Оглавление

[Структура проекта 1](#_Toc94014301)

[Описание файла index\_\_with\_\_form.php 3](#_Toc94014302)

[Описание файла /phpQuery\_\_crawling/**get\_\_json.php** 6](#_Toc94014303)

[Описание раздела components\_for\_front 9](#_Toc94014304)

[Файл /phpQuery\_\_crawling/json\_to\_bd.php 10](#_Toc94014305)

# Структура проекта

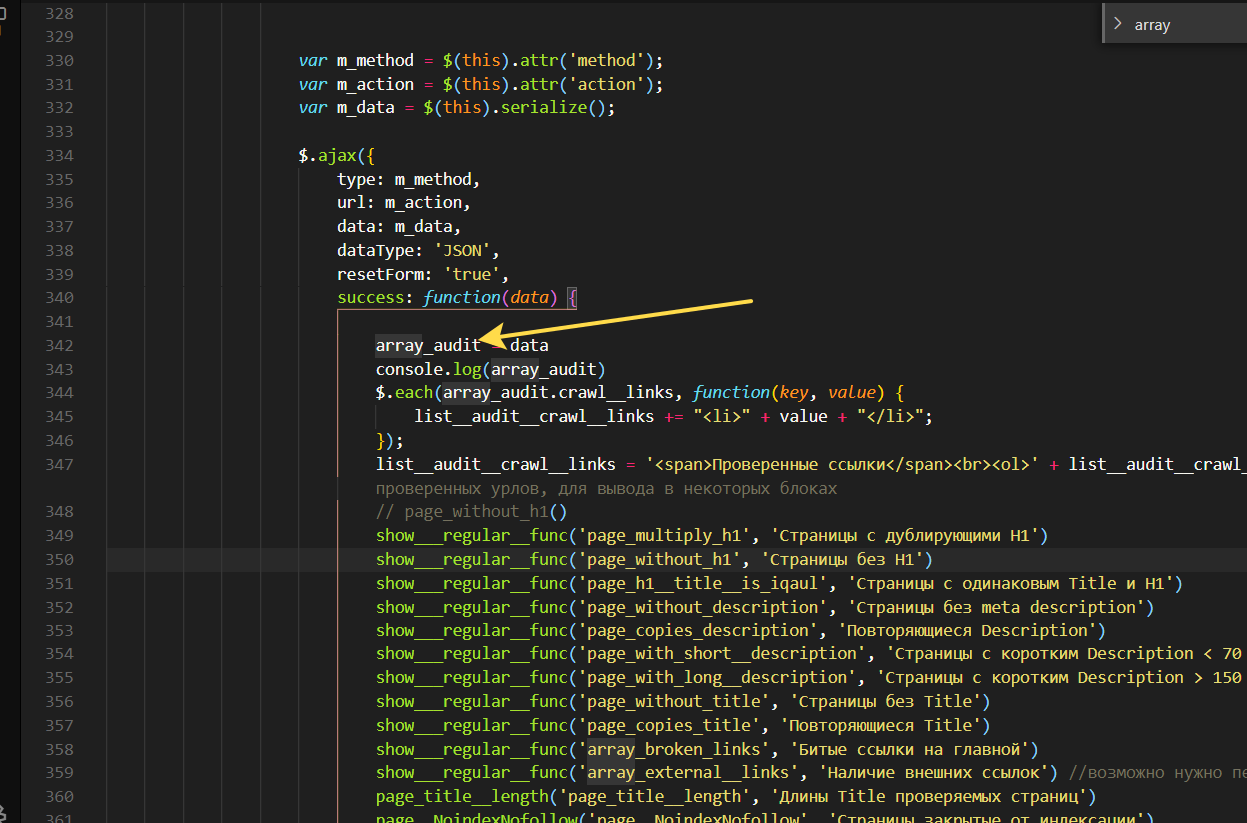
1. Точка входа <http://im-pulse33.ru/phpQuery__crawling/index__with__form.php>

В кратце: там происходит вывод из БД результаты прошлых проверок, форма для запроса\отправки урла проверяемого сайта, после проверки выводятся результаты

1. Первый рабочий файл, к которому идет запрос после ввода урла в форме поиска:

/im-pulse33.ru/public\_html/phpQuery\_\_crawling/get\_\_json.php . В нем собирается основной массив данных, но не весь. На сайте ищутся до 100 ссылок. После происходит переобход этих ссылок и сбор какой-либо инфы

1. После проверки во втором пункте, начинается вывод результата собранного во 2ом пункте
2. Далее сразу после результатов 2ого пункта сразу идет куча паралельных запросов к файлам папки /im-pulse33.ru/public\_html/phpQuery\_\_crawling/components\_for\_front . Каждый файлик выполняет какую-то функцию.

