Dynamische websites

Week 11: Authentication & Authorization

Agenda

- auth.php
- klasse voor authenticatie en autorisatie

Probleemsituatie

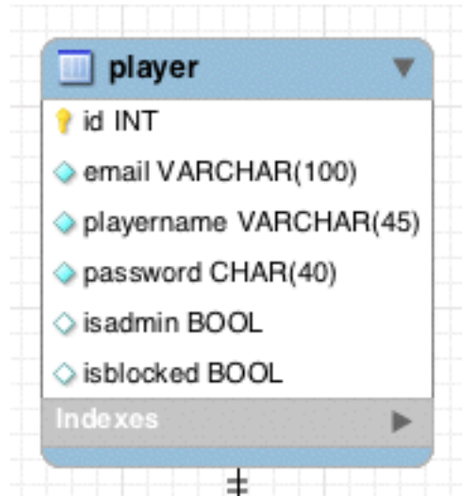
- Wie bezoekt de website?
- Wat mag die persoon doen/zien

1. Authentication

- Wie bezoekt de website?
- Hoe weten we dat Jos effectief Jos is?
- Mag Jos binnen?

Login

- Wie: e-mail adres
- Hoe weten we dat hij het is: wachtwoord
- Mag hij binnen: isBlocked



Login

e-mail Vul je e-mail adres in

wachtwoord

login

Structuur

1. Ophalen van gebruiker voor e-mail adres
2. Controleren of gebruiker bestaat
3. Controleren of wachtwoord correct is
4. Controleren of gebruiker niet geblokkeerd is

HomeController.php

```
public function login($email, $password)
{
    $userForEmail = $this->_userMapper->getPlayerForEmail($email);

    if (!$userForEmail) {
        $message = new Message('Geen gebruiker voor dit e-mail adres gevonden', false);
    } else if ($userForEmail->checkPassword($password)) {
        if ($userForEmail->isblocked) {
            $message = new Message('Je e-mail adres is geblokkeerd door de admin. Je kan niet inloggen.', false);
        } else {
            $_SESSION['user'] = $userForEmail->getId();
            $this->_currentUser = $userForEmail;
            $message = new Message('Succesvol ingelogd');
        }
    } else {
        $message = new Message('Verkeerd wachtwoord', false);
    }

    return $message;
}
```

1. Ophalen

2. Controleren of bestaat

3. Wachtwoord controle

4. Controleren of niet geblokkeerd

system/model/auth.php

```
public function login($email, $password)
{
    $userForEmail = $this->_userMapper->getPlayerForEmail($email);

    if (!$userForEmail) {
        $message = new Message('Geen gebruiker voor dit e-mail adres gevonden', false);
    } else if ($userForEmail->checkPassword($password)) {
        if ($userForEmail->isblocked) {
            $message = new Message('Je e-mail adres is geblokkeerd door de admin. Je kan niet inloggen.', false);
        } else {
            $_SESSION['user'] = $userForEmail->getId();
            $this->_currentUser = $userForEmail;
            $message = new Message('Succesvol ingelogd');
        }
    } else {
        $message = new Message('Verkeerd wachtwoord', false);
    }

    return $message;
}
```

Als controles ok:
- id van gebruiker in sessie
- Gebruikerobject opslaan als instantievariabele

Laden van een pagina

1. Nakijken of gebruikerid in sessie
2. Ophalen van gebruikerobject en in instantievariabele steken

system/model/auth.php

```
private function _initCurrentUser()
{
    if (isset($_SESSION['user'])) {
        $this->_currentUser = $this->_userMapper->get($_SESSION['user']);
    } else {
        $this->_currentUser = false;
    }
}
```

1. Nakijken of userid in sessie

2. Ophalen userobject en in instantievariabele steken

Logout

1. Wissen van instantie van gebruikersobject
2. Wissen van gebruikersid in sessie

HomeController.php

```
public function logout()
{
    $this->_auth->logout();
    $this->_setStatusMessage(new Message('Succesvol uitgelogd'));
    redirect('home/index');
}
```

system/model/auth.php

```
public function logout()
{
    $this->_currentUser = false;

    if (isset($_SESSION['user'])) {
        unset($_SESSION['user']);
    }

    return true;
}
```

1. Wissen instantieobject

2. Wissen gebruikersid in sessie

2. Authorization

- Tonen we bepaalde opties om te doen?
- Mag Jos doen wat hij wil doen?
  - controllers / methods afschermen

Structuur

- Als we weten dat gebruiker Jos is (gebruiker is aangemeld)
    - Kijken naar userlevel (isadmin)
- Als niet aangemeld
    - In laten loggen

