



Как я делал инструменты разработки WordPress

Владимир Петрозаводский

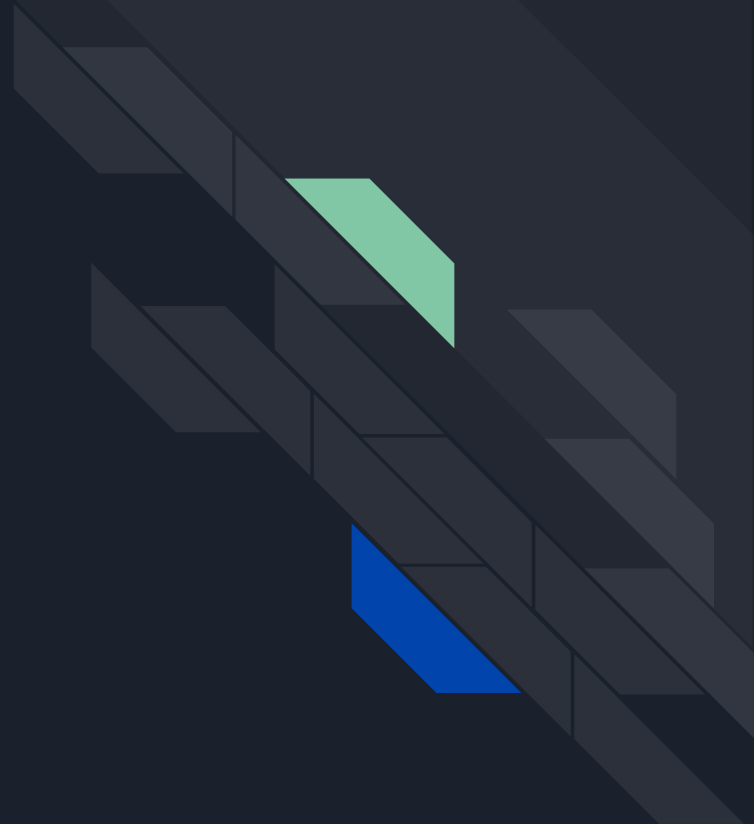
Вводные

Десятки тысяч материалов;

Миллионы уникальных посетителей;

Частные релизы фитч;

Сложная верстка.





Проблемы на бекенде

- 1 . Приходится писать много `include`.
2. Постоянное добавление префиксов функций приводит к унынию.
3. Хранить состояние придется в `$GLOBALS` потому что больше нигде.

IDE не в силах нам помочь

И к тому же импорты долго писать...

```
/**
 * Single answer output
 */
add_action('single-answer', function($data) {
    $data = array_merge([
        'attr' => [
            'class' => 'comment group',
        ],
        'questionId' => null,
        'questionStatus' => -1,
        'answer' => null,
        'canMarkAsBest' => false,
        'isAuthor' => false,
    ], $data);
    include MY_TEMPLATE_PARTS . '/questions/single-answer.php';
});

/**
 * Answer discussion output
 */
add_action('qa:answer-discussion', function($answer) {
    include MY_TEMPLATE_PARTS . '/questions/discussion.php';
});
```

Path 'MY_TEMPLATE_PARTS/questions/single-answer.php' not found more... (Ctrl+F1)

Долго писать

Как назвать `initial_function` в следующих плагинах ?

```
bootstrap.php x
1  <?php
2  /** @package Auth ... */
18
19  function initial_function(){
20      include_once ("Core.php");
21      $core = Core();
22      $core::init();
23  }
24
25  add_action('plugins_loaded', 'initial_function');
26
```

Название плагина

awesome-updater

может ПРЕФИКС ДОБАВИТЬ

awesome_updater_initial_function...



