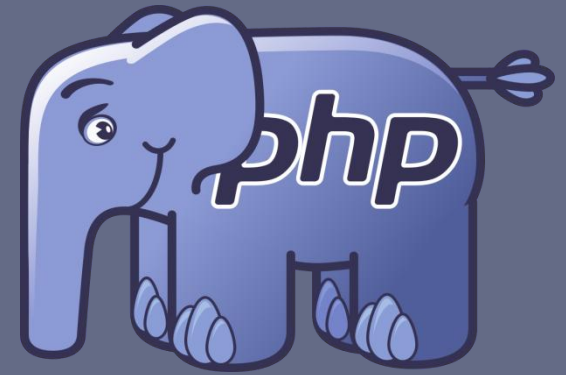


PHP framework



CODEIGNITER



PHP > 7.2

CodeIgniter 4.0.4

Création de pages statiques

Plan

2

- 3 ☐ Pages statiques vs dynamiques
- 5 ☐ Analyse de la page Home
- 12 ☐ Règles de nommage
- 15 ☐ Création d'une nouvelle page
- 21 ☐ La page d'accueil du site « Conteneurs »
- 26 ☐ Créer un template

Pages statiques vs dynamiques



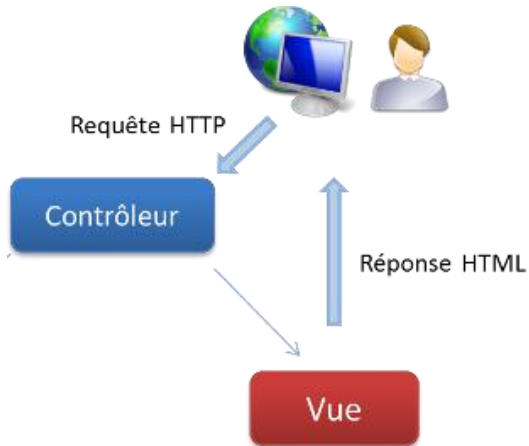
CodeIgniter



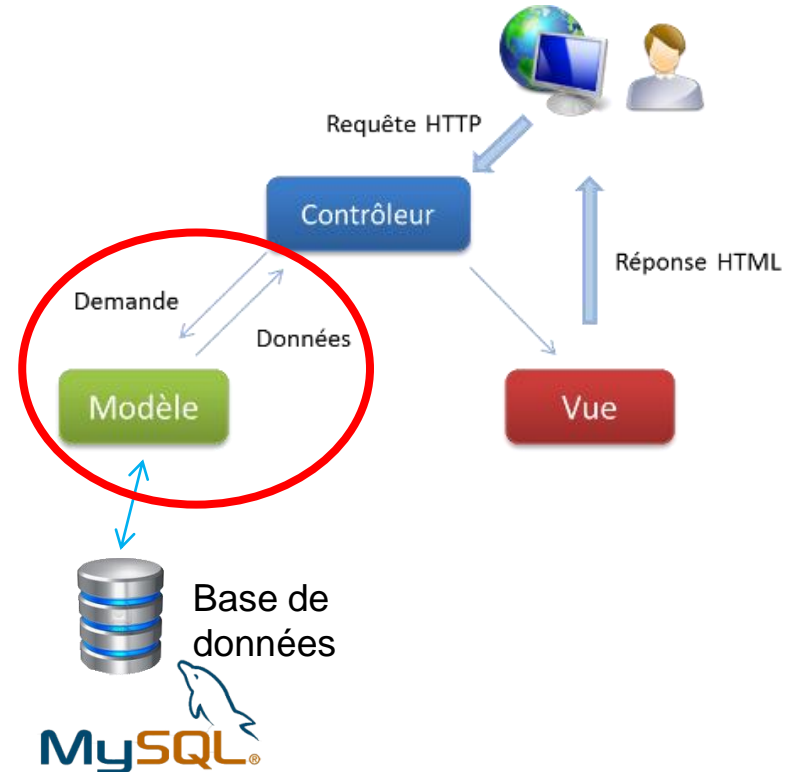
Pages statiques vs dynamiques

4

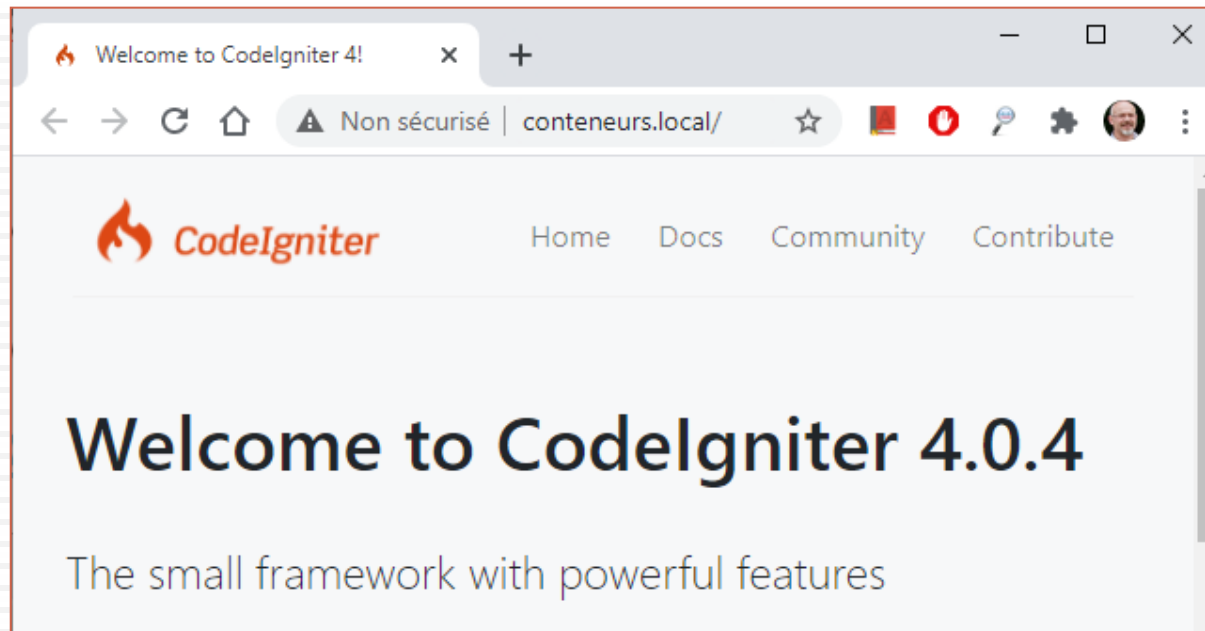