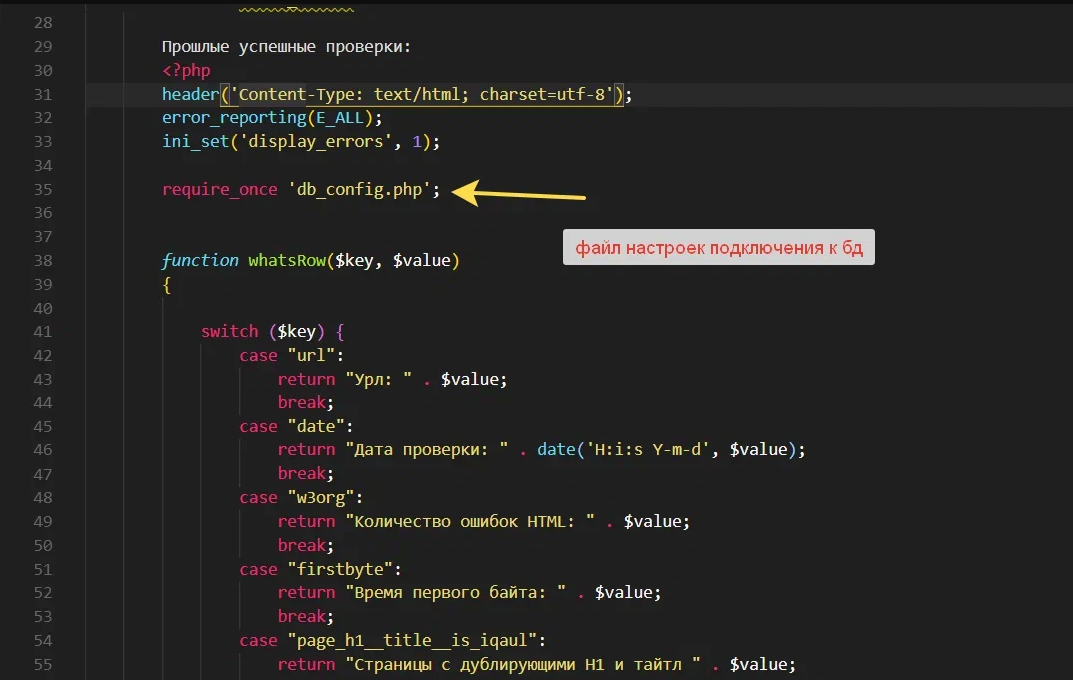
Сразу же предупрежу,Что данные, которые выводятся на после проверки (красные, зеленые, типа прошло проверку или нет – это не ИСТИННА). В процессе работы я много чего менял. Поэтому важно смотреть на выходящий массив, смотреть что в нем за данные. Вот в нем находятся истинные данные **array\_audit[].** Первое формирование этого массива тут: 

Алгоритм работы, описание этапов

# Описание файла index\_\_with\_\_form.php

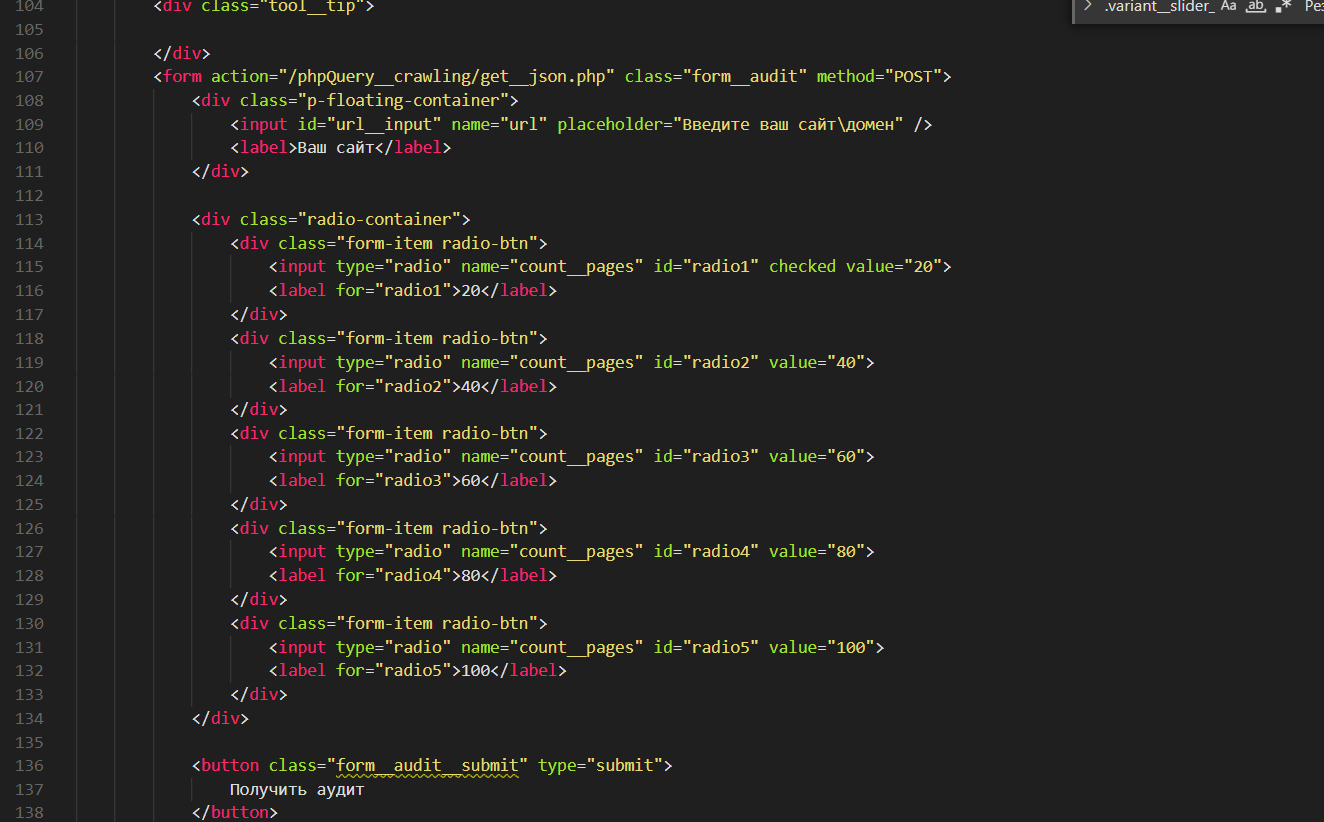
Все начинается с файла public\_html\phpQuery\_\_crawling\**index\_\_with\_\_form.php**

