

فهاد الجزء الرابع والأخير غادي نكمولو الجزء الثاني من الملف Auth.php والملف لي غادي تكون فيه الفورم لي غادي تمكن المستخدم يتكونيكطا والملف لي غادي يكون فيه routes ديالنا باش نتمكنو من التنقل بين الصفحات ونفدو ل actions لي زدنا.

1- الملف Auth.php الجزء الثاني

فهاد الجزء الثاني من الملف Auth.php كايين fonction registerForm ولي فيها form validation ديال الحقول ولي كتحقق من form لي ترسلت من الملف login.php ولي كتمكن المستخدم باش يتسجل فقاعدة البيانات بلي جميع الحقول كتستاجب لهاد الشروط فيلا كانو كيستاجبو كترد true مكانش كترجع رسالة خطأ كتخص كل حقل ثم كايين processRegister لي من بعد مكتسترجع الحقول كتزيردهم وكتزيد المعلومات ديال المستخدم ف session وبالتالي كيتكونيكطا اوتوماتيكيا ثم كايين logoutAction لي كتخرج المستخدم وديكونيكطيه وكتخوي session الكود ديال الملف هو :

```
/**
 * Build the registration form
 * @return Form
 */
protected function registerForm()
{
    $validate = $this->components()->validate();
    $validator = $validate->validator();
    $document = $validator->rule()->addDocument();
    // By default the validator won't allow any fields that were not
defined.
    // This call turns off this validation and allows extra fields to be
passed.
    // In our case the extra field is the hidden "register" field.
    $document->allowExtraFields();
    // Name is required and must be at least 3 characters long
    $document->valueField('name')
        ->required("Nom est obligatoire.")
        ->addFilter()
        ->minLength(3)
        ->message("Nom doit avoir au moins 3 caractères");

    // Email is required and must be a valid email
    $document->valueField('email')
        ->required("Email est obligatoire.")
        ->filter('email', "Veuillez entrer une adresse email valide");

    $document->valueField('password')
        ->required("Passe est obligatoire.")
        ->addFilter()
        ->minLength(8)
```

```

        ->message("Passe doit avoir au moins 8 caractères");

        $document->valueField('passwordConfirm')
            ->required("Confirmez votre passe");

        // In this callback rule we check that password confirmation matches
the password
        $validator->rule()->callback(function($result, $value) {
            // If they don't match we add an error to the field
            if($value['password'] !== $value['passwordConfirm']) {
                $result->field('passwordConfirm')->addMessageError("les deux
mots de passe ne sont pas identiques");
            }
        });
        // Build a form for this validator
        return $validate->form($validator);
    }
/**
 * Process registration
 * @param Form $registerForm
 * @return bool Whether the user was successfully registered
 */
protected function processRegister($registerForm)
{
    /** @var UserRepository $userRepository */
    $userRepository = $this->components()->orm()->repository('user');
    // Check if the email already exists and if so add an error to the
form
    if($userRepository->getByLogin($registerForm->email)) {
        $registerForm->result()->field('email')->addMessageError("Email
déjà utilisé");
        return false;
    }
    // Hash password and create the user
    $provider = $this->passwordProvider();
    $user = $userRepository->create([
        'name' => $registerForm->name,
        'email' => $registerForm->email,
        'passwordHash' => $provider->hash($registerForm->password)
    ]);
    $user->save();
    // Manually log the user in
    $provider->setUser($user);
    $_SESSION['defaultUserId'] = $user->id;
    $_SESSION['username'] = $user->name;
    return true;
}

/**
 * Log the user out
 * @return mixed

```

```

    */
public function logoutAction()
{
    // Get auth domain and forget the user
    $domain = $this->components()->auth()->domain();
    $domain->forgetUser();
    session_destroy();
    // Then redirect to the homepage
    $http = $this->builder->components()->http();
    $httpResponses = $http->responses();
    return $httpResponses->redirect('/');
}
/**
 * The password login provider that we configured
 * /assets/config/auth.php
 * @return Password
 */
protected function passwordProvider()
{
    $domain = $this->components()->auth()->domain();
    return $domain->provider('password');
}
}

```

-2 الملف login.php

فالمجلد bundles/app/assets/templates زيد ملف جديد سميهِ login.php هادا الملف لي غادي تكون فيه الفورم لي غادي تمكن المستخدم من التسجيل فقاعدة البيانات بواسطة fonction defaultAction لي زدناها فالملف Auth.php و الفورم لي غادي تمكن المستخدم من تسجيل الدخول بواسطة fonction defaultAction دائما لي زدناها فالملف Auth.php وأفول الملف كايين الكود لي كيمكن من إضافة الصفحة الأساسية لي فيها القائمة ديال التصفح للملف ديالنا الكود ديال الملف هو:

```

<?php
$this->layout('app:layout');?>
<div class="container content">
    <div class="row">
        <div class="col-md-4 col-md-offset-1">
            <!-- Sign in form -->
            <form method="POST" action="<?=$this->httpPath('app.processor',
[ 'processor' => 'auth' ])?>">
                <h2 class="text-default">Connexion</h2>
                <!-- email field -->
                <?php if($error = $loginForm->fieldError('email')): ?>
                    <div class="text-danger"><?=$error?></div>
                <?php endif;?>
                <div class="form-group"
<?=$this->if($loginForm->fieldError('email'),"alert alert-danger"?>">