Implementatie

- Verschillende implementatie voor
  - Controller
  - Template

In controller

- Nakijken of ingelogd
  - Als niet aangemeld: redirect naar login pagina
  - Als aangemeld: nakijken niveau
    - Als niet juiste niveau: loggen + redirect

system/controller/ Controller.php

```
protected function _checkIfUserHasRequiredAccessLevel($requiredAccessLevel)
{
    if ($this->_checkIfUserIsLoggedIn()) {
        if ($this->_auth->getUserAccessLevel() >= $requiredAccessLevel) {
            return true;
        } else {
            $this->_setStatusMessage(new Message('Je hebt niet het juiste gebruikersniveau.', false));

            // todo: eventueel loggen van poging

            redirect('home/index');
        }
    }

    return false;
}

protected function _checkIfUserIsLoggedIn()
{
    if ($this->_auth->getCurrentUser()) {
        return true;
    } else {
        $this->_setStatusMessage(new Message('Log in om deze pagina te bekijken.', false));
        redirect('home/login');
    }

    return false;
}
```

In template

- Dingen tonen als juiste niveau
- Als niet aangemeld: niets tonen of login link

system/view/ Template.php

```
protected function _getCurrentUser()
{
    return $this->_auth->getCurrentUser();
}

protected function _checkIfUserHasRequiredAccessLevel($requiredAccessLevel)
{
    if ($this->_getCurrentUser()) {
        return ($this->_auth->getUserAccessLevel() >= $requiredAccessLevel);
    } else {
        return false;
    }
}
```

view/home/index.php

```
<div class="row">
    <div class="col-md-12">
        <p>Welkom bij Quistet</p>
        <?php if ($this->_checkIfUserHasRequiredAccessLevel(2)): ?>
            <p>Welkom admin!</p>
        <?php elseif ($this->_checkIfUserHasRequiredAccessLevel(1)): ?>
            <p>Welkom <?php echo $this->_getCurrentUser(); ?></p>
        <?php else: ?>
            <p>Je bent niet ingelogd: <a href="<?php echo baseUrl('home/login'); ?>">log in</a></p>
        <?php endif; ?>
    </div>
</div>
```

auth.php

- in system/model
- singleton:
  - instance in controller
  - instance in template

system/controller/Controller.php

```
class Controller
{

    ...

    protected $_auth;

    public function __construct()
    {

        ...

        // authentication
        $this->_auth = Auth::getInstance();

        ...

    }

    ...
```

HomeController.php

```
class HomeController extends Controller
{

    public function __construct()
    {
        parent::__construct();
    }

    public function login()
    {
        if ($_POST) {
            $statusMessage = $this->_auth->login($this->_input->post('email'), $this->_input->post('password'));
            if ($statusMessage->getStatus()) {
                $this->_setStatusMessage($statusMessage);
                redirect('home/onlyLoggedIn');
            } else {
                $this->_setStatusMessage($statusMessage, true);
            }
        }

        $this->_template->render('home/loginform');
    }

    public function logout()
    {
        $this->_auth->logout();
        $this->_setStatusMessage(new Message('Succesvol uitgelogd'));
        redirect('home/index');
    }
}
```

Methods afschermen

- Bijvoorbeeld:
  - /home/onlyLoggedIn alleen als ingelogd
  - /home/login alleen als niet ingelogd

HomeController.php

```
class HomeController extends Controller
{

    public function __construct()
    {
        parent::__construct();
    }

...

    public function onlyLoggedIn()
    {
        $this->_checkIfUserIsLoggedIn();

        $this->_template->render('home/private');
    }
}
```

Controle (+ logging) + herroutering

HomeController.php

```
class HomeController extends Controller
{

    public function __construct()
    {
        parent::__construct();
    }

...

    public function login()
    {
        if ($this->_auth->getCurrentUser()) {
            $this->_setStatusMessage(new Message('Je bent al ingelogd.', false));
            redirect('home/onlyLoggedIn');
        }

        ...
    }

}
```

Auth methode aanspreken, herroutering in controller

Controllers afschermen

- Bijvoorbeeld:
  - /category/* alleen voor admins

CategoryController.php

```
class CategoryController extends Controller
{
    ...

    public function __construct()
    {
        parent::__construct();

        $this->_checkIfUserHasRequiredAccessLevel(2);

        ...
    }

    ...
```

Auth methode aanspreken, doet herroutering

Admin controllers afschermen

- Bijvoorbeeld:
  - /category/* alleen voor admins
  - /question/* alleen voor admins
  - …