1. Получение данных из записей в бд:

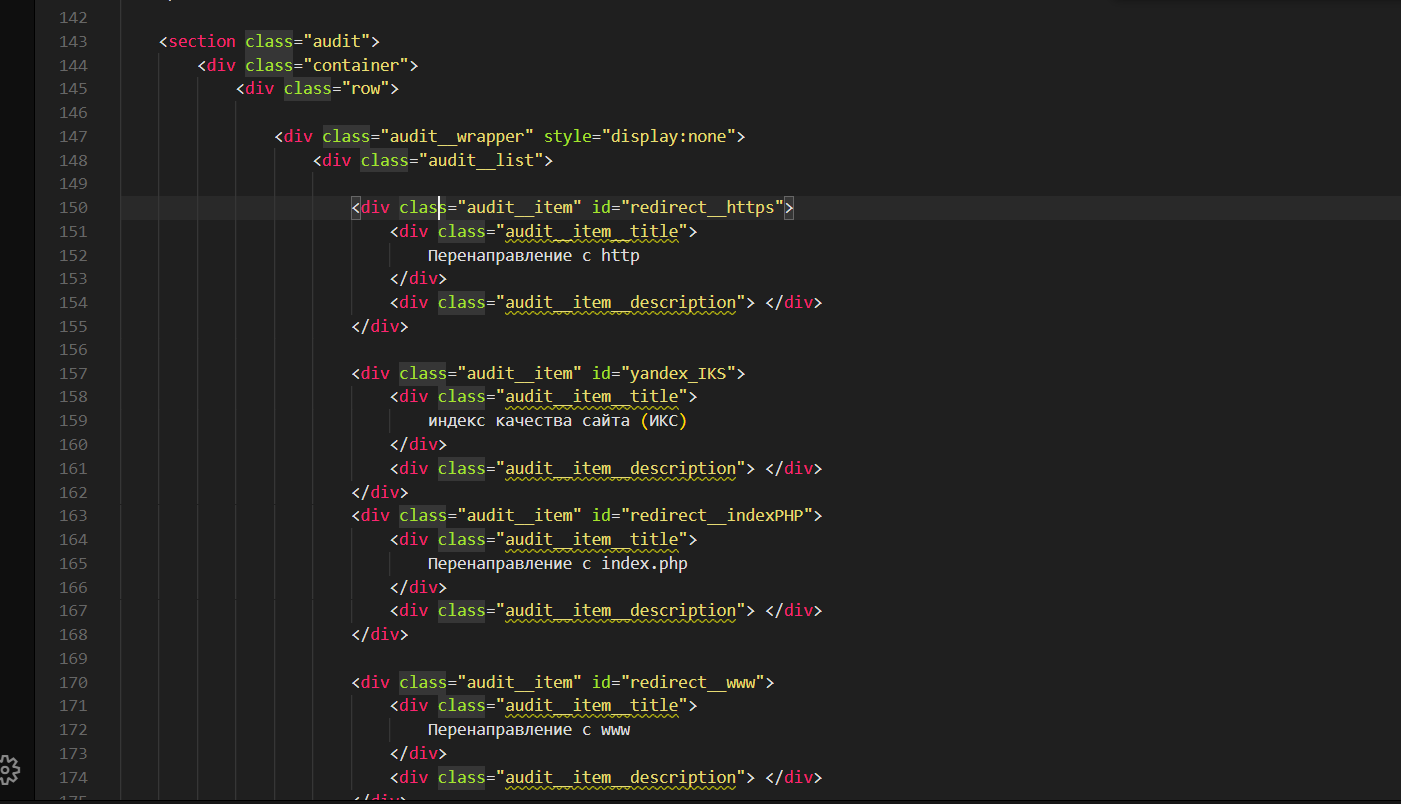




1. Далее идет форма с выбором ограничения числа найденных ссылок на главной страницы введенного сайта (это можно если что легко убрать )



1. Далее идут блоки, в которых важны названия ID – они формируют запросы к компонента, которые называются также, как и ID\_name. Все эти блоки нужны для проверок, которые начинают работать после получения ответа от файла /im-pulse33.ru/public\_html/phpQuery\_\_crawling/get\_\_json.php . Когда ответ придет, то начинается куча запросов на урл: /phpQuery\_\_crawling/components\_for\_front**/ID\_NAME – ID\_name это ID блока.** Каждый блок что-то выводит или получается какие-то данные. Везде данные сырые если что, результаты на фронте могут быть ошибочными. Самое главное формирование общего **массива audit\_array,** c которым можно будет далее работать.



