

# LAB 8 – Email em PHP

---

**1.** Assume-se neste lab que concluiu com sucesso o LAB7. Copie todos os ficheiros da pasta “LAB7” para a pasta “LAB8”:

```
a12345@daw:~$ cd public_html
a12345@daw:~/public_html$ cp -R LAB7 LAB8
a12345@daw:~/public_html$ chmod g+w LAB8/templates_c
```

## FUNCIONALIDADE “RECOVER PASSWORD”

**2.** Desenhe o template “password\_reset\_template.tpl”

The screenshot shows a web form template for password reset. At the top, there is a blue header bar with two placeholders: {MENU\_1} on the left and {MENU\_2} {MENU\_3} on the right. Below the header is a white content area. In the center of this area, there is a red placeholder {MESSAGE}. Below the message is the title "Password reset". Underneath the title is an input field labeled "Email". At the bottom center of the form is an orange button with the text "Go". At the very bottom of the page, there is a footer with two lines of text: "© 2016 Desenvolvimento de Aplicações Web" on the left and "Designed by [Aluno](#)" on the right.

Este template utiliza o método “POST” para enviar o campo de Email ao servidor:

```
<form method="post" action="password_reset_action.php">
```

**3.** Construa o código PHP “password\_reset.php” associado a este template

No caso do email enviado não existir na base de dados, o “placeholder” {MESSAGE} deve mostrar a mensagem “Error: email does not exist”

(Nota: a passagem de informação entre password\_reset\_action.php e password\_reset.php é feita através da variável “error” da sessão)

## 4. Construa o programa `password_reset_action.php` que:

- verifica se o email introduzido consta da base de dados
- em caso de insucesso faz redirect para `password_reset.php`
- em caso de sucesso
  - Cria um token “md5” do tempo actual  
`$reset_digest = substr(md5(time()), 0, 32);`
  - guarda este valor na base de dados (tabela `users`, coluna `reset_digest`), bem como tempo actual (tabela `users`, coluna `reset_sent_at`)
  - envia ao utilizador registado um email personalizado com o assunto “Password reset” e com o texto

```
Olá Sr.(a) {$name}

Para obter uma nova password clique no link
http://all.deei.fct.ualg.pt/~a12345/LAB8/new_password.php?token={$reset_digest}
Este link tem a validade de uma hora.
Se NÃO pediu uma nova password IGNORE este email.

Cumprimentos,

webmaster!
Página web:      http://all.deei.fct.ualg.pt/~a12345/LAB8/
E-mail:          a12345@deei.fct.ualg.pt

NOTA: Não responda a este email, não vai obter resposta!
```

Note que o texto do email contem um link para a página “`new_password.php`” (substituir 12345 pelo seu numero de aluno) **com o valor do token embutido**

- coloca na variável de sessão `$_SESSION['message']=10`
- faz redirect para o URL `message.php`

## 5. Utilize o template `message_template.tpl`



{MESSAGE}

## 6. Construa<sup>1</sup> o código PHP `message.php` associado a este template

Caso a variável de sessão `$_SESSION['message']=10` sugere-se o seguinte conteúdo para o placeholder `{MESSAGE}`

```
"Password reset activated! <br> Email sent to you :-)"
```

## 7. Desenhe o template `"new_password_template.tpl"`

{MENU\_1} {MENU\_2} {MENU\_3}

New password

Password

Password confirmation

Go Clear

© 2016 Desenvolvimento de Aplicações Web Designed by [Aluno](#)

que utiliza o método “POST” para enviar a nova password ao servidor, bem como o token recebido (como um input do tipo “hidden”)

```
<form method="post" action="new_password_action.php">
<input type="hidden" name="token" value="{ $TOKEN }">
```

## 8. Construa o código PHP `new_password.php` associado a este template que actualiza o placeholder `{TOKEN}` (com o valor da variável “token” recebida embutida no link)

