

Systemutveckling

PHP

Föreläsning 12 - Felhantering, mail

Dagens ämnen

- Deploy
- Felhantering
- Composer
- Mail

Deploy

- Att introducera ändring i produktionsmiljön.
- Föra över filer från en annan utvecklingsmiljö, git-repo eller liknande
- Publicera databasändringar

Deploy - ftp

- Koppla upp mot ftp-servern och för över alla filer som behövs.

Local plats: /Users/micke/www/projects/attentec/backup-live-20190305/public/

Fjärrplats: /htdocs/se

Inamn	Filstorlek	Filtyp	Senast ändrad
..			
wp-admin		Katalog	05.03.2019 13:0...
wp-content		Katalog	05.03.2019 13:4...
wp-includes		Katalog	05.03.2019 13:4...
wp-snapshots		Katalog	05.03.2019 13:5...
.ftpconfig	1 868	Fil	05.03.2019 13:0...
.htaccess	1 456	Fil	05.03.2019 13:0...
.user.ini	116	Visual Studio ...	05.03.2019 13:0...
index.php	420	Visual Studio ...	05.03.2019 13:0...
licens-sv_SE....	14 640	Visual Studio ...	05.03.2019 13:0...
license.txt	19 550	Visual Studio ...	05.03.2019 13:0...
php.ini	67 747	Visual Studio ...	05.03.2019 13:0...
php_errors.log	14 755 474	Log File	05.03.2019 13:0...
readme.html	8 962	HTML docum...	05.03.2019 13:0...
wordfence-w...	421	Visual Studio ...	05.03.2019 13:0...
wp-activate....	6 919	Visual Studio ...	05.03.2019 13:0...
wp-blog-hea...	369	Visual Studio ...	05.03.2019 13:0...
wp-comment...	2 283	Visual Studio ...	05.03.2019 13:0...
wp-config-s...	2 894	Visual Studio ...	05.03.2019 13:0...
wp-cron.php	3 847	Visual Studio ...	05.03.2019 13:0...
wp-links-op...	2 502	Visual Studio ...	05.03.2019 13:0...
wp-load.php	3 306	Visual Studio ...	05.03.2019 13:0...
wp-login.php	38 883	Visual Studio ...	05.03.2019 13:0...
wp-mail.php	8 403	Visual Studio ...	05.03.2019 13:0...

Filnamn	Filstorlek	Filtyp	Senast ändrad	Rättigheter	Ägare/Grupp
..					
.quarantine		Katalog	09.03.2019 1...	flcdmpe (...)	10005 1004
.tmb		Katalog	09.03.2019 1...	flcdmpe (...)	10005 1004
wp-admin		Katalog	09.03.2019 1...	flcdmpe (...)	10005 1004
wp-content		Katalog	11.03.2019 0...	flcdmpe (...)	10005 1004
wp-includes		Katalog	09.03.2019 1...	flcdmpe (...)	10005 1004
wp-snapshots		Katalog	09.03.2019 1...	flcdmpe (...)	10005 1004
.htaccess	401	Fil	09.03.2019 1...	adfrw (06...	10005 1004
401.html	1 970	HTML do...	09.03.2019 1...	adfrw (06...	10005 1004
403.html	1 952	HTML do...	09.03.2019 1...	adfrw (06...	10005 1004
404.html	1 944	HTML do...	09.03.2019 1...	adfrw (06...	10005 1004
500.html	1 976	HTML do...	09.03.2019 1...	adfrw (06...	10005 1004
502.html	1 956	HTML do...	09.03.2019 1...	adfrw (06...	10005 1004
503.html	1 996	HTML do...	09.03.2019 1...	adfrw (06...	10005 1004
503.jpg	7 687	jpg-fil	09.03.2019 1...	adfrw (06...	10005 1004
504.html	1 966	HTML do...	09.03.2019 1...	adfrw (06...	10005 1004
50x.html	2 017	HTML do...	09.03.2019 1...	adfrw (06...	10005 1004
index.php	420	Visual Stu...	09.03.2019 1...	adfrw (06...	10005 1004
license.txt	19 935	Visual Stu...	09.03.2019 1...	adfrw (06...	10005 1004
maintenance.html	2 098	HTML do...	09.03.2019 1...	adfrw (06...	10005 1004
readme.html	8 962	HTML do...	09.03.2019 1...	adfrw (06...	10005 1004
wp-activate.php	6 919	Visual Stu...	09.03.2019 1...	adfrw (06...	10005 1004
wp-blog-header...	369	Visual Stu...	09.03.2019 1...	adfrw (06...	10005 1004
wp-comments-p...	2 283	Visual Stu...	09.03.2019 1...	adfrw (06...	10005 1004