Page statique



Page dynamique



Analyse de la page Home



URLs pour accéder à la page Home

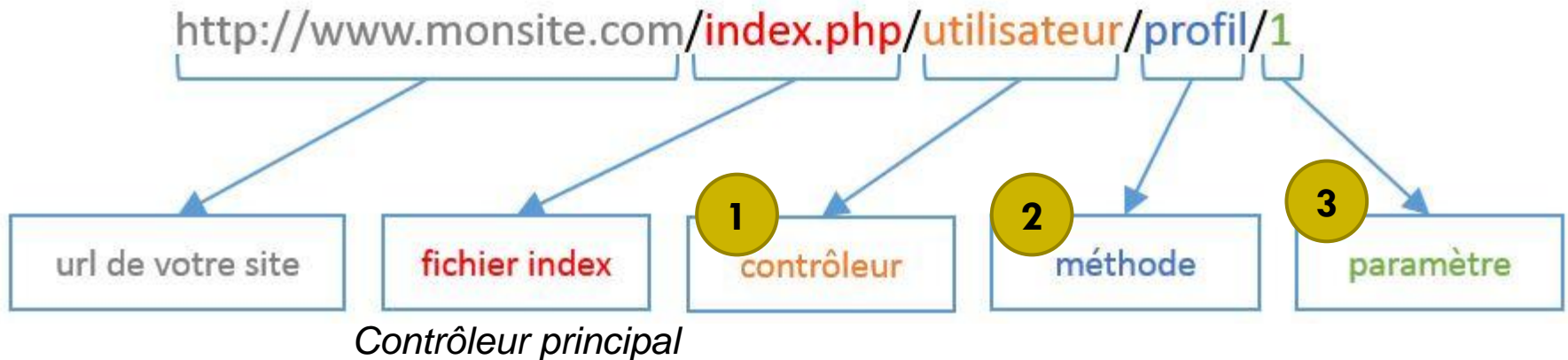
6

- La page d'accueil (Home) du site CodeIgniter est :
 - ▣ <http://conteneurs.local/>
- Mais de façon plus complète cela correspond à :
 - ▣ <http://conteneurs.local/index.php>
- Et en fait c'est un raccourci vers :
 - ▣ <http://conteneurs.local/index.php/Home/index>
 - ▣ Ou
 - ▣ <http://conteneurs.local/Home/index>

Vous pouvez tester, toutes ces URLs sont équivalentes !