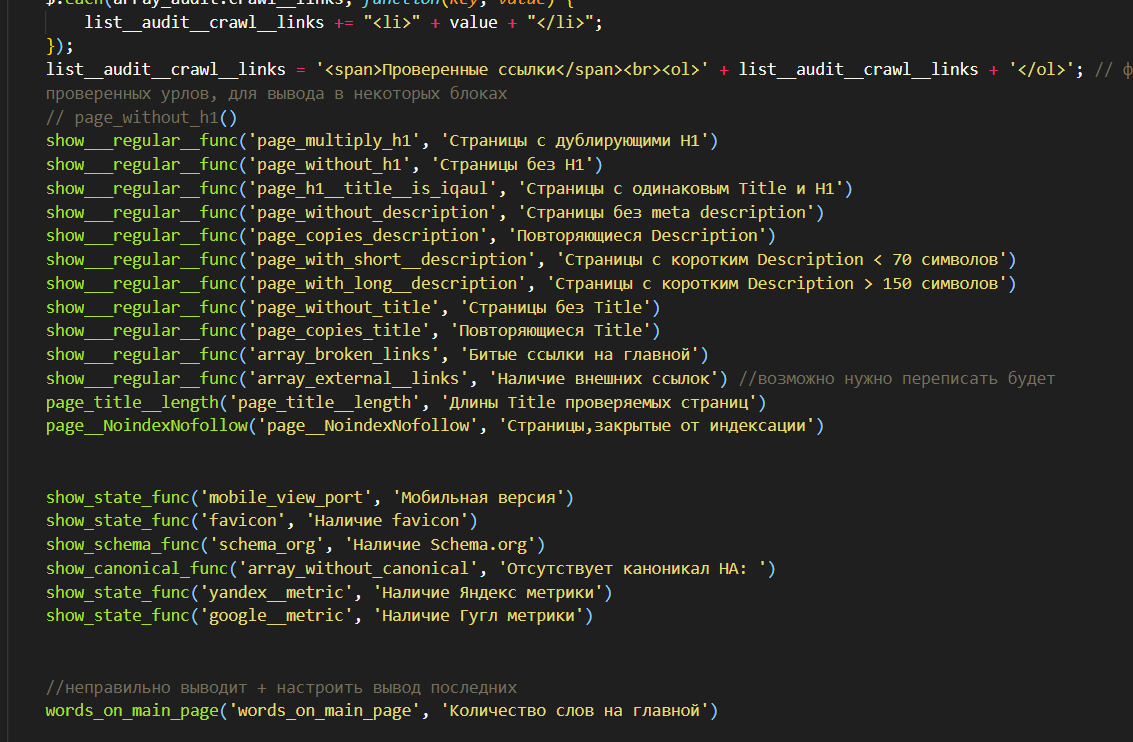
1. Далее идут различные JS функции, которые регулируют отправки.



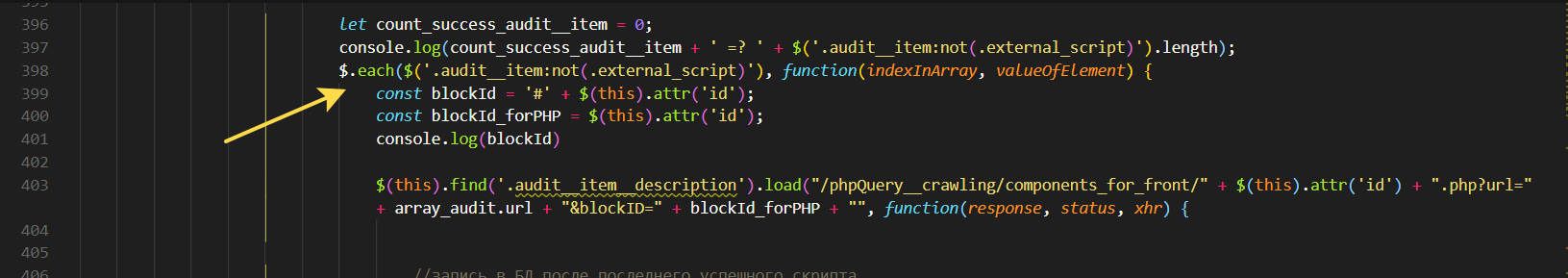
Перед отправкой формы идет проверка на url. Формируется строка вида : [*~~http://~~*domain.RU\com](http://domain.RU\com)

1. Идет запрос к файлу /im-pulse33.ru/public\_html/phpQuery\_\_crawling/**get\_\_json.php**, если нажали кнопку отправить
2. После успешной отправки получаем данные из этого файла

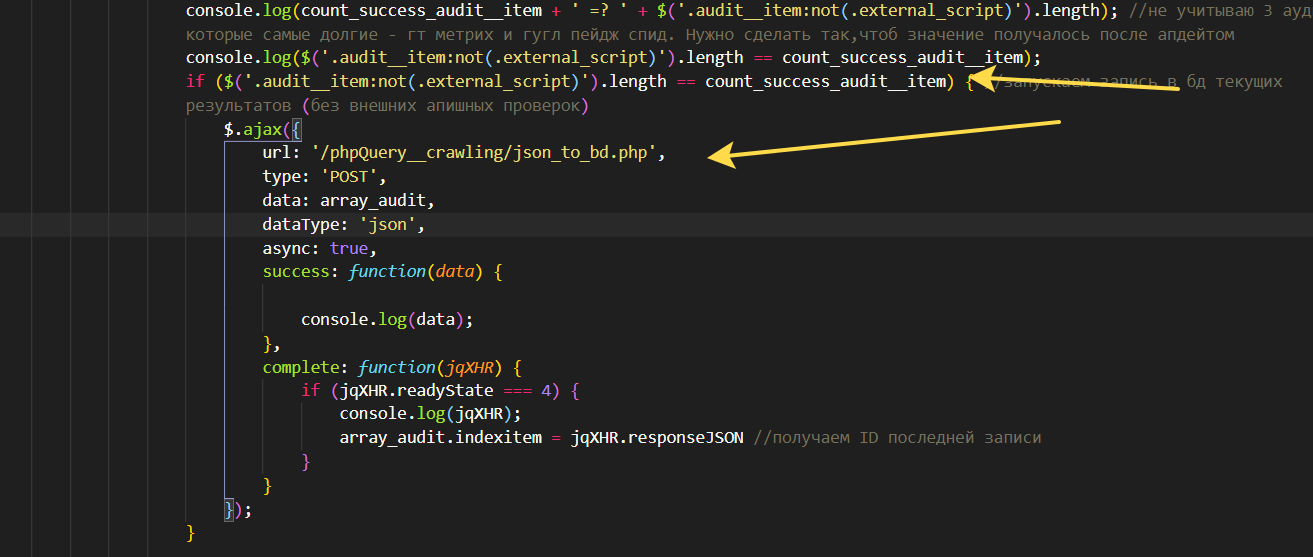
Далее идут всякие функции (они все описаны в самом низу индексного файла), которые как-то парсят полученные данные. **РЕЗУЛЬТАТ ПАРСА МОЖЕТ БЫТЬ НЕ ИСТИННЫМ**. Это надо учитывать, так как правок было миллион, **главное результрующий массив audit\_array**



