

オートローダー

composerに付属のオートロード機能を使うことで、他クラスを逐一requireする
必要がなくなります。

```
require_once 'ArticleModel.php';  
require_once 'PostModel.php';  
require_once 'UserModel.php';  
  
use App¥Models¥ArticleModel;  
use App¥Models¥PostModel;  
use App¥Models¥UserModel;  
  
class UserController {}
```



オートローダーを使うと...

```
require_once 'autoload.php';  
  
use App¥Models¥ArticleModel;  
use App¥Models¥PostModel;  
use App¥Models¥UserModel;  
  
class UserController {}
```

オートローダーの自動生成

composer.jsonに「autoload」キーと「psr-4」キーを記述し、その中に名前空間とディレクトリの対応表を書きます。

名前空間とディレクトリの対応が、PSR-4に従っている必要があります。

```
{
  "autoload": {
    "psr-4": {
      "App¥¥" : "app/",
      "App¥¥Controllers¥¥" : "app/Modules/User/Controllers"
    }
  }
}
```

PSR-4 オートローダー

- PSR-4は、オートローディングに関するコーディング規約です。
主に、名前空間と、その名前空間に対応するファイルパスの対応ルールについてまとめられています。
<https://www.php-fig.org/psr/psr-4/>
- Composerに付属のオートローダーは、PSR-4準拠です。
そのため、Composerがオートロードできるのは、PSR-4に準拠しているクラスのみです。

PSR-4のエッセンス(1)

※以後、クラス・インターフェース・トレイトなどのことをまとめて「クラス」と呼びます。

★クラスは、少なくとも1つ以上の名前空間に所属させましょう。

¥<NamespaceName>(¥<SubNamespaceNames>)*¥<ClassName>

PSR-4のエッセンス(2)

★クラス名とPHPファイル名は、一致させましょう。

例:

「SwiftMailer」というクラスの定義は、「SwiftMailer.php」というファイルに記述する。

PSR-4のエッセンス(3)

PSR-4に準拠した例を見てみましょう。
以下の表は、公式ページからの引用です。

完全修飾クラス名 (FULLY QUALIFIED CLASS NAME)	名前空間接頭辞 (NAMESPACE PREFIX)	ベースディレクトリ (BASE DIRECTORY)	結果のファイルパス (RESULTING FILE PATH)
¥Acme¥Log¥Writer¥File_Writer	Acme¥Log¥Writer	./acme-log-writer/lib/	./acme-log-writer/lib/File_Writer.php
¥Aura¥Web¥Response¥Status	Aura¥Web	/path/to/aura-web/src/	/path/to/aura-web/src/Response/Status.php
¥Symfony¥Core¥Request	Symfony¥Core	./vendor/Symfony/Core/	./vendor/Symfony/Core/Request.php
¥Zend¥Acl	Zend	/usr/includes/Zend/	/usr/includes/Zend/Acl.php

PSR-4の個別例(1)

完全修飾クラス名	¥Zend¥Acl	完全修飾クラス名とファイル名が一致している
名前空間接頭辞	Zend	名前空間接頭辞とベースディレクトリの1つが一致している
ベースディレクトリ	/usr/includes/Zend/	
結果のファイルパス	/usr/includes/Zend/Acl.php	

PSR-4の個別例(2)

完全修飾クラス名	¥Symfony¥Core¥Request
名前空間接頭辞	Symfony¥Core
ベースディレクトリ	./vendor/Symfony/Core/
結果のファイルパス	./vendor/Symfony/Core/Request.php

サブ名前空間をもっている

ベースディレクトリが相対パスになっている

PSR-4の個別例(3)

完全修飾クラス名	¥Acme¥Log¥Writer¥File_Writer	アンダースコアが入っているが、特別な意味を持つ記号ではない
名前空間接頭辞	Acme¥Log¥Writer	名前空間接頭辞 Acme¥Log¥Writerと一致している
ベースディレクトリ	./acme-log-writer/lib/	ベースディレクトリの下に 名前空間接頭辞と一致しないサブディレクトリがある
結果のファイルパス	./acme-log-writer/lib/File_Writer.php	

PSR-4の個別例(4)

完全修飾クラス名	¥Aura¥Web¥Response¥Status	個別例(3)と大きく違うのはここ。Aura¥Webまでを名前空間接頭辞とみなしている
名前空間接頭辞	Aura¥Web	名前空間接頭辞と一致
ベースディレクトリ	/path/to/aura-web/src/	名前空間接頭辞に出てこなかったResponseがここに出ている。このディレクトリ名「Response」は、大文字・小文字も含めて一致させる。
結果のファイルパス	/path/to/aura-web/src/Response/Status.php	

PSR-4の本アプリケーションでの例(1)

完全修飾クラス名	¥App¥Libs¥Mailer¥SwiftMailSender
名前空間接頭辞	App
ベースディレクトリ	./app
結果のファイルパス	./app/Libs/Mailer/SwiftMailSender.php

PSR-4の本アプリケーションでの例(2)

完全修飾クラス名	¥App¥Controllers¥IndexController
名前空間接頭辞	App¥Controllers
ベースディレクトリ	./app/Modules/User/Controllers
結果のファイルパス	./app/Modules/User/Controllers/IndexController.php

composer.jsonのオートロード定義を変更したら

composer.jsonの「autoload」定義を変更したときは、オートローダーを作り直すよう、composerに命令する必要があります。
これを行うのが、composer dump-autoloadコマンドです。

★例: step1¥composer.jsonを変更したとき

```
> cd C:¥xampp7¥htdocs¥enjoy-eats¥step1  
> composer dump-autoload
```

参考情報

- PHP本格入門(上)
「3-6 名前空間を使って、関連するクラス群をカテゴリ分けする」
- PHP本格入門(下)
「10-2-8 自作のクラスをオートロードする」