Deploy - git

- Vanligast är att använda ssh för att logga in på server och köra git pull från den.

Deploy - db

- Behöver ni uppdatera struktur? Data?
- Ibland kan man behöva göra skript som ändrar data, speciellt om man ändrar datastruktur.
 - Ex: om man går från att spara dokument i en tabell till att spara versioner av dokument.
- Det finns verktyg, t ex:
 - <https://www.liquibase.org/>
 - <https://flywaydb.org/>
 - <https://dbmstools.com/version-control-tools>

Deploy

- Fler saker att tänka på:
 - Uppdatera config? (Databaslösenord mm.)
 - Uppdatera filägare eller filrättigheter?

Deploy

- Hur ser era rutiner ut för att deploya kod?
- Vem deployar?
- Hur hanterar ni ev fel?
- *Uppgift:* deploya era applikationer

Felhantering

```
$file=fopen("welcome.txt","r");
```

- Om filen inte finns kommer man att få ett felmeddelande.

```
Warning: fopen(welcome.txt) [function.fopen]: failed to open stream:  
No such file or directory in C:\webfolder\test.php on line 2
```

Felhantering

- Vi kan göra egna tester för att låta användaren slippa såna felmeddelanden.

```
if(!file_exists("welcome.txt")) {  
    die("File not found");  
} else {  
    $file=fopen("welcome.txt","r");  
}
```

Egna felhanteringsfunktioner

```
error_function(error_level,error_message, error_file,error_line,error_context)
```

Parameter	Description
error_level	Required. Specifies the error report level for the user-defined error. Must be a value number. See table below for possible error report levels
error_message	Required. Specifies the error message for the user-defined error
error_file	Optional. Specifies the filename in which the error occurred
error_line	Optional. Specifies the line number in which the error occurred
error_context	Optional. Specifies an array containing every variable, and their values, in use when the error occurred

Value	Constant	Description
2	E_WARNING	Non-fatal run-time errors. Execution of the script is not halted
8	E_NOTICE	Run-time notices. The script found something that might be an error, but could also happen when running a script normally
256	E_USER_ERROR	Fatal user-generated error. This is like an E_ERROR set by the programmer using the PHP function trigger_error()
512	E_USER_WARNING	Non-fatal user-generated warning. This is like an E_WARNING set by the programmer using the PHP function trigger_error()
1024	E_USER_NOTICE	User-generated notice. This is like an E_NOTICE set by the programmer using the PHP function trigger_error()
4096	E_RECOVERABLE_ERROR	Catchable fatal error. This is like an E_ERROR but can be caught by a user defined handle (see also set_error_handler())
8191	E_ALL	All errors and warnings (E_STRICT became a part of E_ALL in PHP 5.4)

Egna felhanteringsfunktioner

```
//error handler function
function customError($errno, $errstr) {
    echo "<b>Error:</b> [$errno] $errstr";
}

//set error handler
set_error_handler("customError");

//trigger error
echo($test);
```

Error: [8] Undefined variable: test

Trigga ett fel

- Vi kan trigga ett error med funktionen `trigger_error()`.

```
$test=2;  
if ($test>=1) {  
    trigger_error("Value must be 1 or below");  
}
```

Trigga ett fel

```
//error handler function
function customError($errno, $errstr) {
    echo "<b>Error:</b> [$errno] $errstr<br>";
    echo "Ending Script";
    die();
}

//set error handler
set_error_handler("customError",E_USER_WARNING);

//trigger error
$test=2;
if ($test>=1) {
    trigger_error("Value must be 1 or below",E_USER_WARNING);
}
```

```
Error: [512] Value must be 1 or below
Ending Script
```

Övning

- Gör en felhanteringsfunktion som hanterar notiser och varningar olika, t ex skriver ut dem på olika sätt.
- Trigga fel för att testa.
- <http://php.net/manual/en/function.set-error-handler.php>

Logga fel

- Som standard skickar PHP ett felmeddelande till serverns loggsystem eller en fil, beroende på konfiguration.