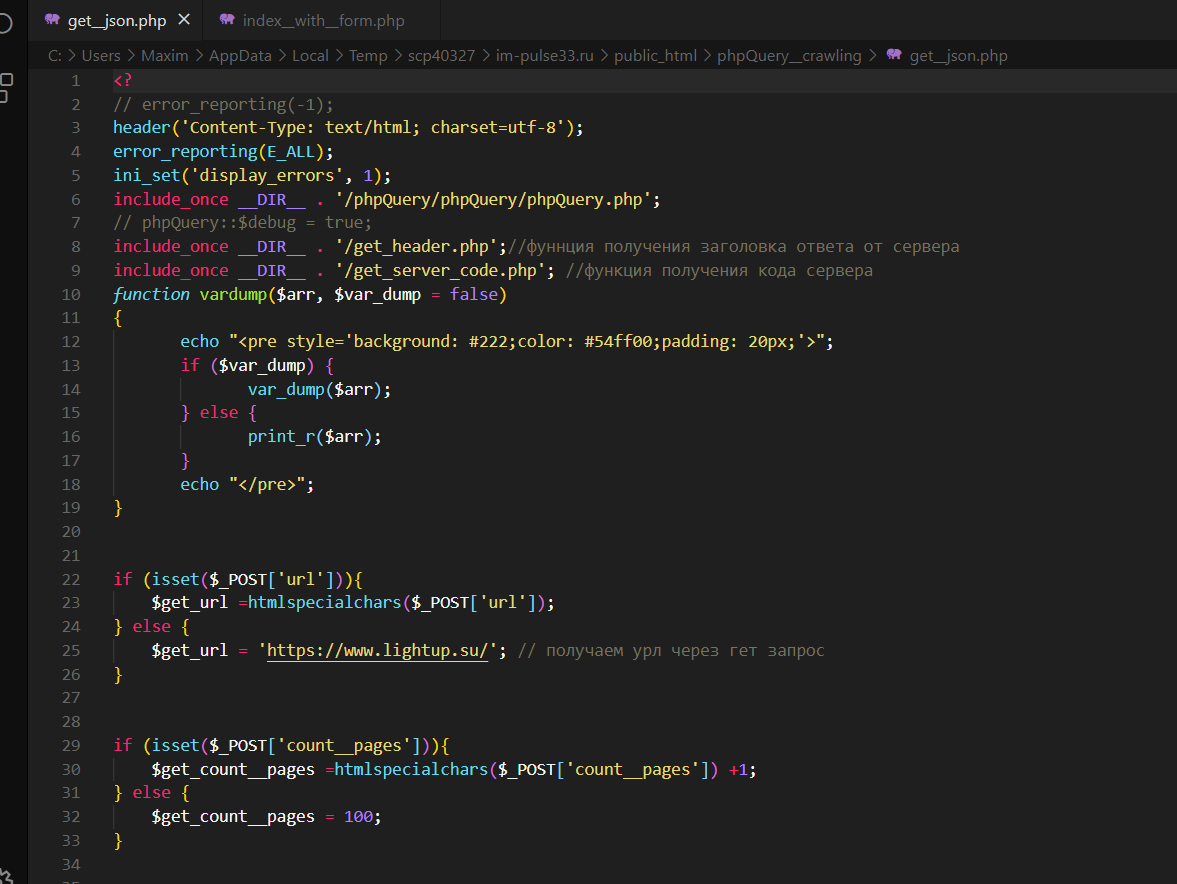
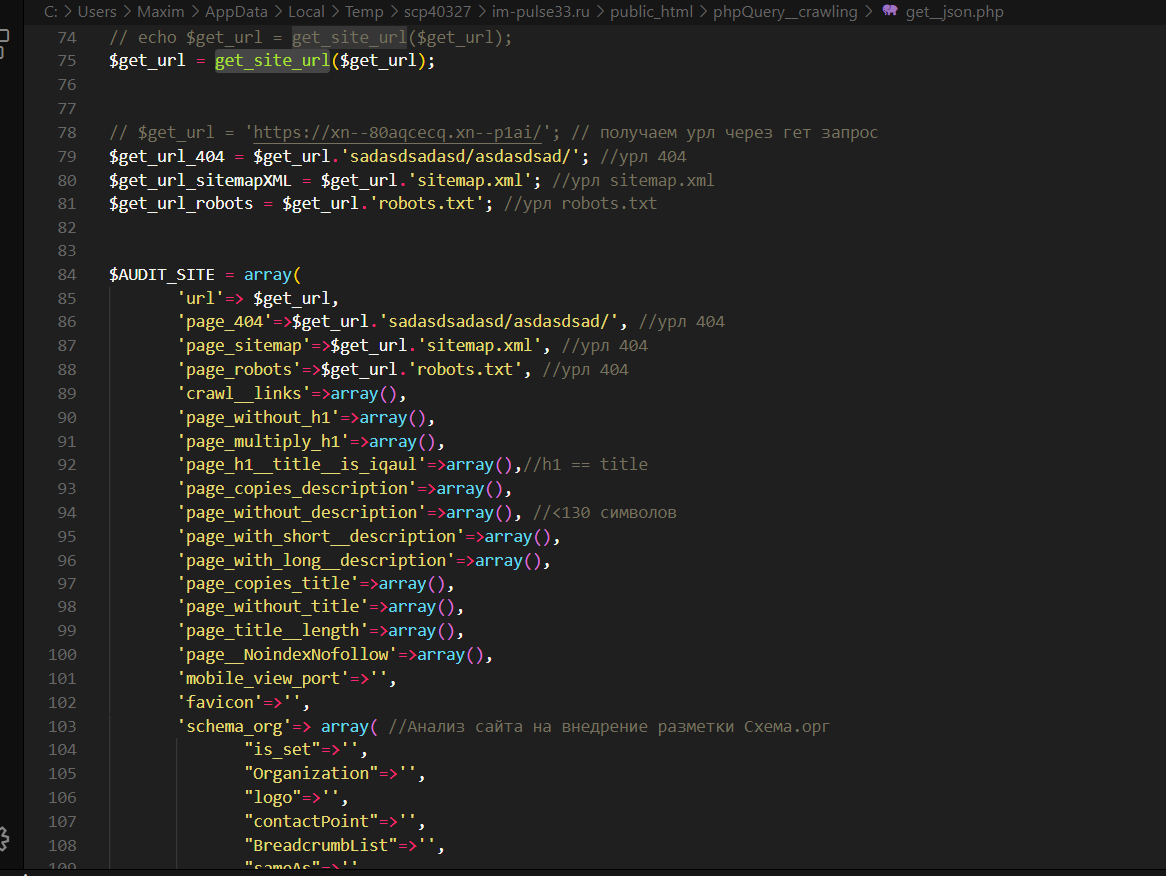
1. Далее сразу же после успешного ответа от файла get\_json создаются запросы к отдельным компонентам, которые что-то выводят или добавляют в основной массив