Les URLs avec CodeIgniter

7



- 1 **Segment 1** : correspond à la **classe contrôleur** que vous voulez appeler
- 2 **Segment 2** : correspond à la **méthode du contrôleur** que vous voulez appeler
- 3 **Segment 3** : correspond au **paramètre** que recevra la méthode

Index.php non présent dans l'URL

8

- Si le nom du contrôleur principal est omis dans l'URL

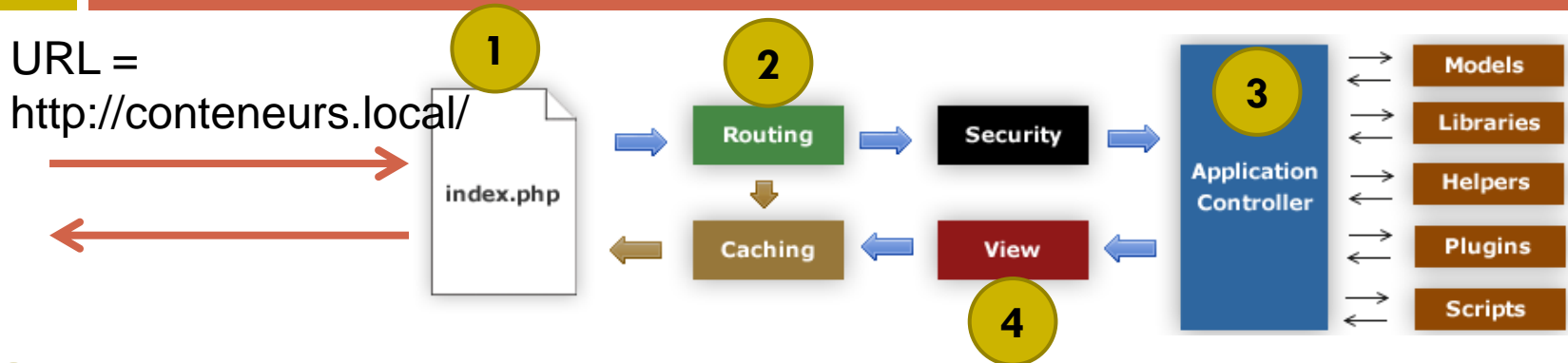
`http://conteneurs.local/index.php/Home/index`

Apache le rajoute automatiquement grâce au fichier de configuration d'Apache pour ce site :
/public/.htaccess

DONC, le contrôleur principal est toujours appelé AVANT l'exécution de votre contrôleur /app/Controllers/xxx

Comment ça marche pour la page Home

9



1 Le contrôleur principal initialise CodeIgniter
`/public/index.php`

2 Le routage analyse l'URL pour instancier le contrôleur par défaut
`/app/Config/Routes.php`

```
$routes->get('/', 'Home::index');
```