(Opcional: pode adicionar o placeholder `{MESSAGE}` para alertar se chegaram ao servidor duas passwords diferentes)

## 9. Crie o código PHP `new_password_action.php` que vai actualizar a base de dados com a nova password do utilizador.

- verifica se o token recebido existe na base de dados

---

<sup>1</sup> Os passos 5 e 6 apenas são necessários se não criou `message.php` e `message_template.tpl` no lab anterior

- em caso de sucesso e se não passou mais de uma hora entre a hora actual e a hora de envio do email
  - encripta e actualiza a password na base de dados
  - coloca na variável de sessão `$_SESSION['message']=11`
  - faz redirect para `message.php`

(sugere-se que o placeholder {MESSAGE} tenha o seguinte texto a amarelo: `Password reset successfully!` )

- em caso de insucesso coloca na variável de sessão `$_SESSION['message']=12` e faz redirect para `message.php`

(sugere-se que o placeholder {MESSAGE} tenha o seguinte texto a vermelho:

`ERROR: WRONG TOKEN OR TOKEN EXPIRED, PASSWORD RESET FAILED!`

Considere o lab concluído quando tiver reproduzido a funcionalidade "PASSWORD RESET" no seu portal.

**NOTA: Por razões de segurança (envio de SPAM) apenas envie emails para o dominio ualg.pt. Faça testes com um utilizador registado no portal com o seu email válido: a12345@ualg.pt.**

Caso tenha trabalhado no seu portátil/PC, faça o upload dos ficheiros para o servidor

- `password_reset.php`
- `password_reset_action.php`
- `message_template.php`
- `message.php`
- `new_password__action.php`

para a pasta "LAB8"

- `password_reset_template.tpl`
- `new_password_template.tpl`
- `message_template.tpl`

para a pasta "LAB8/templates"

## REFERÊNCIAS:

- <http://intranet.deei.fct.ualg.pt/DAW/embeddedLinks/files.html>
- <http://intranet.deei.fct.ualg.pt/DAW/auth-db-sessions/files.html>
- <http://intranet.deei.fct.ualg.pt/DAW/email.php.html>
- [http://all.deei.fct.ualg.pt/~a999990/smarty\\_exame2/](http://all.deei.fct.ualg.pt/~a999990/smarty_exame2/)

## **ANEXO 1** Acesso à base de dados MySQL

- O acesso à base de dados MySQL pode ser feita utilizando um cliente gráfico à sua escolha (por exemplo <http://www.heidisql.com/>),

ou em linha de comando

```
a12345@daw:~$mysql -u a12345 -p -h 10.10.23.183 db_a12345
```

ou ainda utilizando o software **phpMyAdmin** disponível no URL

<http://all.deei.fct.ualg.pt/phpMyAdmin>

## ANEXO 2 : estrutura da base de dados

```
CREATE TABLE `microposts` (  
  `id` int(11) NOT NULL auto_increment,  
  `content` text,  
  `user_id` int(11) default NULL,  
  `created_at` datetime NOT NULL,  
  `updated_at` datetime NOT NULL,  
  PRIMARY KEY (`id`),  
  KEY `fk_user_id` (`user_id`),  
  CONSTRAINT `fk_user_id` FOREIGN KEY (`user_id`)  
REFERENCES `users` (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `users` (  
  `id` int(11) NOT NULL auto_increment,  
  `name` varchar(255) default NULL,  
  `email` varchar(255) NOT NULL,  
  `created_at` datetime NOT NULL,  
  `updated_at` datetime NOT NULL,  
  `password_digest` varchar(255) default NULL,  
  `remember_digest` varchar(255) default NULL,  
  `admin` tinyint(1) default NULL,  
  `activation_digest` varchar(255) default NULL,  
  `activated` tinyint(1) default NULL,  
  `activated_at` datetime default NULL,  
  `reset_digest` varchar(255) default NULL,  
  `reset_sent_at` datetime default NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `index_users_on_email` (`email`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```