

Exercice : Logger

Introduction

But

Le but principal de cet exercice est de revoir et approfondir les fonctions en php.

Il permettra également d'entraîner le développeur :

- à coder à l'aide du debugger intégré à l'IDE,
- à utiliser les outils d'aide aux commentaires intégrés à l'IDE,
- à coder en partant de diagramme de flux,
- à utiliser les régions dans PHPStorm pour aider à la lisibilité du code
- à se référer à la documentation officielle de Php pour comprendre le fonctionnement des fonctions du langage.

Pré-requis

- Env. de développement PhpStorm
 - L'interpréteur php
 - Le debugger

Consigne

Partie 1 : Comprendre et préparer ses outils de travail

Dans ce petit travail pratique il vous est demandé de compléter du code existant. Etape par étape vous aller procéder à un enrichissement fonctionnel.

L'architecture du code (sa décomposition en fonction) vous est imposée. Les signatures des fonctions ne devront pas être altérées.

Gérer les régions en php

En ajoutant la balise

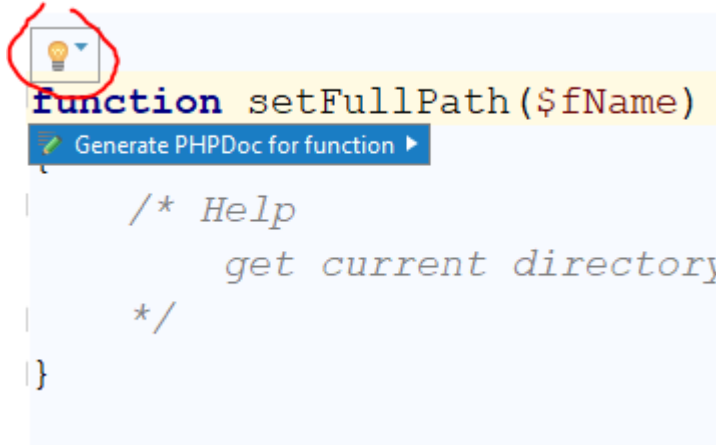
`//<editor-fold desc="[Le nom de ma région]">` pour ouvrir une région et
`//</editor-fold>` pour la fermer, vous pouvez ainsi regrouper des lignes (« collapse »).

Voilà un exemple concret :

```
//<editor-fold desc="tests automation - entry point">
//create file and set header
writeMsgInFile($fileFullPath, $logHeader, true);
//</editor-fold>
```

Commentaires avec PhpStorm

En vous plaçant dans une zone, idéalement la signature d'une fonction, PhpStorm affiche une ampoule qui vous permettra de vous aider à générer la documentation.



Voici le résultat une fois que l'assistant de commentaires a été activé :

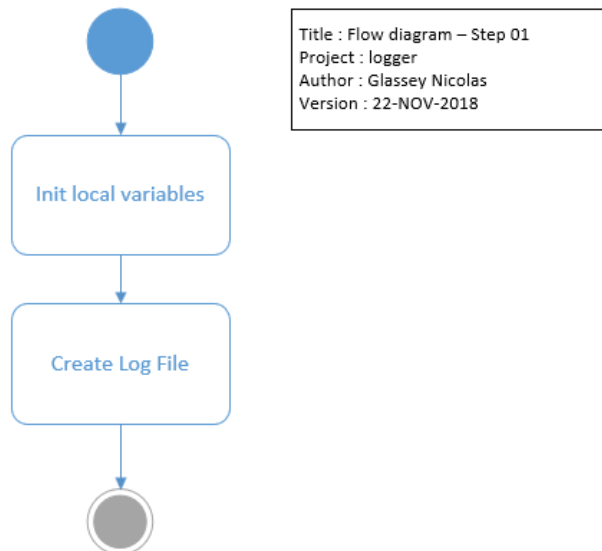
```
/**
 * This function is designed to
 * -The path will be found
 * @param $fName : The file name
 * @return [String] full path
 * @example File Name : /path/to/file
 */
```

L'attribut `@param` accompagnera chacun des paramètres présents dans la signature (`$fName` dans cet exemple). Le retour est lui mentionné `@return`.

Il est également bien vu d'ajouter une phrase explicative sur la nature de la fonction. Que doit-elle faire ?

Prise en main du code

La logique implémentée dans le code de départ est la suivante :



Aide :

- 1) Découvrez le code en partant de la région « private attributes »
- 2) Lisez ensuite la région « test automation – entry point »
- 3) Observez les 2 signatures de fonctions « setFullPath » et « writeMsgInFile »
- 4) Compléter le corps des fonctions observées en point 3

Le code est largement commenté, aidez-vous-en !

Bonne chance !