

Qu'est ce que les PSR ?

PSR est l'abréviation de PHP Standards Recommendation. Il s'agit d'un regroupement de standards et de conventions. L'objectif principal est de réunir les différentes méthodes de programmation présentes à travers les frameworks et CMS existants en PHP (Symfony, Laravel, PHP Natif,...).

Ces conventions sont gérées par un groupe nommé **PHP-Fig** : ils créent, modifient ou suppriment les standards permettant de générer une uniformité parmi les différents projets réalisés en PHP. Les PSR comportent plusieurs normes nommées de la façon suivante : le préfix PRS - suivi du numéro d'un nombre. Ainsi on a les normes suivantes : PSR-0, PSR-1...

Enfin, les normes PSR ne sont adoptées qu'après être passées dans un processus assez rigoureux. Voici les différents étapes d'une norme PSR :

1. **Pre-draft** : L'objectif de l'étape est de déterminer si une majorité de PHP FIG est intéressée à publier un PSR pour un concept proposé. Par exemple une proposition pour le nommage des variables;
2. **Draft** : l'objectif de l'étape est de discuter et de peaufiner une proposition jusqu'à ce qu'elle puisse être examinée par le Comité principal de la FIG;
3. **Review** : La phase d'examen est une occasion pour la communauté d'expérimenter une cible raisonnablement fixe afin d'évaluer l'aspect pratique d'une proposition. À ce stade, le commanditaire de la proposition est l'autorité finale quant aux changements à la spécification ainsi que toute décision de faire avancer ou reculer la proposition, cependant, l'éditeur peut opposer son veto aux modifications proposées qui, selon lui, sont préjudiciables à la conception du devis;
4. **Accepted** : Si elle est acceptée, la proposition devient officiellement une norme PSR;
5. **Deprecated** : Un PSR déprécié est un PSR qui a été approuvé, mais n'est plus considéré comme pertinent ou recommandé. Généralement, cela est dû au remplacement du PSR par une nouvelle version, mais cela n'est pas obligatoire. C'est le cas du PSR-4 qui vient remplacer le PSR-0 qui est maintenant déprécié;
6. **Abandoned** : Après qu'un PSR est marqué déprécié, elle est abandonnée après un certain temps;

La vidéo parle du PSR-1 et du PSR-4. Très légèrement du PSR-2 qui a été remplacé par le PSR-12.

Mais il en existe d'autres, on peut les regrouper en 4 grandes catégories :

- Le chargement automatique : PSR-4;
- Les interfaces : PSR-3, PSR-6, PSR-11...;
- HTTP ou la gestion des messages HTTP : PSR-7, PSR-15...;
- Et les styles de codage : PSR-1, PSR-12.

Pour conclure, PHP FIG n'a pas la prétention de se présenter comme un exemple ou des référents en matière de programmation PHP. Les PSRs sont des « recommandations » et rien ne vous oblige à les suivre. Par contre ces recommandations sont toutefois appréciées de la communauté PHP et largement reconnues.

Les Responsables de PSR ?

Les responsables de PSR sont le groupe PHP Framework Interop Group qui a son lancement en 2009 était composé de 5 personnes, puis s'est rejoint par d'autres avec le temps.

Leurs utilités :

Le PSR est une spécification PHP qui sert à la standardisation des concepts de programmation en PHP. L'objectif est de permettre l'interopérabilité des composants et de fournir une base technique commune à la mise en œuvre de concepts pour améliorer les pratiques de programmation.

Le respect des normes PSR permet par exemple de pouvoir créer des composants pour Symfony qui fonctionnent avec un autre Framework tel que Laravel .

Ce qu'elles contiennent :

Chaque PSR est associé à un numéro et représente un dépôt contenant une liste de conventions ou diverses interfaces permettant d'améliorer la coopération entre Développeurs PHP. Ces interfaces sont fonctionnelles dans leur état actuel mais restent néanmoins personnalisables selon le besoin. Il existe une liste assez conséquente de PSR dont certains encore en "draft" (en attente de validation ou de développement). Parmi ceux-ci, la partie qui nous intéresse est la liste "active". Il s'agit de la liste validée par PHP-Fig et qui doit être mise en place durant la programmation d'un projet en PHP. Cette liste est en constante évolution et sera amenée à changer avec le temps.

La PSR 1 évoque les règles de base de php.

En langage php les fichiers doivent utiliser les balises `<?php ?>` et les balises `<?= ?>`.

Il est impératif d'utiliser le codage UTF-8 sans BOM (Byte Order Mark).

Les classes et les namespace doivent se charger par autoloading : cela permet le chargement et le lien automatique des parties d'un programme à partir du stockage de masse lorsque cela est nécessaire, de sorte que le programmeur n'est pas tenu de définir ou d'inclure ces parties du programme de manière explicite.

Utiliser la convention UPPERcamelcase pour le nom des classes.

Écrire les méthodes en camel case.

La PSR-12 ou 2 dit qu'après le namespace il doit y avoir une ligne blanche, après une déclaration de classe ou méthode on va à la ligne mais pas pour un if. Pareil, il dit qu'il faut se limiter à 120 caractères par ligne.

La PSR-4 est une amélioration de la PSR-0, cette PSR évoque ce qu'on a vu précédemment l'autoloading ou l'auto-chargement. Il faut avoir dans son code un namespace principale et il est possible d'avoir plusieurs sous-namespace, il doit finir par le nom de la classe, interfaces, traits ou tout autre structures similaires.

Le premier namespace doit être le nom du fichier.

Ce PSR donne des informations pour bien nommer les fichiers afin que l'auto-chargement soit plus compréhensible. Ils déterminent la règle permettant de décider quel dossier correspond à quel namespace et inversement.

Liens intéressants :

Par exemple, les PSR php sont notées sur le site :

<https://www.php-fig.org/psr/>

<https://laravel.sillo.org/psr-7/>

<https://stackoverflow.com/psr-les-bonnes-pratiques-php/>

<https://www.adimeo.com/blog/psr-les-standards-de-programmation-en-php>

<https://www.itefficiency.com/article/qu-est-ce-que-les-psrs-et-a-quoi-servent-ils/>

<https://grafikart.fr/tutoriels/psr-standards-903>