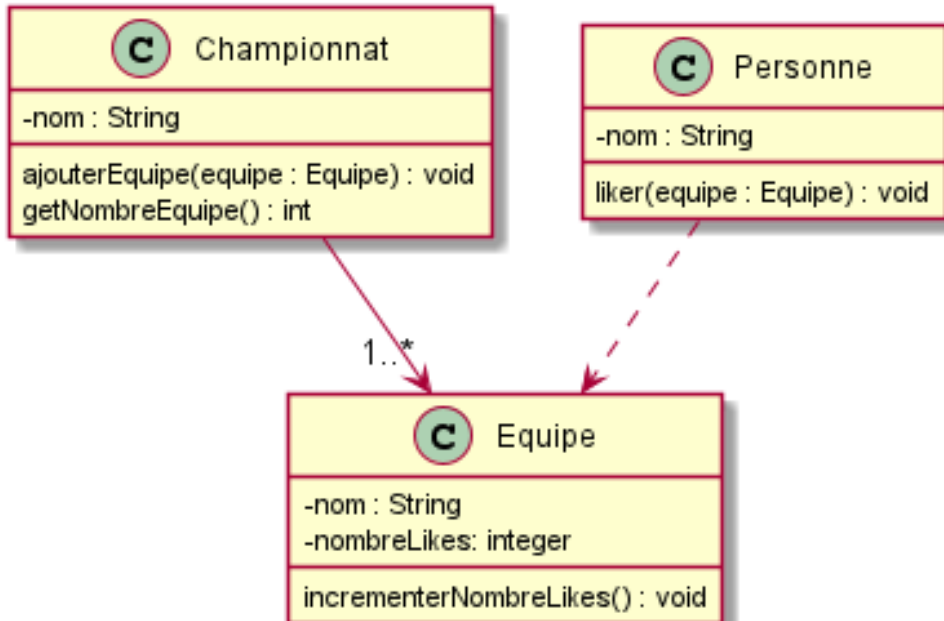




# PHP - Namespaces

Composer - Autoload



```

@startuml
skinparam classAttributeIconSize 0
class Championnat {
    -nom : String
    ajouterEquipe(equipe : Equipe) : void
    getNombreEquipe() : int
}
class Equipe {
    -nom : String
    -nombreLikes: integer
    incrementerNombreLikes() : void
}
class Personne {
    -nom : String
    liker(equipe : Equipe) : void
}

Championnat --> "1..*" Equipe
Personne ..> Equipe
@enduml
  
```

→ index.php

```
<?php

$dupond = new Personne("Dupond Jean");
$martin = new Personne("Martin pierre");

$liverpool = new Equipe("Liverpool");
$leeds = new Equipe("Leeds United");

$dupond->liker($liverpool);
$dupond->liker($leeds);
$martin->liker($leeds);

echo $liverpool->getNom() . ' a ' . $liverpool->getNombreLikes() . ' like(s)' . PHP_EOL;
echo $leeds->getNom() . ' a ' . $leeds->getNombreLikes() . ' like(s)' . PHP_EOL;

$premierLeague = new Championnat("Premier League");
$premierLeague->ajouterEquipe($liverpool);
$premierLeague->ajouterEquipe($leeds);
echo "Il y a " . $premierLeague->getNombreEquipes() . ' en Premier League';
```



- Utilisation de la fonction **require** de PHP

index.php

```
require './src/Championnat.php';  
require './src/Equipe.php';  
require './src/Personne.php';
```

```
$ php index.php  
Liverpool a 1 like(s)  
Leeds United a 2 like(s)  
Il y a 2 équipes en Premier League
```

1 classe = 1 require



- La solution : utiliser un **autoloader**



- **autoload** : mécanisme permettant de charger automatiquement le code d'une classe sans avoir à utiliser **require** pour chaque classe

- Plusieurs solutions

- Utilisation de **composer**





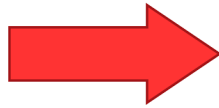
Gestionnaire  
de  
dépendances

<https://getcomposer.org/>

- Initialiser **composer** dans le projet



**composer init**



Création du fichier **composer.json**

- Configurer l'**autoloader** de **composer** afin d'indiquer à composer comment réaliser l'autoloading des classes du projet

 **composer.json**

```
{  
  "require": {},  
  
  "autoload": {  
    "psr-4": {  
      "App\\": "src/"  
    }  
  }  
}
```