3 La fonction `index()` du contrôleur
« Home » est exécuté
`/app/Controllers/Home.php`

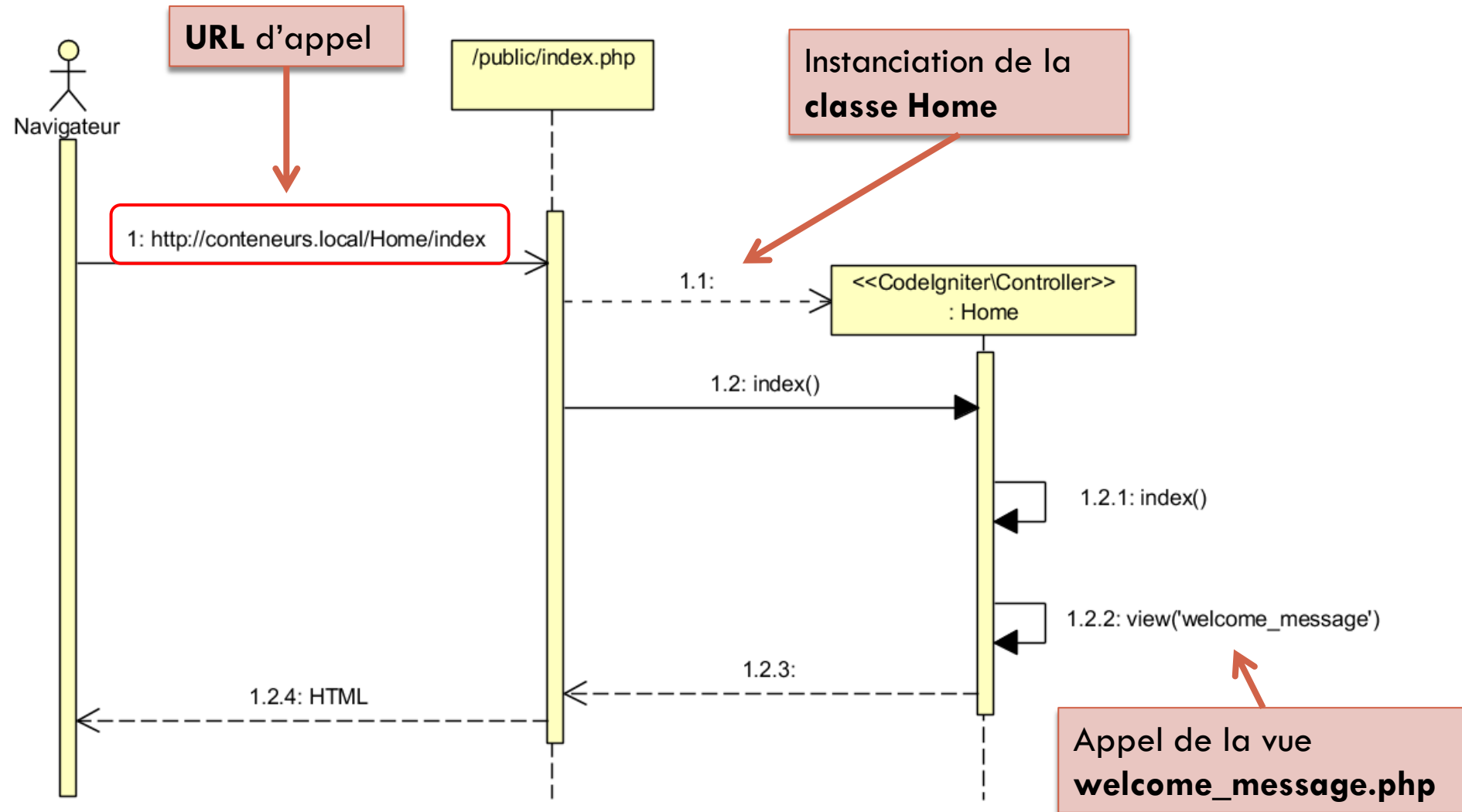
```
class Home extends BaseController {  
    public function index() {  
        return view('welcome_message');
```

4 La vue « `welcome_message` » est renvoyée au navigateur
`/app/Views/welcome_message.php`

HTML et PHP

UML : fonctionnement pour Home

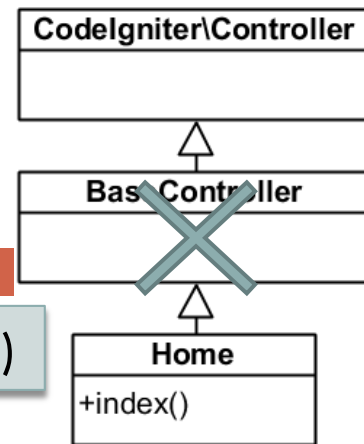
10



Le contrôleur « Home »

11

Si on fait abstraction de la classe BaseController (qui n'est pas utile ici)



/app/Controllers/Home.php

```
<?php namespace App\Controllers;
use CodeIgniter\Controller;

class Home extends Controller
{
    public function index()
    {
        return view('welcome_message');
    }
}
```

Toutes les classes Contrôleur doivent :

- être créées dans le dossier /app/Controllers
- faire partie du namespace App\Controllers
- importer le namespace CodeIgniter\Controller
- dériver de CodeIgniter\Controller

- La méthode index() est exécutée par défaut
- La fonction view() charge la vue, fichier « welcome_message.php » situé dans le dossier /app/Views

La vue « welcome_message »

12

/app/Views/welcome_message.php

```
<!DOCTYPE html>
<html lang="en">
  ...
  <h1>Welcome to CodeIgniter ...</h1>
  ...
</html>
```

- Une vue est tout simplement un **fichier .php** qui va contenir le code html de votre page.
- L'idée est d'**appeler la vue depuis le contrôleur** afin de lui envoyer des données.
- La vue se chargera d'intégrer ces données dans le **format HTML classique**.