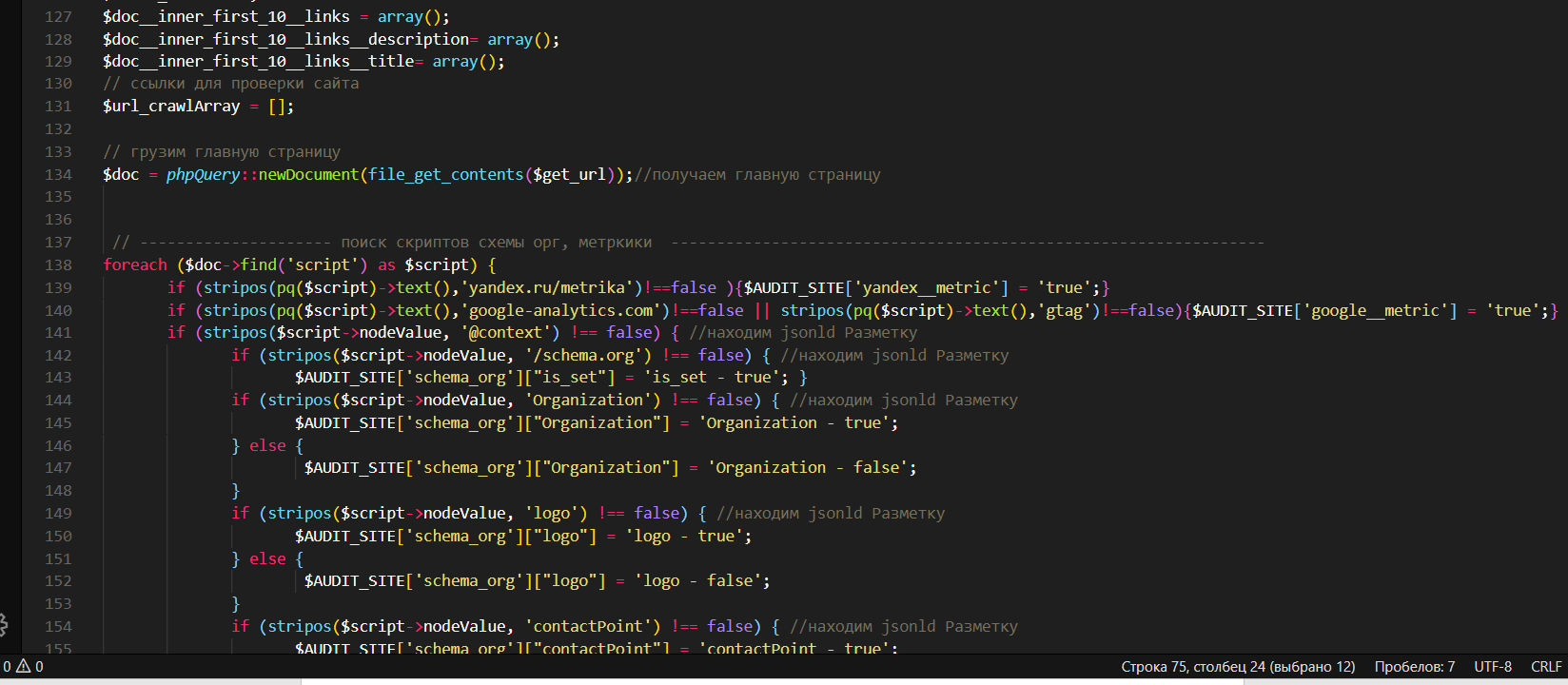


1. Далее идет запись в бд, когда все основные проверки сделаются. Записываются все поля array\_audit



