

Curso Técnico Superior Profissional  
Programação de Sistemas de Informação

## **Serviços e Interoperabilidade de Sistemas**

### **SERVIÇOS REST AUTENTICAÇÃO**

**Objetivos:**

- Implementar serviço REST CRUD com Yii com autenticação

**Duração: 1 aula**

©2021-2022: {bruno .madeira, mario.fernandes, nuno.costa, nuno.costa, romeu.paz}@ipleiria.pt

Cofinanciado por:



**UNIÃO EUROPEIA**  
Fundo Social Europeu

## Descrição

A presente ficha serve de guia de construir um sistema de autenticação em Yii. O serviço deve permitir a um utilizador registar-se, validar o seu registo através de email e autenticar-se utilizando as credenciais registadas, após a validação.

## Configurações Introdutórias

Utilizando o projecto da aula anterior:

1. Adicione à base de dados uma tabela chamada “utilizadores” com os seguintes campos:
  - a. IdUtilizador (Int)
  - b. Nome (Varchar)
  - c. NomeUtilizador (Varchar)
  - d. PalavraPasse (Varchar)
  - e. Email (Varchar)
  - f. Estado (int)
  - g. IdValidacao (Varchar)
2. No Gii crie o modelo e controlador:
  - a. O nome do modelo deve ser “utilizador” no singular.
  - b. No controlador o nome deve ser: «app\controllers\UtilizadoresController».Desmarque “views\...” antes de o gerar.
3. Para que não seja possível obter ou alterar os utilizadores desligue todas as ações exceto POST para o controlador.

```
public function actions()
{
    $actions = parent::actions();
    unset($actions['index'],$actions['view'],
    $actions['update'],$actions['delete']);
    return $actions;
}
```

4. Para que no momento do registo não seja necessário introduzir o **Estado** nem o **IdValidacao** por parte do utilizador e para estes serem gerados automaticamente:
  - a. No modelo altere o método “rules” para que não obrigue a introdução destes.
  - b. Acrescente ao modelo o seguinte método:

```
public function beforeSave($insert)
{
    parent::beforeSave($insert);
    if($insert)
    {
        //Definir o Estado e o IdValidacao

        $dt=Yii::$app->formatter->asTimestamp(date('Y-d-m h:i:s'));

        $dtHash = md5($dt);
        $this->IdValidacao=$dtHash;
        $this->Estado=1;
        return true;
    }
    else
    {
        return true;
    }
}
```

- c. Introduza um novo utilizador para testar que permite o registo.
5. Para que o utilizador possa validar o seu registo é necessário enviar um email ao utilizador com o link de validação:
  - a. Na linha de comandos instalar o suporte para envio de emails  
«composer require swiftmailer/swiftmailer»
  - b. Verifique se o PHP tem instalado o SSL
    - i. Crie um ficheiro /web/teste.php.
    - ii. Insira o conteúdo abaixo e aceda ao ficheiro pelo browser.

```
<?php
$xmlportlist = stream_get_transports();
print_r($xmlportlist);
?>
```

- iii. O resultado deverá ser:

```
Array ( [0] => tcp [1] => udp [2] => ssl [3] => sslv3 [4] => tls [5] => tlsv1.0
[6] => tlsv1.1 [7] => tlsv1.2 )
```

Nota: para utilizar o gmail é necessário activar app menos seguras em:

<https://myaccount.google.com/lesssecureapps>

- c. No modelo crie um método para enviar emails.

```
private function SendEmail($IdValidacao, $email)
{
    $strMsg="Clique para validar: http://localhost:8888/usuarios/validacao/".
    $IdValidacao;
    return (Yii::$app->mailer->compose()
        ->setFrom('meumail@gmail.com')
        ->setTo($email)
        ->setSubject('Validacao SIS')
        ->setTextBody($strMsg)
        ->send());
}
```

- d. Ainda no modelo adicione o seguinte método para que envie o email após guardar os dados.

```
public function afterSave($insert, $changedAttributes)
{
    parent::afterSave($insert, $changedAttributes);
    if($insert)
    {
        $this->SendEmail($this->IdValidacao, $this->Email);
    }
}
```

- e. Para configurar as credenciais de envio de email editar o ficheiro config/web.php e alterar a secção 'mailer'. Altere os campos necessários:

```
'mailer' =>
[
    'class' => 'yii\swiftmailer\Mailer',
    'viewPath' => '@app/mail',
    'useFileTransport' => false,
    'transport' => [
        'class' => 'Swift_SmtpTransport',
        'host' => 'smtp.gmail.com', //Se falhar usar ip v4...
        'username' => 'user@gmail.com',
        'password' => 'password',
        'port' => '465', //465 ou 587,
        'encryption' => 'ssl', //ssl ou tls
    ],
],
```

- f. Agora é necessário criara uma nova ação REST Get para que os utilizadores possam seguir o link enviado no email:

- i. Altere o /config/web.php na secção urlManager

```
'urlManager' => [
    'enablePrettyUrl' => true,
    'showScriptName' => false,
    'rules' => [
        ['class' => 'yii\rest\UrlRule', 'controller' => 'clientes'],
        //CUSTOM ACTIONS
        [
            'class' => 'yii\rest\UrlRule',
            'controller' => 'utilizadores',
            'tokens' => ['{idvalidacao}'=><idvalidacao:\\w+>'],
            'extraPatterns' => [ 'GET validacao/{idvalidacao}' => 'validacao'],
        ],
    ],
],
```

- ii. Adicione ao controlador o método para a nova ação.

```
public function actionValidacao($idvalidacao)
{
    $model = new $this->modelClass;
    $utilizador = $model::find()->where(['IdValidacao' => $idvalidacao])->one();
    if(is_null($utilizador))
        echo "Erro durante validacao...";
    else
    {
        $utilizador->IdValidacao=" ";
        $utilizador->Estado=2;
        if($utilizador->save())
            echo "Operação realizada com sucesso";
        else
            echo "Impossivel validar registo";
    }
}
```

- iii. Teste com <http://localhost:8888/utilizadores/validacao/«IdValidacao»>

6. Agora é necessário validar um utilizador (receber um *token*) quando se autentica:

- a. Altere o urlManager no ficheiro /config/web.php para contemplar os novos *tokens* (parâmetros):

```
'class' => 'yii\rest\UrlRule',
'controller' => 'utilizadores',
'tokens' => [
    '{idvalidacao}'=><idvalidacao:\\w+>,
    '{nomeutilizador}'=><nomeutilizador:\\w+>,
    '{palavrapasse}'=><palavrapasse:\\w+>
],
'extraPatterns' => [
    'GET validacao/{idvalidacao}' => 'validacao',
    'GET autenticacao/{nomeutilizador}/{palavrapasse}' => 'autenticacao'
],
```

- b. Implemente o método para validar o utilizador no controlador

“public function actionAutenticacao(\$nomeutilizador, \$palavrapasse)”.

Este método deve guardar o *token* gerado no campo *IdValidacao* para ser usado mais tarde.