

2. Hooks

1. Primeiros passos

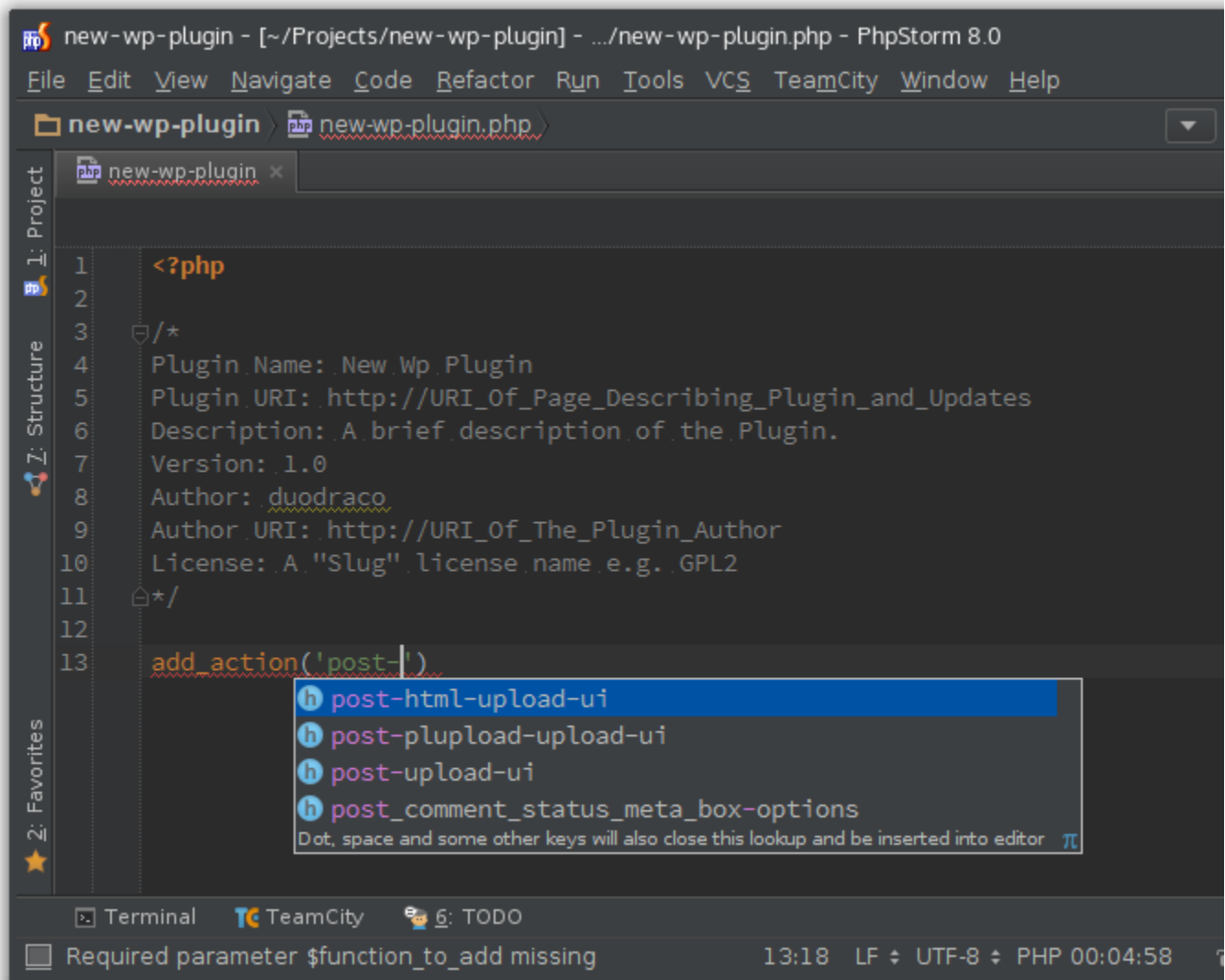
Desenvolvendo com o WordPress no PhpStorm

3. Padrão de código do WordPress


- Acabamento de código e filtro de parâmetros
- Navegação de uma Action ou Filter para a invocação do hook
- Callbacks do registro de Hooks
- Navegar para o símbolo ... para Hooks
- Encontrar utilizações para funções de registro de Hook.

