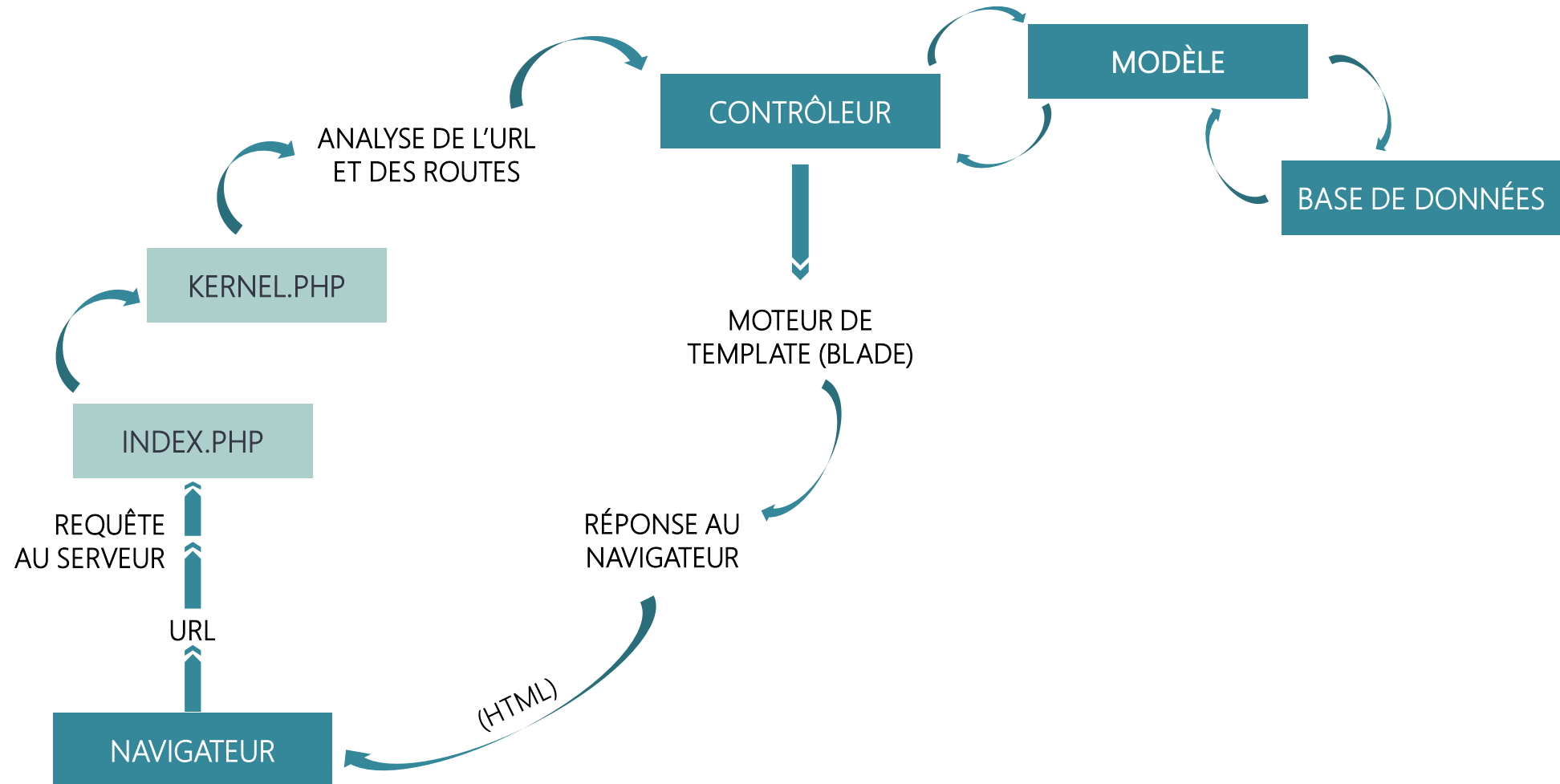


Chapitre 3 : Routage / Contrôleurs

Comprendre et utiliser le mécanisme de routage afin de faire correspondre une URI à une action