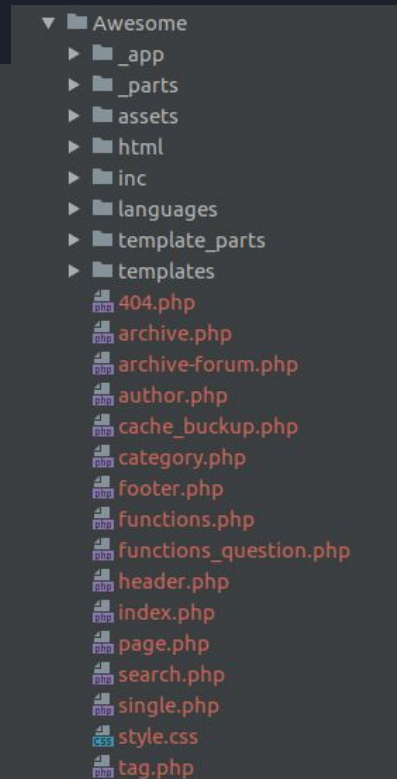
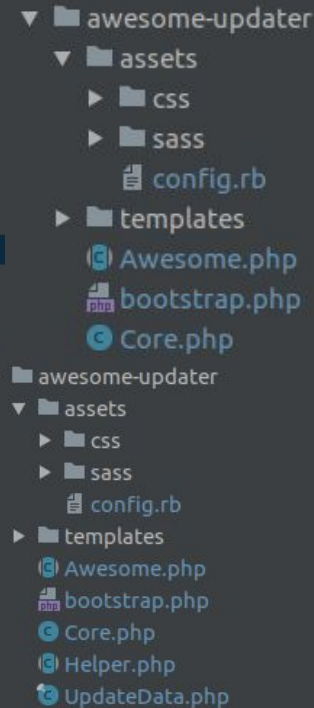
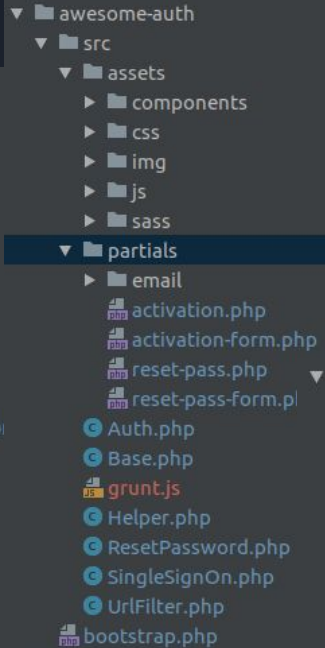
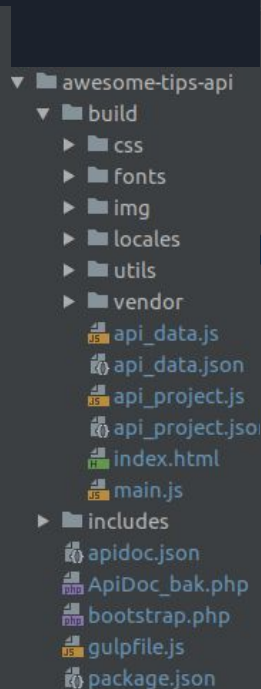
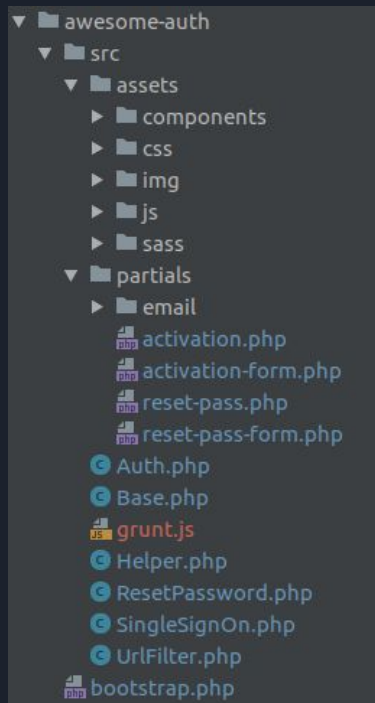
Так плохо

```
3 $awesome_updater_array = [];  
4  
5 function awesome_updater_add($elem)  
6 {  
7     global $awesome_updater_array;  
8     $awesome_updater_array = $elem;  
9 }  
10  
11 function awesome_updater_remove($key)  
12 {  
13     global $awesome_updater_array;  
14     unset($awesome_updater_array[$key]);  
15 }
```