# Описание файла /phpQuery\_\_crawling/**get\_\_json.php**

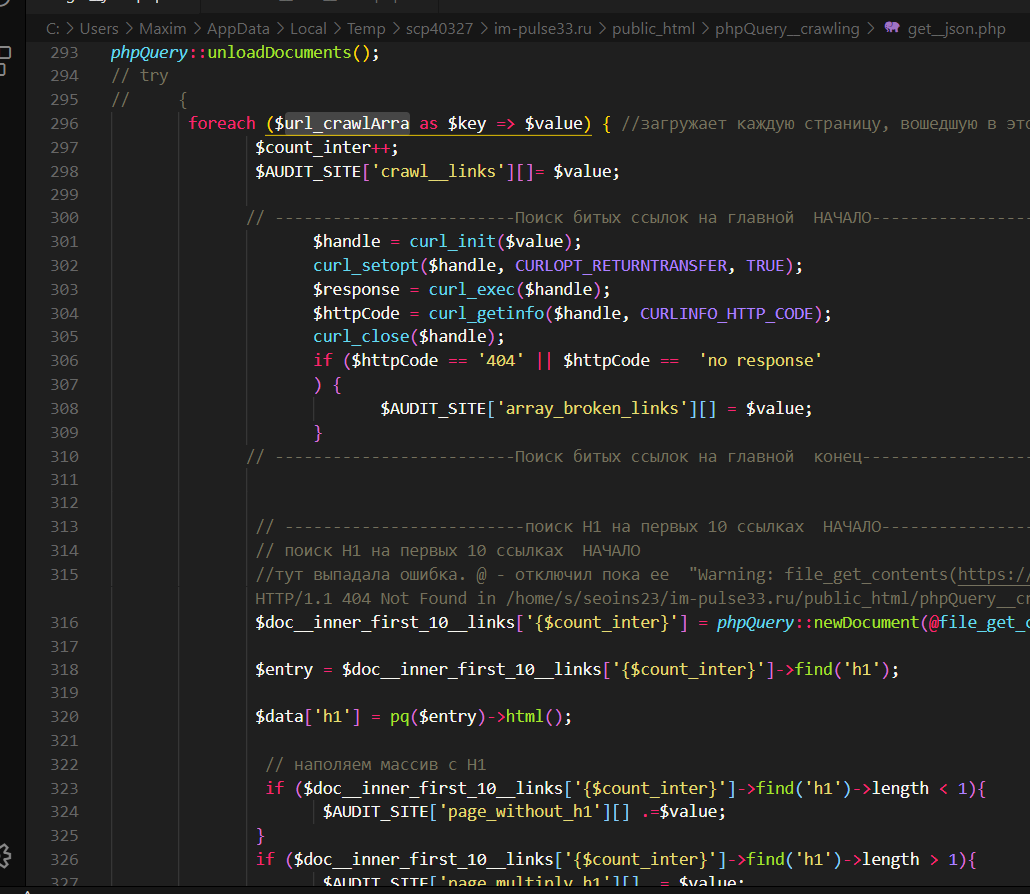
1. В самом верху файла идет подключение библиотеки phpQuery (это некое подобие jquery, с помощью которого парсятся страницы). Далее идет проверка что пришло из $\_POST : ограничения количества проверяемых страниц и сам непосредственно УРЛ сайта
2. Далее идет объявления основного массива и получение проверяемого урла через доп функцию, которая смотрит какой урл пришел : http или https
3. Далее идет запрос к самой первой страницы, что ввели в поиске. Собирается всякая инфа на первой странице (ищется схема орг , количество слов и прочая мелочь)



1. После идет поиск всех ссылок в HEADER и BODY. Группировка их, убирание дублей. Так мы формируем массив ссылок сайта, по которым в дальнейшем будем бегать.



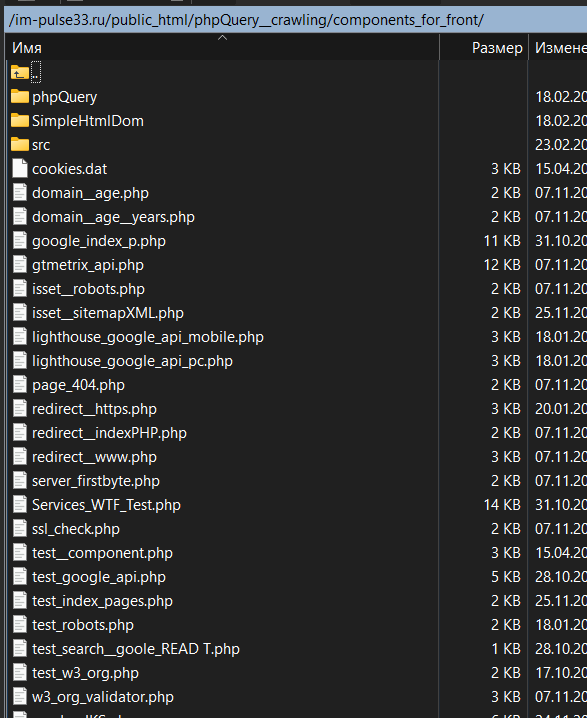
1. После собранных ссылок – идет прогон\переобход их. Собирается какая-то информация и заполняются данные в результирующий массив



