

# Convention de codage WaveContact

---

## PHP

### Formatage des fichiers

#### Général

Si un fichier ne contient que du code PHP, il est impératif de ne pas insérer les balises de fin de script « ?> » à la fin du fichier. Ceci permet de se prémunir de l'ajout accidentel d'espaces blancs dans la sortie.

### Hierarchie des fichiers dans L'API

Tous les fichiers de l'api doivent être organiser par rôle dans des dossiers.

Par exemple : afin de récupérer tous les clients, toutes la méthode ce trouveras dans le fichier index.php comme ci-dessous :

```
Mondomaine.ch
|---- api
    |--- clients
        |--- get
            |--- Index.php
```

L'url d'accès sera donc la suivante :

```
www.mondomaine.com/api/clients/get
```

### Convention de nommage

#### Fonctions et méthodes

Les noms des fonctions et méthodes doivent être de type alphanumérique et les underscores ne sont pas permis.

La première lettre de chaque fonction/méthode doit toujours être en majuscule. Quand un nom de fonction est composé de plusieurs mots, la première lettre de chaque mot doit être en majuscule. Il s'agit de la notation camelCase (Par exemple : « GetUserList() »).

Les fonctions et méthodes doivent être nommées de façon claire et explicite. La fonction « GetUserList » a un nom explicite, contrairement à la fonction « DoSomething() ».

Pour la programmation orientée objet, les accesseurs aux objets doivent impérativement être précédés par « get » ou « set ».

Les méthodes de classes ayant une visibilité « protected » ou « private » doivent absolument être préfixée par un underscore « \_ ».

Il est recommandé de ne pas utiliser des fonctions à portée globale. Toute fonction doit être intégrée à une classe. Utiliser une classe statique si besoin est.

## Classes

Les noms des classes ne doivent contenir que des caractères alphanumériques. Les nombres sont autorisés, mais déconseillés.

Un fichier ne doit contenir qu'une et une seule classe.

### Variables

Les noms de variables ne peuvent contenir que des caractères alphanumériques. Les underscores ne sont pas permis. Les nombres sont autorisés mais déconseillés.

De la même façon que les méthodes de classe, les variables membres de classe qui sont déclarées comme « private » ou « protected », sont précédées d'un underscore.

La notation camelCase est également applicable pour les variables. Les noms de variables doivent être explicites et clairs.

Afin de clarifier au mieux l'utilisation d'une variable, le premier caractère doit donner l'information de son typage. Voici la liste des typages possibles :

- i int (entier)
- f float (nombre a virgule flottante)
- d double (float double)
- c char (caractere)
- s string (chaîne de caractères)
- b boolean (booléen)
- o object (objet)

```
$iTest = 10;  
$sTest = "test";  
$bTest = true;  
$oTest = new Test();  
$aTest = array(1, 6, 7);
```

## Constantes

Les constantes peuvent contenir des caractères alphanumériques et des underscores. Les nombres sont autorisés.

Les constantes doivent toujours être en majuscule, cependant les mots pouvant les composer doivent être séparés par des underscores ("\_"). Par exemple, « ALLINONE\_LANGUAGE ».

# C# & PHP

## Formatage des fichiers

### En-tête

Chaque fichier créer doit contenir le header suivant :

```
/*  
* PROJET : WAVECONTACT  
* Date de création : JJ.MM.YYYY  
* Description : Décrire ce que fait la classe ou le fichier en question  
* Version : 1.0  
* Commentaires : -  
* Créateur : Mettre le nom du créateur du fichier afin de pouvoir lui poser  
* des questions en cas de besoin  
*/
```