Requête Laravel



Requête Laravel

- Consultation via une URI
Ex : **/video/15-avatar.html**
- Le serveur Web intercepte la requête et consulte index.php grâce à la réécriture d'URL
- Index.php charge le Kernel, qui appelle le routeur
- Le routeur tente de faire matcher l'URI saisie avec une correspondance présente dans la configuration
- Si une route correspond : appel du contrôleur et de l'action associée

Routeur Laravel

- Routeur écrit en PHP
- Utilisation automatique de la Facade « Route »
- /routes/web.php
- La fonction utilisée exprime le verbe HTTP correspondant

```
Route::get('/', 'HomeController@index');
```

Routeur Laravel

- Possibilité d'écrire une closure au lieu d'appeler un contrôleur (micro-api, fonction simple, déclenchement de mail ...)

```
Route::get('foo', function () {  
    return 'Hello World';  
});
```

Contrôleurs

- /app/Http/Controllers/
- Ne pas oublier le namespace
- Ils étendent Illuminate\Routing\Controller
- Les actions ne nécessitent pas de suffixe (contrairement à Symfony)
- Génération d'un contrôleur
- Possibilité de créer un contrôleur lié à un modèle (-m=XXX) ou contrôleur CRUD (-r)

```
php artisan make:controller TotoController
```

Action

- Une action = une fonction dans le contrôleur
- Retour d'une action
 - Retour d'une response (redirection)
 - Si interdiction : `abort(404);`
 - Retour d'une view (*return view()*): affichage HTML (via Blade)
 - Retour de texte : affichage brut
 - Retour de tableau : affichage JSON
 - Retour d'une instance de modèle : affichage JSON
 - Aucun retour : page blanche (pas d'erreur)

Routing

- Redirection via router

```
Route::redirect('/news', '/actualites', 301);
```

- Route vers vue

```
Route::view('/welcome', 'welcome', ['name' => 'Taylor']);
```

- Nommage des routes

```
Route::get('/test', 'TestController@test')->name('test');
```


Routing - Paramètres

- Paramètre obligatoire

```
Route::get('/news/{id}', 'TestController@test')->name('test');
```

- Côté contrôleur

```
public function test($id)
{
    dd($id);
}
```

Routing - Paramètres

- Paramètre facultatif

```
Route::get('/news/{id?}', 'TestController@test')->name('test');
```

- Côté contrôleur

```
public function test($id=0)
{
    dd($id);
}
```

Routing - Paramètres

- Expression régulière

```
Route::get('/test/{id?}', 'TestController@test')  
    ->where('id', '[0-9]+')  
    ->name('test');
```

```
Route::get('/test/{id}/{name}', 'TestController@test')  
    ->where([  
        'id' => '[0-9]+',  
        'name' => '[a-z]+'    ])  
    ->name('test');
```

Routing - Paramètres

- Contrainte globale sur un nom de paramètre
- Dans le fichier RouteServiceProvider.php

```
public function boot()  
{  
    Route::pattern('id', '[0-9]+');  
    parent::boot();  
}
```

Reverse routing

- Principe : générer une URL à partir d'un nom de route
- Utile s'il est nécessaire de modifier par la suite, l'URL d'une page
- Les liens seront automatiquement mis à jour

```
Route::get('/test/{id}', 'TestController@test')->name('test');
```

```
$url = route('test', ['id' => 1]);  
return response()->redirectToRoute('test', ['id' => 1]);
```

Grouper les routes, préfixes, ...

```
Route::prefix('admin')
    ->name('admin.')
    ->namespace('Admin')
    ->group(function () {
        Route::get('/test/{id}', 'TestController@testAdmin')->name('testAdmin');
    });
```

Gestion de sous-domaines

```
Route::domain('admin.monsite.com')->group(function () {  
    Route::get('/test/{id}', 'TestController@testAdmin')->name('testAdmin');  
});
```

TP 2

Mise en place des principales pages