Так лучше

```
3 namespace Awesome;  
4  
5 class Core  
6 {  
7     public $array = [];  
8  
9     public function add($elem)  
10    {  
11        $this->array[]=$elem;  
12    }  
13  
14     public function remove($key){  
15         unset($this->array[$key]);  
16     }  
17 }
```

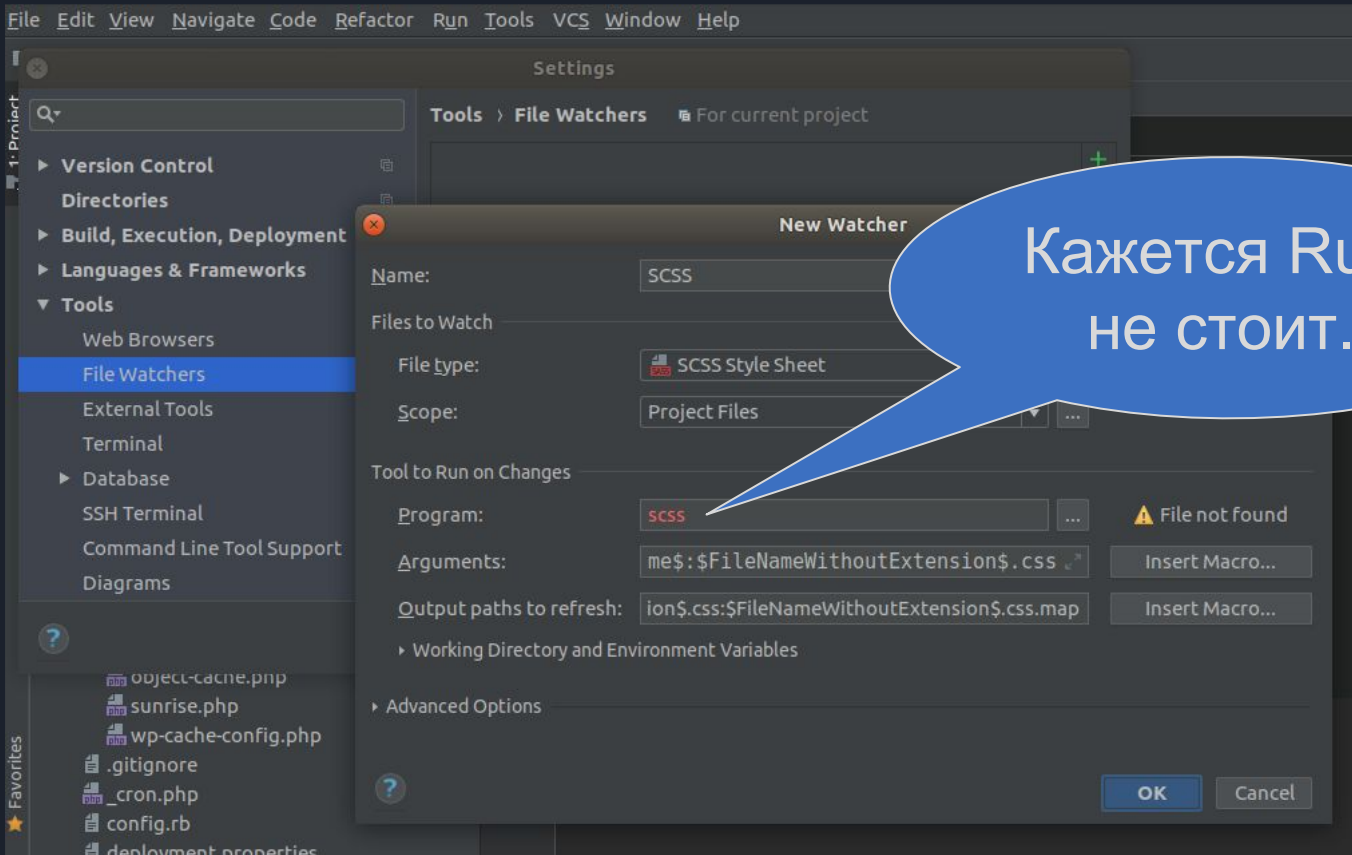
Проблемы на фронтенде





Анализ проблем

- 1 В каждом плагине разная структура каталогов
- 2 Используется множество разных сборщиков, нет общего конфига и плана действий
- 3 Много css и js файлов работают не оптимально при подключении на страницу
- 4 Скрипты имеют неразрешенные зависимости





3 Фатальных недостатка

- 1 . Нет стандартов использования PSR, использования composer пакетов, рекомендаций по применению IDE, непонятно нужно ли ООП (нужно)
2. Нет стандарта написания фронтенда и сборки, оптимизировать и и управлять файлами скриптов и стилей придется вручную
3. WordPress писали не мы !

Пришло завладеть инициативой

Нужен сценарий для быстрого создания плагинов;

Сборщик и препроцессор;

Автозагрузка классов;

Готовность к установке composer пакетов;

Подключение скриптов только на нужных страницах



Yeoman/YeoPress



GenerateWP

GenerateWP

Delightful tools for WordPress Developers

The easiest and fastest way to create custom and high quality code for your WordPress project using the latest WordPress coding standards and API's.

Join Now

Login

How it Works

1 Generator

Post Type Generator
Taxonomy Generator
Shortcodes Generator
Meta Box Generator

2 New

New Product
New Item
New Item
Edit Item
Product

3

```
'Products', 'Product', 'Product', 'Product', 'text', 'Post Type', 'Item Archives', 'Parent Product: All Products', 'Add New Product', 'New Product', 'New Item'
```

GenerateWP