```

```

                <input name="email" type="text"
value="<?=$_($loginForm->fieldValue('email'))?>"
                class="form-control" placeholder="Email">
            </div>
            <?php if($error = $loginForm->fieldError('password')): ?>
                <div class="text-danger"><?=$error?></div>
            <?php endif;?>
            <!-- password field -->
            <div class="form-group
<?=$this->if($loginForm->fieldError('password'), "alert alert-danger")?>">
                <input name="password" type="password" class="form-
control" placeholder="Passe">
            </div>
            <!-- Placeholder for additional error messages -->
            <?php if($error = $loginForm->resultError()): ?>
                <div class="form-group">
                    <div class="alert alert-danger"><?=$error?></div>
                </div>
            <?php endif;?>
            <button class="btn btn-success"
type="submit">Connexion</button>
        </form>
    </div>
    <div class="col-md-4 col-md-offset-2">
        <!-- Registration form -->
        <form method="POST" action="<?=$this->httpPath('app.processor',
['processor' => 'auth'])?>">
            <h2>Inscription</h2>
            <!-- name field -->
            <?php if($error = $registerForm->fieldError('name')): ?>
                <div class="text-danger"><?=$error?></div>
            <?php endif;?>
            <div class="form-group
<?=$this->if($registerForm->fieldError('name'), "alert alert-danger")?>">
                <input name="name" type="text"
value="<?=$_($registerForm->fieldValue('name'))?>"
                    class="form-control" placeholder="Nom">
            </div>
            <?php if($error = $registerForm->fieldError('email')): ?>
                <div class="text-danger"><?=$error?></div>
            <?php endif;?>
            <!-- email field -->
            <div class="form-group
<?=$this->if($registerForm->fieldError('email'), "alert alert-danger")?>">
                <input name="email" type="text"
value="<?=$_($registerForm->fieldValue('email'))?>"
                    class="form-control" placeholder="Email">
            </div>
            <?php if($error = $registerForm->fieldError('password')): ?>
                <div class="text-danger"><?=$error?></div>
            <?php endif;?>

```

```

        <!-- password field -->
        <div class="form-group
<?=$this->if($registerForm->fieldError('password'), "alert alert-danger")?>">
            <input name="password" type="password" class="form-
control" placeholder="Passe">
        </div>
        <?php if($error =
$registerForm->fieldError('passwordConfirm')): ?>
            <div class="text-danger"><?=$error?></div>
        <?php endif;?>
        <!-- password confirmation field -->
        <div class="form-group
<?=$this->if($registerForm->fieldError('passwordConfirm'), "alert alert-
danger")?>">
            <input name="passwordConfirm" type="password"
class="form-control" placeholder="Passe confirmation">
        </div>
        <!-- placeholder for additional errors -->
        <?php if($error = $registerForm->resultError()): ?>
            <div class="alert alert-danger">
                <div class="text-danger"><?=$error?></div>
            </div>
        <?php endif;?>
        <!-- We will check for this flag to see if submitted form is
registration or login -->
        <input type="hidden" name="register" value="1">
        <button class="btn btn-lg btn-primary btn-block"
type="submit">Valider</button>
    </form>
</div>
</div>
</div>

```

ORM.php الملف -3

فباش ل framework يعرف ل Model User و Controller UserRepository لي زدنا ف bundles/src/ORM لازم ما نزيدوهم فالملف ORM.php ولي كاين ف bundles/app/src الكود ديال الملف هو:

```

<?php

namespace Project\App;

/**
 * Here you can define wrappers for the ORM to use.
 */
class ORM extends \PHPixie\DefaultBundle\ORM
{

```

```

protected $entityMap = array(
    'user' => 'Project\App\ORM\User'
);

protected $repositoryMap = [
    'user' => 'Project\App\ORM\User\UserRepository'
];
}

```

4- الملف routeResolver.php

فالمجلد bundles/app/assets/config routeResolver.php كايں الملف فيه les routes ديالنا ولي زدنا ولي كيمكنو من التوجه للصفحة الخاصة بالأدمن ولي كيمكن من مشاهدة رابط وتعديل رابط وإضافة رابط وباش تمكن من التنقل ما بين الصفحات وتستعمل لactions لي زدنا فل controllers لازم ما تكون زايدهم فهاد الملف الكود ديالو بعد التعديل هو :

```

                                                                    <?php
return array(
    'type'      => 'group',
    'defaults'  => array('action' => 'default'),
    'resolvers' => array(
        'action' => array(
            'path' => '<processor>/<action>'
        ),
        'processor' => array(
            'path'      => '<processor>',
            'defaults' => array('processor' => 'weblink')
        ),
        // view route
        'view' => array(
            'path' => 'weblink/view/<id>',
            'defaults' => ['processor' => 'weblink', 'action'=> 'view'],
        ),
        // admin route
        'admin' => array(
            'path' => 'weblink/admin',
            'defaults' => ['processor' => 'weblink', 'action'=> 'admin'],
        ),
        //add route
        'add' => array(
            'path' => 'weblink/add',
            'defaults' => ['processor' => 'weblink', 'action'=> 'add'],
        ),
        //edit route
        'edit' => array(
            'path' => 'weblink/edit/<id>',
            'defaults' => ['processor' => 'weblink', 'action'=> 'edit'],

```

```
),
//delete route
'delete' => array(
    'path' => 'weblink/delete/<id>',
    'defaults' => ['processor' => 'weblink','action'=> 'delete'],
)
);
```