[](https://styde.net/php-companion-para-sublime-text-3)

Cuando trabajamos con PHP y utilizamos programación orientada a objetos, como en el caso de Laravel, siempre es conveniente usar atajos que nos permitan por ejemplo ir al archivo donde está definida una función especifica, importar un *namespace*, agregar automáticamente una definición a un constructor, etc. Por defecto Sublime Text no nos permite realizar ninguna de estas tareas, pero gracias al plugin PHP Companion (disponible únicamente para Sublime Text 3) podemos obtener todos estos atajos.

**Instalación de PHP Compaion**

Lo primero que debemos hacer es ingresar a la ventana del administrador de paquetes de Sublime Text desde ***menu > preferences> package control*** y posteriormente buscamos la opción ***Install Package,***una vez se abra el dialogo buscamos **PHP Companion**y esperamos a que finalice la instalación.

Si no tienes activo el administrador de paquetes puedes hacerlo siguiendo la primera parte del tutorial sobre [GitGutter para Sublime Text](https://styde.net/gitgutter-para-sublime-text/)

Así de fácil es! ya tienes instalado el plugin ahora vamos a continuar con la configuración.

**Configuración de accesos rápidos**

Por defecto el plugin no tiene configurada ningún acceso rápido que nos permita tomar ventaja de sus funciones, pero esto podemos hacerlo fácilmente desde otro de los archivos de configuración de Sublime en **menu > preferences > key binding > user**, allí podrás ver un array donde puedes crear accesos directos a diversas funciones de tu editor de texto, a continuación un ejemplo de los accesos más utlies que nos da sublime, toma en cuenta que puedes editar estos datos para usar la combinación de teclas de tu preferencia.



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | [    { "keys": ["f6"], "command": "expand\_fqcn" },    { "keys": ["f5"], "command": "find\_use" },    { "keys": ["f4"], "command": "import\_namespace" },    { "keys": ["shift+f12"], "command": "goto\_definition\_scope" },    { "keys": ["f7"], "command": "insert\_php\_constructor\_property" }  ] |

* **find\_use:** permite importar la sentencia ***use***en la cabecera del archivo con el *namespace*correcto.
* **expand\_fqcn:** permite obtener el nombre completo de la clase seleccionada.
* **goto\_definition\_scope:** permite buscar el archivo donde esta definida la clase seleccionada.
* **import\_namespace**: permite agregar la definición del *namespace* basado en el nombre del archivo donde se encuentre el cursor.
* **insert\_php\_constructor\_property:**permite agregar una propiedad al constructor de la clase donde se encuentre el cursor, si  el constructor no existe, se crea automáticamente y en caso de que exista, simplemente actualiza el código agregando la nueva propiedad.

**Configuración adicional**

Algunas veces el plugin falla o muestra un mensaje molesto en la consola como este



|  |  |
| --- | --- |
| 1 | FileNotFoundError: [Errno 2] No such file or directory: 'C:\\Users\\USERNAME\\AppData\\Roaming\\Sublime Text 3\\Packages\\PHP Companion\\PHP Companion.sublime-settings' |

Para ver la consola de sublime ingresa a ***menu > view > Show Console.***

Una solución sencilla que no afecta al funcionamiento del plugin o de tu editor en general es simplemente crear el archivo faltante en la ruta indicada.

Debes crear el archivo tal cual como se muestra en la consola incluyendo espacios en blanco.

Perfecto, para probarlo simplemente ingresa a cualquier controlador o clase de tu proyecto y escribe por ejemplo:



|  |  |
| --- | --- |
| 1 | $var = new Request |

Luego pon el cursor sobre el nombre de la clase (Request) y prueba cada una de las opciones que configuraste anteriormente (Las teclas F4, F5, F6…).

Espero que te haya gustado esta lección, si utilizas algún otro pluguin que consideras de utilidad y quieres compartirlo, puedes hacerlo en la sección de comentarios.