The easiest and the fastest way to create custom and high quality code for your WordPress project using the latest WordPress coding standards and API's.

How it works

1 Content

Taxonomy
Post Type
Post Status
Shortcode

Select a tool

Select from dozens of generators to enhance your workflow.

2 Add New Item

Add New Product
Add New
New Product

Fill in the form

GenerateWP uses user-friendly forms to output valid code.

3

```
'_x', 'Product', 'Cake', 'Product', 'Parent Product', 'All Product', 'View Product', 'Add New Product', 'New Product', 'Edit Product', 'Item'
```

Generate the code

Copy the ready-to-use code directly to your project.

Signin



После обсуждения с командой

Хорошо бы ещё и графику в процессе оптимизировать.

И SVG тоже оптимизировать.

При верстке хорошо бы source map создавать.

Нужен инструмент для создания pot файла, руками долго.

Пусть виджеты сами регистрируются классы же умеют это.

Пусть css и js файлы виджетов и шорткодов

только на нужных страницах будут.

И сахар для подключения стилей, стандартный неудобен.

И это не все, потому что не хватит никаких слайдов.



Подключение стилей

```
// Было  
wp_enqueue_script('newscript', plugin_dir_url(__FILE__) . '/js/custom_script.js');  
// Стало  
$this->addJs("newscript");
```



Избавляемся от require

```
// Было  
include_once ( "src/WordPressMagicBoilerplate/Classes/MyClass.php" );  
new MyClass( $my_var );  
// Стало  
new MyClass( $my_var );
```


Теперь у нас есть namespase и это первый шаг к настоящему ООП



Но это не точно



Спасибо за внимание!

Ставьте звезды в
гитхабе.





WordPress Plugin Boilerplate

<https://github.com/DevinVinson/WordPress-Plugin-Boilerplate>





WordPress Plugin Boilerplate

<https://github.com/petrozavodsky/wordpress-magic-boilerplate>

