in the form itself.

Use form parameters

Use the code in **create.php** and modify it for salamanders. The only value we are passing by the `$_POST` superglobal is the salamander name.

form action

Modify the form action in **new.php** so it references create.php. Use the `url_for` utility function.

Detect form submission

Add these two functions to your **functions.php** file.

```
is_post_request()  
is_get_request
```

Modify your **create.php** file so it uses the two functions you just created.

Single Page Form Processing

This is a new concept for our class and it breaks one of the rules called "separation of concerns"; however it also makes coding easier since everything is in one place. It also shows how you can use GET and POST together.

Final Checklist

These are the files you should complete while working through the assignment.

Add the following functions to the **functions.php** file

- error_404
- error_500
- `redirect_to($location)`
- `is_post_request()`
- `is_get_request()`

- Add `ob_start()` to **initialize.php**
- Edit the **salamanders/index.php** link
- Add an **sas/index.html** file that lists the two phases. Delete the previous **index.php** file that automatically redirected the user.
- Create the following files in the salamanders folder.
 - new.php
 - create.php
 - edit.php

GitHub

You should have already created a GitHub repo on GitHub's site for the project. Contact your instructor if you are having difficulty with this.

Once you have finished your code, run the staging and committing commands before pushing your code to GitHub.

Note: When I first started learning git, I used the status command a lot so made sure I knew what was happening.

```
git status
git add .
git commit -m"Finished phase 2 of the sas project."
git status
```

Now that everything is ready, it is time to push your code to GitHub.

```
git push
```

Update your webhost

Upload your code to your webhost. The file structure below shows

1. The **index.html** file at the root (top position). This file contains links to phases 1 and 2.
2. The tree shows both phases 1 and 2.

```
├─ index.html
├─ phase01
|   ├─ index.php
|   ├─ private
|   |   ├─ functions.php
|   |   ├─ initialize.php
|   |   └─ shared
|   |       ├─ salamander-footer.php
|   |       └─ salamander-header.php
|   └─ public
|       ├─ images
|       ├─ index.php
|       ├─ salamanders
|       |   ├─ index.php
|       |   └─ show.php
|       └─ stylesheets
|           └─ salamanders.css
├─ phase02
|   ├─ index.php
|   ├─ private
|   |   ├─ functions.php
|   |   ├─ initialize.php
|   |   └─ shared
|   |       ├─ salamander-footer.php
|   |       └─ salamander-header.php
|   └─ public
|       ├─ images
|       ├─ index.php
|       ├─ salamanders
|       |   ├─ create.php
|       |   ├─ edit.php
|       |   ├─ index.php
|       |   ├─ new.php
|       |   └─ show.php
|       └─ stylesheets
|           └─ salamanders.css
```

Test your code on your host and make sure all of your links work.

Submit your work

Create a text file named proj02-yourLastName.txt and paste in your Webhost and your GitHub

addresses and submit it in Moodle.