- **psr-4** : standard PHP
- App : un **namespace**

- Namespace = espace de noms
- Un espace de noms permet de regrouper dans un emplacement logique (identifié par un nom) un ensemble de classes
- Cet emplacement logique est lié à un emplacement physique = un dossier du projet





Toutes les classes appartenant au **namespace App** sont dans le **dossier src**

Toutes les classes présentes au niveau du **dossier src** appartiennent au **namespace App**

- Indiquer à **composer** de prendre en compte la configuration de l'**autoloading** définie dans le fichier **composer.json** afin de générer les fichiers d'autoloading

→ **composer.json**

```
{  
  "require": {},  
  "autoload": {  
    "psr-4": {  
      "App\\": "src/"  
    }  
  }  
}
```



**composer dump-autoload**

Génération des  
fichiers

vendor

- composer
  - autoload\_classmap.php
  - autoload\_namespaces.php
  - autoload\_psr4.php
  - autoload\_real.php
  - autoload\_static.php
  - ClassLoader.php
  - LICENSE
  - autoload.php

- Modifier les classes afin de **définir le namespace** auquel elles appartiennent
- Instruction PHP **namespace**



```
namespace App;
```



Pour chaque classe

**App** : nom du namespace

→ index.php

```
<?php
// PLUS DE REQUIRE !

$dupond = new Personne("Dupond Jean");
$martin = new Personne("Martin pierre");

$liverpool = new Equipe("Liverpool");
$leeds = new Equipe("Leeds United");

$dupond->liker($liverpool);
$dupond->liker($leeds);
$martin->liker($leeds);

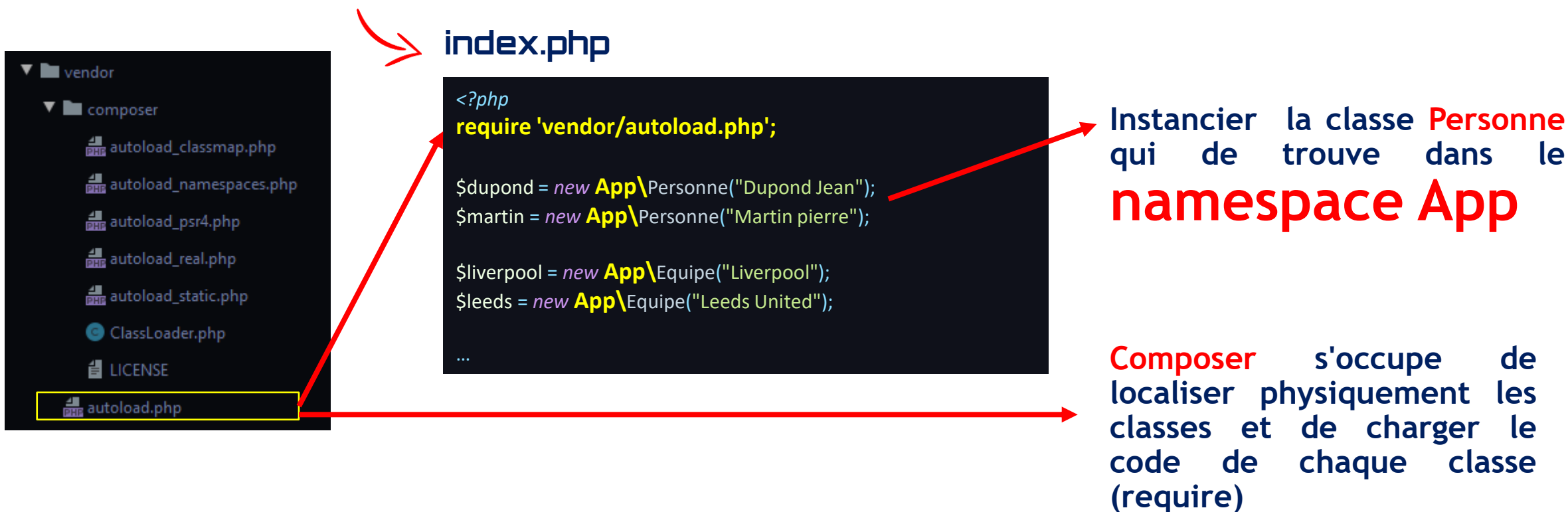
echo $liverpool->getNom() . ' a ' . $liverpool->getNombreLikes() . ' like(s)' . "\n";
echo $leeds->getNom() . ' a ' . $leeds->getNombreLikes() . ' like(s)' . "\n";

$premierLeague = new Championnat("Premier League");
$premierLeague->ajouterEquipe($liverpool);
$premierLeague->ajouterEquipe($leeds);
echo "Il y a " . $premierLeague->getNombreEquipes() . ' en Premier League';
```



## Tout ça pour ça !

- Il faut charger le **fichier d'autoloading** de **composer** !