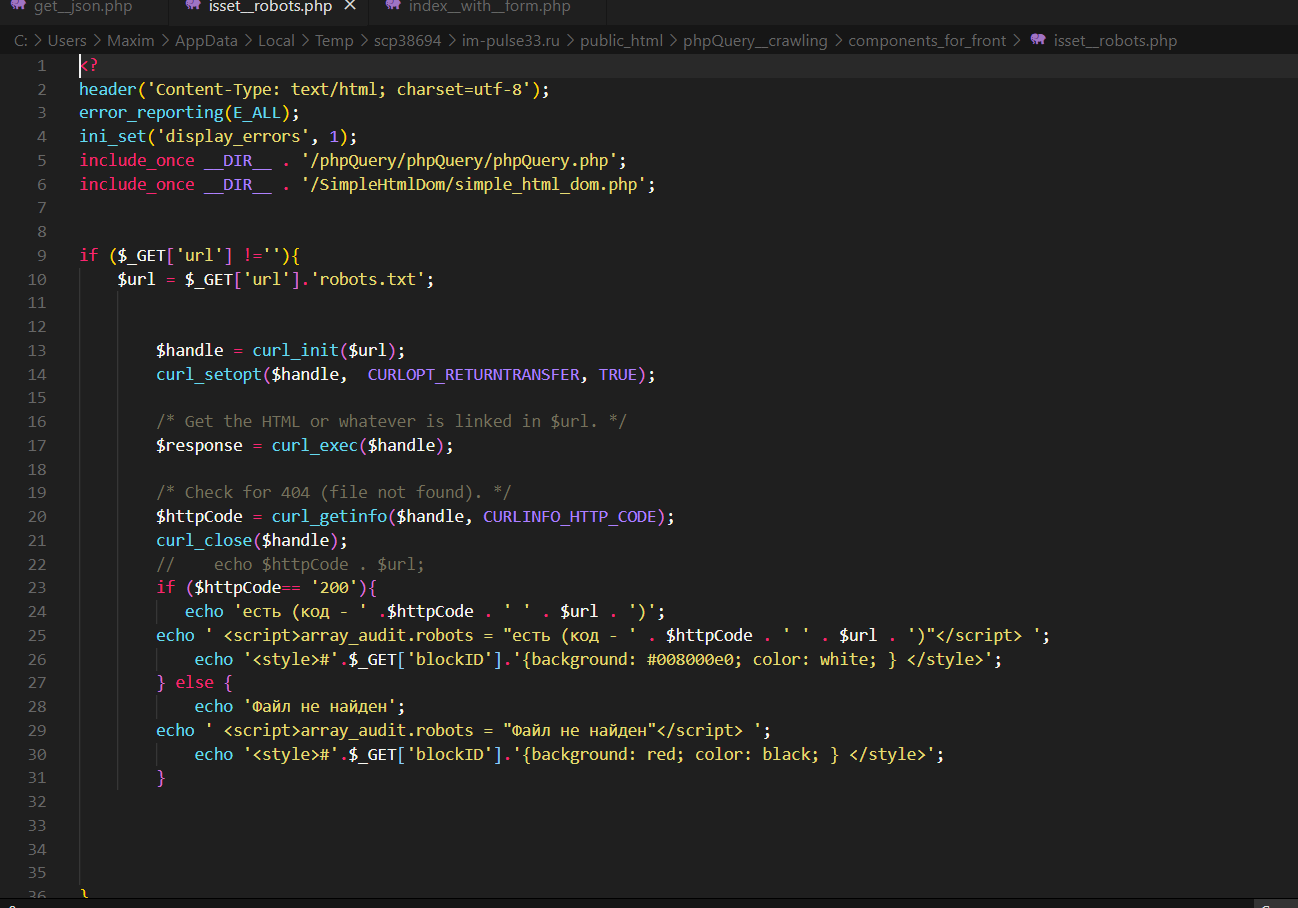
1. Все или многое закоменчено. Можно проследить, посмотреть логику. Все называется в соответствии с собираемыми параметрами. После всех проверок результаты падают в массив **audit\_array**, который отправляется на фронт (index\_\_with\_\_form.php).
2. После этого сразу же начнут работать скрипты по сбору данных из компонентов через **ID\_name** блоков

# Описание раздела components\_for\_front

Тут все достаточно просто. Каждый файл выполняет какую-то функцию. К файлу идет запрос в соответствии с названием ID\_name



Пример рядового файла:



1. В нем идет получение урла с индексного файла
2. Далее идет какая-то проверка
3. Идет какая-то выдача результата + **ВАЖНО**: запись результата в основной массив **array\_audit**

Основные файлы, требующие внимания (в них есть ключ апи, какой-то период или ограниченное число запросов за какое-то время):

gtmetrix\_api.php

lighthouse\_google\_api

w3\_org\_validator

# Файл /phpQuery\_\_crawling/json\_to\_bd.php

Через него идет запись в БД всех полей. Поля соответствуют названию переменных в результирующем массиве.