CodeIgniter

Règles de nommage CodeIgniter

14

PSR-1: Basic Coding Standard

- ❑ Les noms de fichiers contenant des classes doivent correspondre au nom de la classe
- ❑ **Nom des classes et des namespaces** en CamelCase avec la première lettre en Majuscule (**S**omeLibrary.php)
- ❑ **les autres noms de fichiers** (views ..) en snake_case minuscule (**a**pplication_config.php)
- ❑ **Les noms de méthodes et variables** doivent être en CamelCase avec la première lettre en minuscule (function **g**et**F**ile**P**roperties())

Règles spécifiques à ce cours

15

- Nom des **contrôleurs** préfixé par **C**
class Caccueil dans le fichier Caccueil.php
- Nom des **modèles** préfixé par **M**
class Maccueil dans le fichier Maccueil.php
- Nom des **vues** préfixé par **v_**
fichier v_accueil.php

Création d'une nouvelle page



CodeIgniter

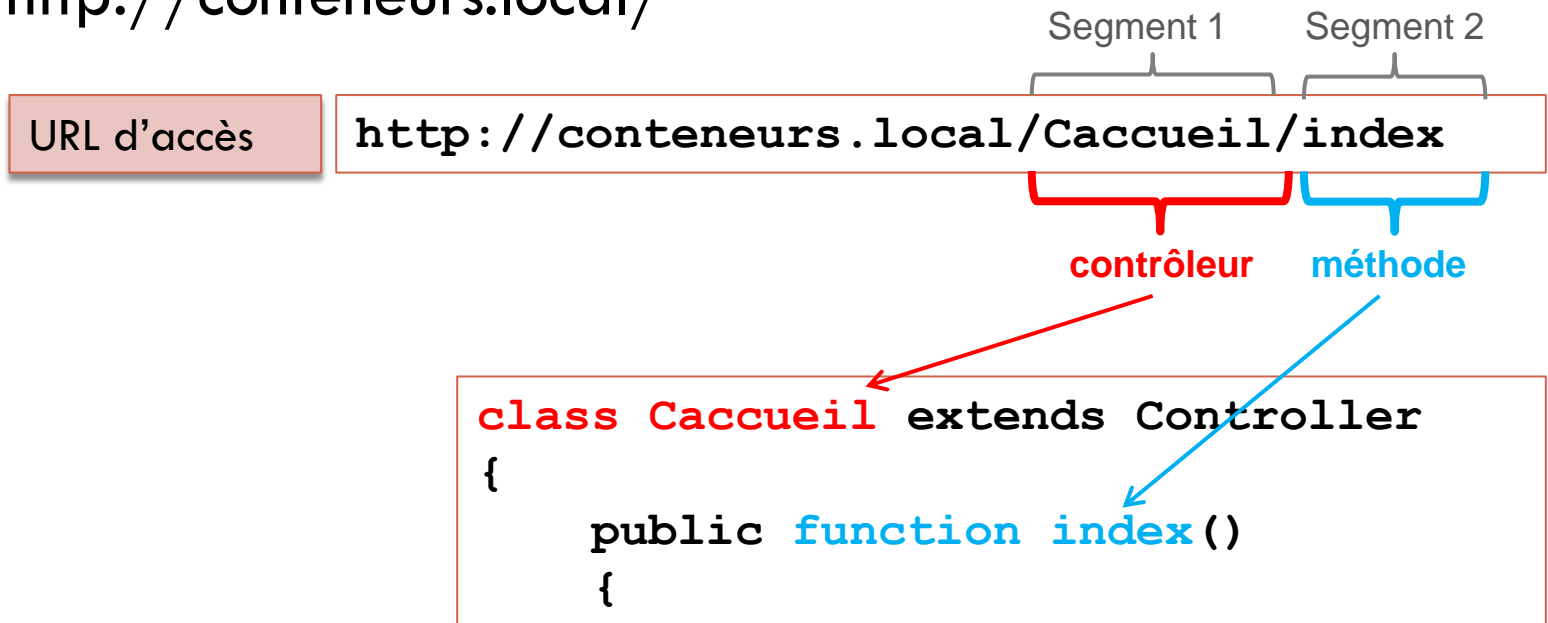
```
php spark make:controller Caccueil --bare
```