Acabamento de código e filtro de parâmetros

Qualquer hook declarado em plugins inclusos ou no core da instalação WordPress definida na configuração do projeto é indexado no IDE. Os nomes dos hooks ficam disponíveis no acabamento de código (Ctrl + espaço) para parâmetros de filters e actions (add_action e add_filter).



Navegação de uma Action ou Filter para a invocação do hook

A partir do registro de um hook (via `add_action` e `add_filter`) você pode navegar para a invocação do hook com o ícone `h` na margem do editor .

```
new-wp-plugin x
1 <?php
2 /*...*/
11
12 do_action( 'admin_menu', 'example_posts_menu' );
```

Is invoked in 'menu.php'

Isso te levará para a linha onde o hook é invocado, neste caso com o `do_action`.

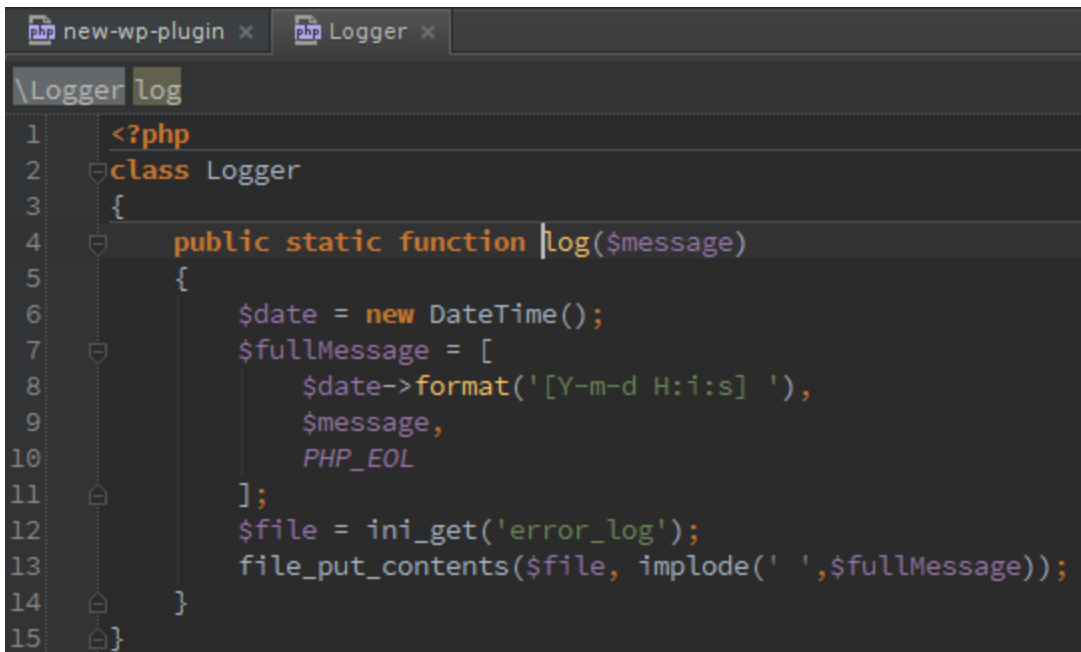
```
new-wp-plugin x menu x
120 if ( is_network_admin() ) {
121
122 /* Fires before the administration menu loads in the Network Admin. ... */
129 do_action( 'network_admin_menu', '' );
130 } elseif ( is_user_admin() ) {
131
132 /* Fires before the administration menu loads in the User Admin. ... */
139 do_action( 'user_admin_menu', '' );
140 } else {
141
142 /* Fires before the administration menu loads in the admin. ... */
149 do_action( 'admin_menu', '' );
150 }
151
```

Callbacks do registro de Hooks

Você pode ir para a declaração da função ou método especificada pelo segundo parâmetro das funções de registro de hook (`do_action` e `add_filter`) com `Ctrl + Click` ou `Ctrl + B` (`Cmd + Click` ou `Cmd + B` no Mac).