**index.php**

```
<?php
require 'vendor/autoload.php';

$dupond = new App\Personne("Dupond Jean");
$martin = new App\Personne("Martin pierre");

$liverpool = new App\Equipe("Liverpool");
$leeds = new App\Equipe("Leeds United");

...
```

Instancier la classe **Personne** qui se trouve dans le **namespace App**

**Composer** s'occupe de localiser physiquement les classes et de charger le code de chaque classe (require)

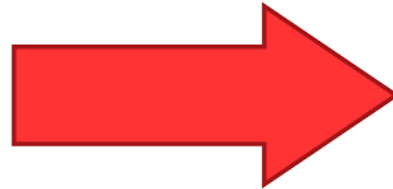
- Instruction PHP **use**

→ index.php

```
<?php
require 'vendor/autoload.php';

$dupond = new App\Personne("Dupond Jean");
$martin = new App\Personne("Martin pierre");

$liverpool = new App\Equipe("Liverpool");
$leeds = new App\Equipe("Leeds United");
...
```



→ index.php

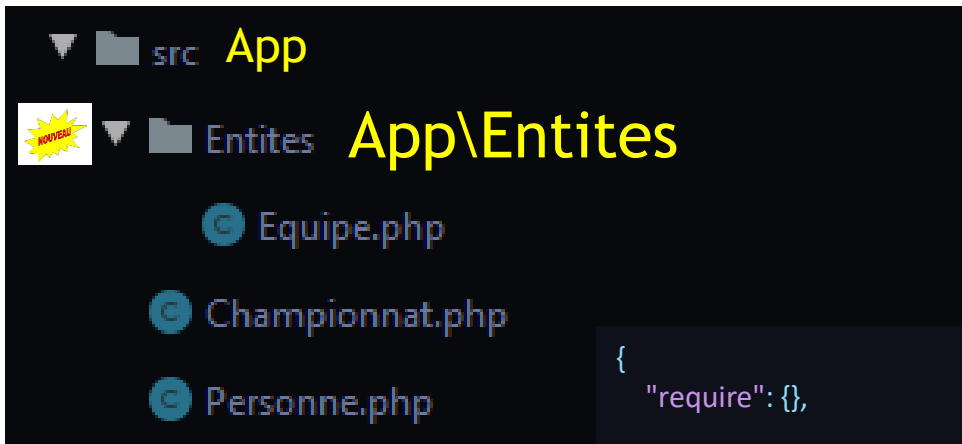
```
<?php
require 'vendor/autoload.php';
use App\Personne;
use App\Equipe;

$dupond = new Personne("Dupond Jean");
$martin = new Personne("Martin pierre");

$liverpool = new Equipe("Liverpool");
$leeds = new Equipe("Leeds United");
...
```

**use n'est pas un require !**

- **PSR-4** permet de suivre la structure des dossiers pour le **nommage des namespaces**



```
{  
  "require": {},  
  "autoload": {  
    "psr-4": {  
      "App\\": "src/"  
    }  
  }  
}
```

composer.json

Dans **Equipe.php** :

```
namespace App\Entites;
```

Dans **index.php** :

```
use App\Entites\Equipe;
```



## Personne.php

```
<?php
namespace App;

class Personne
{
    private $nom;

    /**
     * Personne constructor.
     * @param $nom
     */
    public function __construct($nom)
    {
        $this->nom = $nom;
    }

    public function liker(Equipe $equipe)
    {
        $equipe->incrementerNombreLikes();
    }
}
```

Référence à la classe **Equipe**



Par défaut, quand une classe fait référence à une autre classe, cette dernière est **recherchée** dans le **même namespace**

Personne -> **App**

Equipe -> **App\Entites**

L'**autoloader** va donc essayer de charger la classe **Equipe** qui se trouve dans **src**



## Personne.php

```
<?php
namespace App;

use App\Entites\Equipe;

class Personne
{
    private $nom;

    /**
     * Personne constructor.
     * @param $nom
     */
    public function __construct($nom)
    {
        $this->nom = $nom;
    }

    public function liker(Equipe $equipe)
    {
        $equipe->incrementerNombreLikes();
    }
}
```

Nécessaire d'indiquer dans **quel namespace** se trouve la classe **Equipe** !



L'**autoloader** va donc essayer de charger la classe **Equipe** qui se trouve dans **src\Entites**