Création d'une page d'accueil



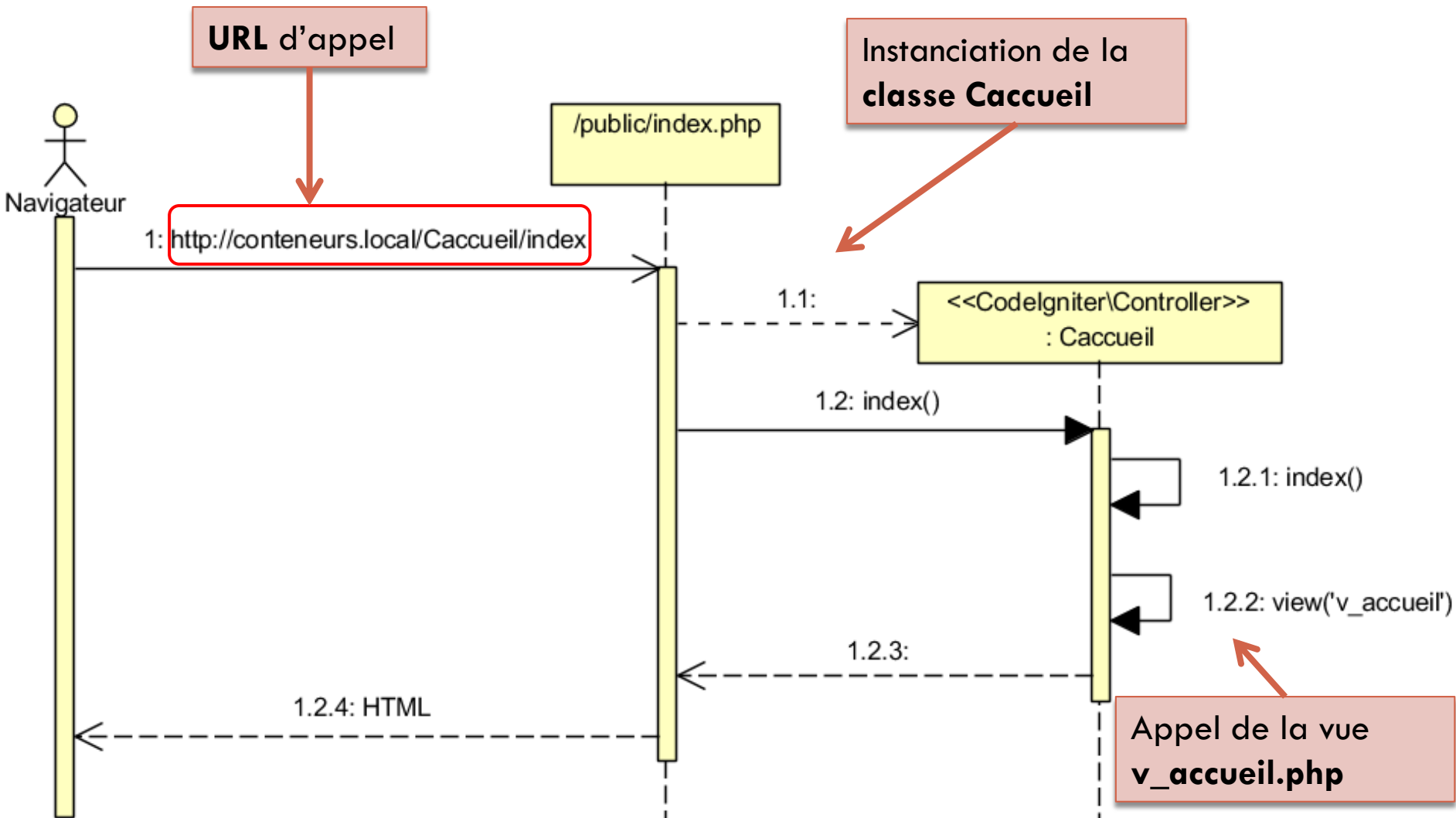
17

- Créez une nouvelle page d'accueil :
 - ▣ Contrôleur : **Caccueil.php**
 - ▣ Vue : **v_accueil.php** (html5 avec un message « coucou »)
 - ▣ Faites en sorte qu'elle s'affiche par défaut pour l'URL :
`http://conteneurs.local/`