AdminController.php

```
abstract class AdminController extends Controller
{
    public function __construct()
    {
        parent::__construct();

        // alleen admin toelaten in deze en alle childControllers
        $this->_checkIfUserHasRequiredAccessLevel(2);

        // andere stijl zetten
        $this->_template->setStyle('admin.min.css', true);
        $this->_template->setStyle('eigenstijl.css');
    }

}
```

Auth methode aanspreken, doet herroutering

Ineens ook andere stijl zetten

CategoryController.php

Overerven van extra (abstracte) controller

```
class CategoryController extends AdminController
{
    ...

    public function __construct()
    {
        parent::__construct();

        ...
    }

...
```

Quistet log in

Log in om deze pagina te bekijken. ×

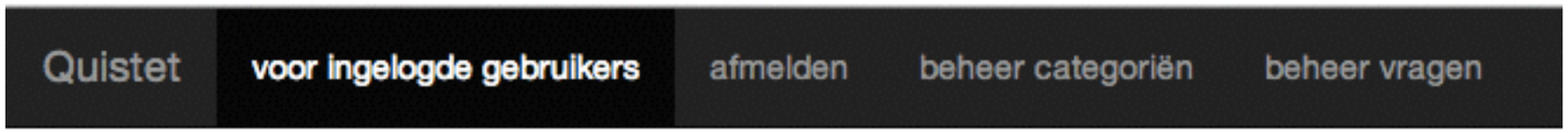
e-mail Vul je e-mail adres in

wachtwoord

login

Menu’s

- Items tonen in functie van gebruikersniveau.



Private. Alleen als je ingelogd bent.

MenuMapper.php

```
class MenuMapper
{

    public function getMenuItems($accesslevel = 0)
    {
        if ($accesslevel > 0) {

            $menuItems = array(
                new Menu('home/onlyLoggedIn', 'voor ingelogde gebruikers'),
                new Menu('home/logout', 'afmelden'),
            );
            if ($accesslevel > 1) {
                $menuItems = array_merge($menuItems,
                    array(
                        new Menu('category', 'beheer categoriën'),
                        new Menu('question', 'beheer vragen'),
                    )
                );
            }
        } else {
            $menuItems = array(
                new Menu('home/login', 'log in'),
            );
        }

        return $menuItems;
    }
}
```

Verdere verbeteringen

- Hele userobject in sessie ipv alleen id
- Verschillende menu’s maken: uitsplitsen login/ loguit, admin, rest

SVN

- https://projectwerk.webontwerp.khleuven.be
  - dynweb2013examples/theorie11/auth

Extra's

- config.php houdt rekening met lokaal development (voor als je lokale webserver hebt draaien)

- javascript van bootstrap voor nifty effectjes en jQuery gepruts (géén prioriteit!)

config.php

```
// bepalen of het lokaal of online is
if (strstr($_SERVER['HTTP_HOST'], 'khleuven.be')) {
    define('ENVIRONMENT', 'production');
} else {
    define('ENVIRONMENT', 'dev');
}

require_once('../.db_password.php');

$db_config = array(
    'driver' => 'pgsql',
    'username' => $username,
    'password' => $password,
    'schema' => 'project_u0082528', // verander dit in je eigen schema
    'dsn' => array(
        'host' => (ENVIRONMENT == 'dev') ? 'gegevensbanken.khleuven.be' : 'localhost',
        'dbname' => 'webontwerp', // verander dit in de db van je reeks
        'port' => '51314',
    )
);
```

view/_partials/ headermeta.php

```
<meta charset="utf-8">
<title><?php echo $this->getPagetitle(); ?></title>
<?php foreach ($this->_styles as $style): ?>
<link href="<?php echo baseUrl('/css/' . $style); ?>" rel="stylesheet">
<?php endforeach; ?>

<!-- JavaScript plugins (requires jQuery) -->
<script src="http://code.jquery.com/jquery.js"></script>
<!-- Include all compiled plugins (below), or include individual files as needed -->
<script src="<?php echo baseUrl('js/bootstrap.min.js'); ?>"></script>
```

Quistet

Quistet

Welkom bij Quistet

Welkom admin!

Quistet

voor ingelogde gebruikers

afmelden

beheer categoriën

beheer vragen

Quistet

Succesvol uitgelogd ×

Quistet

Welkom bij Quistet

Je bent niet ingelogd: log in

Quistet

Quistet

Welkom bij Quistet

Je bent niet ingelogd: log in

Project deadline

31 december 2013 - 23u59

1. website op:
   http://<jestudid>.webontwerp.khleuven.be/project
2. svn commit op projectwerk.khleuven.be
3. pdf met gebruikers/wachtwoorden mailen naar DL-Dynweb@khleuven.be