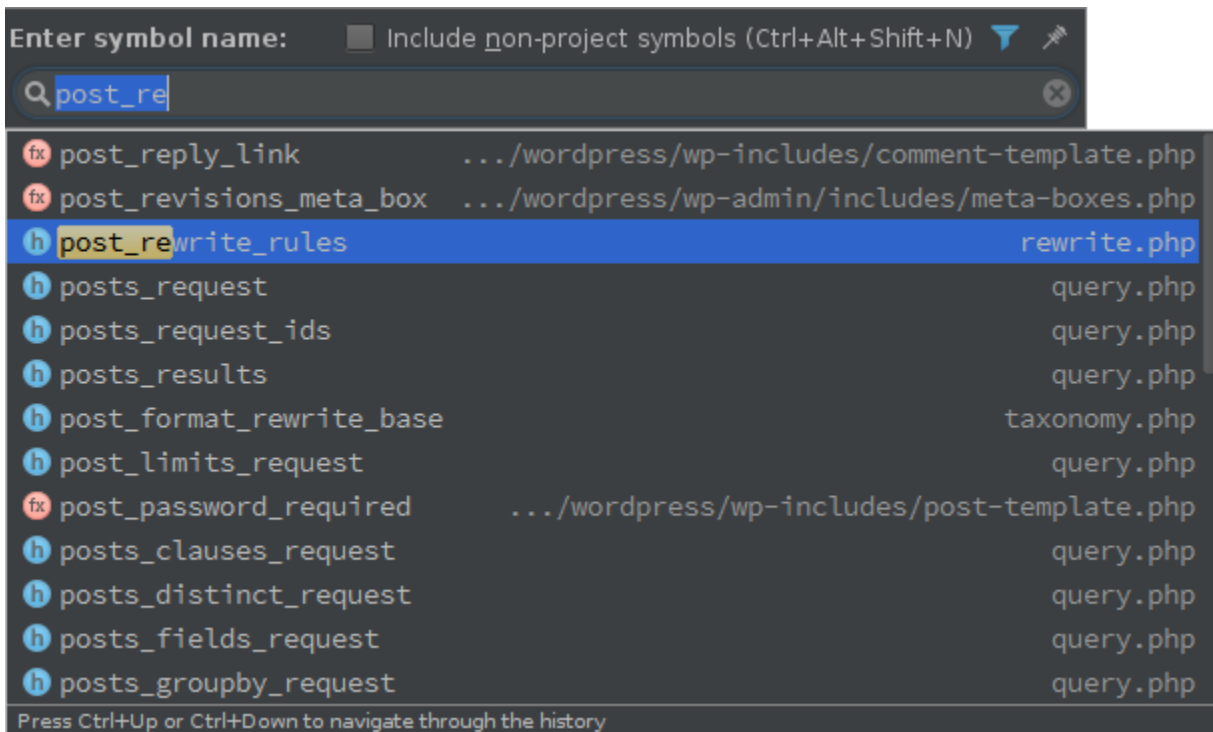
```
new-wp-plugin x
1 <?php
2 /*...*/
11
12 if ( ! function_exists( 'add_action' ) ) {
13     echo "Sou um plugin e devo ser usado como tal";
14     exit;
15 }
16 require __DIR__ . '/Logger.php';
17 add_action( 'init', [ 'Logger', 'log' ] );
```

Você será direcionado para a declaração da função ou método configurado no hook.



Navegar para o símbolo ... para Hooks

Você pode acessar qualquer símbolo no projeto pelo menu `Navigate Symbol...` ou pelo atalho `Ctrl + Alt + Shift + N` (Cmd + Alt + o no Mac), inclusive para Hooks.



Resultados similares podem ser obtidos com o `Buscar em Tudo` (Shift duas vezes). Certifique-se de habilitar o checkbox `Incluir de non-project items` (incluir itens fora do projeto) se necessário.

Encontrar utilizações para funções de registro de Hook.

Invocar o `Encontre Utilização` (Find Usages, Alt + F7) para funções de registro de hook irá fornecer uma lista de possíveis opções para que você selecione o que está procurando em termos de nome de função ou hook.

```
17 add_action('_admin_menu', ['Logger', 'log']);
```

Find Usages Of add_action

- fx add_action .../wordpress/wp-admin/load-styles.php
- fx add_action .../wordpress/wp-admin/load-scripts.php
- fx add_action .../wordpress/wp-includes/plugin.php


Find: Usages of add_action in Project Files Usages of add_action in Project Files

- Function
 - fx add_action
- Dynamic usages (1 usage)
 - Function call (1 usage)
 - simple-logger-plugin (1 usage)
 - (1 usage)
 - simple-logger-plugin.php (1 usage)

(17: 1) add_action('_admin_menu', ['Logger', 'log']);

 Saiba mais sobre Hooks na documentação oficial do Wordpress e no diretório de Hooks WordPress do Adam Brown.

 1. Primeiros passos

 Desenvolvendo com o WordPress no PhpStorm

3. Padrão de código do WordPress 