UML : fonctionnement pour Caccueil

18



Passer des données à la vue

19

- Les données du contrôleur sont passées à la vue grâce à un **tableau associatif (=paires de clés/valeurs)**

Contrôleur



Vue

```
$data['title'] = "My Real Title";  
$data['heading'] = "My Real Heading";  
  
return view('v_accueil', $data);
```

1^{er} param : la vue

2^{ème} param : les données

```
...  
<title><?php echo $title;?></title>  
...  
<h1><?php echo $heading;?></h1>  
...
```

Les clés du tableau **\$data** deviennent des noms de variable dans la vue

Remarque de syntaxe PHP :

<?php echo \$title; ?> peut aussi s'écrire **<?= \$title; ?>**

Passer des données à v_accueil



20

- Modifiez la page d'accueil pour qu'elle affiche les données venant de son contrôleur :
 - ▣ **\$page_title** contiendra le titre de l'onglet
"Les conteneurs de vêtements"
 - ▣ **\$titre1** contiendra le titre principal `<h1>`
"Un projet pilote de borne connectée"



La page d'accueil du site « Conteneurs »

LE RELAIS

Ramassage des points de collecte



Accueil Les tournées de collecte Les conteneurs

« Un projet pilote de borne connectée »



Si vous déposez régulièrement vos vieux vêtements dans les bornes "Le Relais" des compagnons d'Emmaüs, il vous est peut-être déjà arrivé, comme moi, de tomber sur une borne pleine, dans laquelle il n'était plus possible de déposer quoi que ce soit... Pour éviter cette situation, une scop, membre d'Emmaüs, a décidé de lancer un projet pilote de Relais connecté. Située à Chanteloup-les-Vignes, EBS Le Relais Val de Seine développe une activité de réinsertion sociale qui, tout comme les autres scop de l'union des Relais, est centrée sur la collecte, le tri, et la revalorisation des vieux textiles.

Jusqu'ici, la scop utilisait un logiciel maison d'optimisation des tournées. Il fonctionne sur la base d'indicateurs de fréquence moyenne de remplissage calculés à partir de données saisies par les salariés lors de leurs tournées de collecte, via PDA. "Nous testons actuellement des capteurs connectés qui nous permettrait de suivre le niveau de remplissage de nos bornes en temps réel", confirme Jean-François Luthun, PDG d'EBS Le

Relais Val de Seine. En ligne de mire : disposer d'un outil permettant de faire face à tout impondérable pouvant engendrer une saturation de la borne (une braderie, un vide grenier, un déménagement à proximité...).

« Que deviennent vos dons ? »



Une valorisation de vos vêtements à 97%.

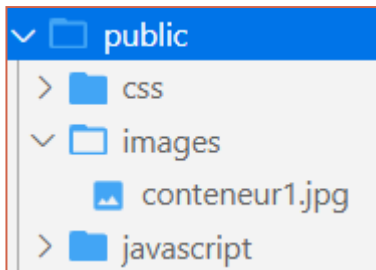
Grâce à vos dons, le Relais récupère chaque semaine plus de 1 800 tonnes de textiles, ce qui représente 55 % de la collecte en France. Ces textiles sont acheminés vers 14 centres de tri, pour connaître différents sorts selon leur état et leur qualité :

- Les textiles utilisables en l'état sont revendus à bas prix dans les 73 boutiques Ding Fring du Relais ou destinés à l'export.
- Les textiles qui ne peuvent plus être portés sont recyclés par le Relais dans le cadre de la production de chiffons d'essuyage pour l'industrie, ou pour la fabrication de matières à partir desquelles est notamment fabriqué l'isolant Métisse®.
- La matière non valorisée représente 3 % des volumes collectés, dont la majeure partie sera utilisée en valorisation énergétique.

Problème des liens web <a>...

22

- Le **code HTML des vues** contient des liens vers des fichiers web : images, css, javascript.
- Ces fichiers sont déposés dans le dossier **/public**
- **Les vues ne peuvent pas toujours y accéder à cause des URLs de CodeIgniter**



/app/Views/v_accueil.php



http://conteneurs.local/



http://conteneurs.local/Caccueil/index

« URL Helper »

23

Dans CodeIgniter, les helpers sont des mini-bibliothèques qui contiennent des fonctions utilitaires.

- *Le URL helper est toujours chargé automatiquement*
- Quelques fonctions utiles de ce helper :
 - ▣ **echo base_url()** affiche l'URL de base du site
http://conteneurs.local
 - ▣ **echo base_url("img/maphoto.jpg")** affiche
http://conteneurs.local/img/maphoto.jpg
 - ▣ **echo site_url()** affiche l'URL de base du site avec index.php
http://conteneurs.local/index.php

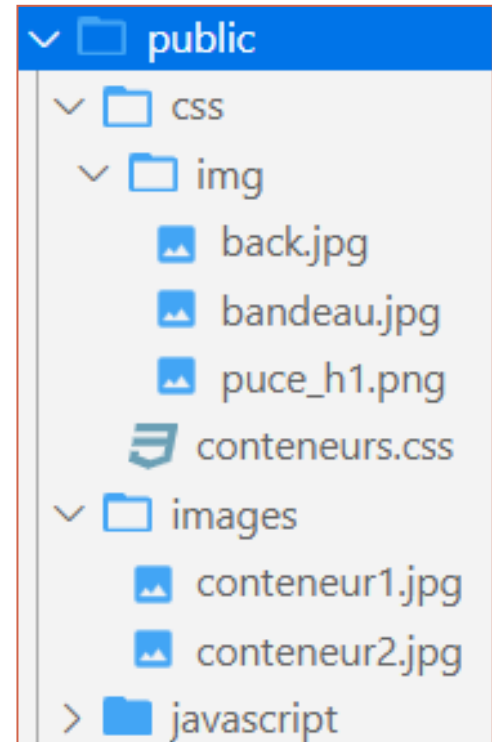
Utilisation de l'URL Helper



24

□ Modifiez **v_accueil.php** pour que la page d'accueil affiche la même chose que le modèle du site fourni.

- Copiez les dossiers css et images dans /public comme indiqué
- Insérer le code HTML dans la vue v_accueil.php
- **Modifiez les liens vers les fichiers css et images avec base_url()**
- Gardez le passage des paramètres depuis Caccueil et ajoutez une variable \$titre2



Testez avec l'URL :
<http://conteneurs.local/Caccueil/index>

25

Créer un template



CodeIgniter

Objectif d'un template

26

- Ne pas dupliquer le code commun à toutes les pages

1 Chaque page ne contient que son code spécifique

/app/Views/**v_accueil.php**

```
<article>
  <h1>Un projet... </h1>
  <p>...</p>
</article>
<article>
  ...
</article>
```

/app/Views/**Commun/v_template.php**

```
<!DOCTYPE html>
<html>
  <head>...</head>
  <body>
    <div id="conteneur">
      <header>...</header>
      <nav>...</nav>
      <section>
```

<?php echo **\$contenu**; ?>

2

Le code de la page est intégré dans le modèle unique

```
    </section>
    <footer>...</footer>
  </div>
</body>
</html>
```

Le contrôleur charge la page puis l'intègre dans le template

27

C'est la même technique que pour passer des données à une vue

1. Charger le contenu de la vue **v_accueil** dans un **tableau associatif \$page**

```
$page['contenu'] = view('v_accueil', $data);
```

2. Charger la vue **v_template** en lui passant le tableau associatif **\$page**

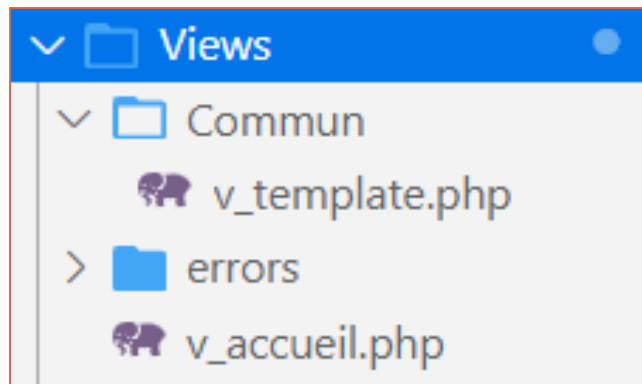
```
return view('Commun/v_template', $page);
```

Créez et utilisez le template



28

- Créez la vue **v_template.php** dans un sous dossier



- Modifiez les contenus de **v_template** et **v_accueil**
- Modifiez le contrôleur **Caccueil**