Övning

- Justera er felhanteringsfunktion så att den loggar alla fel till en tabell i er databas med felmeddelande och kod.

try ... catch ... finally

- Ett exception är en slags fel som man kan välja att implementera som en del av sin felhantering.

```
Exception implements Throwable {
    /* Properties */
    protected string $message ;
    protected int $code ;
    protected string $file ;
    protected int $line ;
    /* Methods */
    public __construct ([ string $message = "" [, int $code = 0 [, Throwable $previous = NULL ]]] )
    final public getMessage ( void ) : string
    final public getPrevious ( void ) : Throwable
    final public getCode ( void ) : mixed
    final public getFile ( void ) : string
    final public getLine ( void ) : int
    final public getTrace ( void ) : array
    final public getTraceAsString ( void ) : string
    public __toString ( void ) : string
    final private __clone ( void ) : void
}
```

try ... catch ... finally

- Om något man gör kan kasta ett exception kan man hantera det genom att använda konstruktionen try catch.

```
function inverse($x) {  
    if (!$x) {  
        throw new Exception('Division by zero.');    }  
    return 1/$x;  
}  
  
try {  
    echo inverse(5) . "\n";  
    echo inverse(0) . "\n";  
} catch (Exception $e) {  
    echo 'Caught exception: ', $e->getMessage(), "\n";  
}
```

```
0.2  
Caught exception: Division by zero.
```

try ... catch ... finally

- Finally kan läggas in efter ett catch-block. Koden kommer alltid att köras oavsett om ett exception har kastats eller inte innan den normala körningen fortsätter.

```
function inverse($x) {  
    if (!$x) {  
        throw new Exception('Division by zero.');    }  
    return 1/$x;  
}  
  
try {  
    echo inverse(5) . "\n";  
} catch (Exception $e) {  
    echo 'Caught exception: ', $e->getMessage(), "\n";  
} finally {  
    echo "First finally.\n";  
}  
  
try {  
    echo inverse(0) . "\n";  
} catch (Exception $e) {  
    echo 'Caught exception: ', $e->getMessage(), "\n";  
} finally {  
    echo "Second finally.\n";  
}
```

Nästlad try ... catch

```
class MyException extends Exception { }

class Test {
    public function testing() {
        try {
            try {
                throw new MyException('foo!');
            } catch (MyException $e) {
                // rethrow it
                throw $e;
            }
        } catch (Exception $e) {
            var_dump($e->getMessage());
        }
    }
}

$foo = new Test;
$foo->testing();
```

string(4) "foo!"

Exempel

```
//create function with an exception
function checkNum($number) {
    if($number>1) {
        throw new Exception("Value must be 1 or below");
    }
    return true;
}

//trigger exception
checkNum(2);
```

Fatal error: Uncaught exception 'Exception'
with message 'Value must be 1 or below' in C:\webfolder\test.php:6
Stack trace: #0 C:\webfolder\test.php(12):
checkNum(28) #1 {main} thrown in **C:\webfolder\test.php** on line **6**

Med try ... catch

```
//create function with an exception
function checkNum($number) {
    if($number>1) {
        throw new Exception("Value must be 1 or below");
    }
    return true;
}

//trigger exception in a "try" block
try {
    checkNum(2);
    //If the exception is thrown, this text will not be shown
    echo 'If you see this, the number is 1 or below';
}

//catch exception
catch(Exception $e) {
    echo 'Message: ' . $e->getMessage();
}
```

Message: Value must be 1 or below

Egen exception-klass

```
class customException extends Exception {
    public function errorMessage() {
        //error message
        $errorMsg = 'Error on line ' . $this->getLine() . ' in ' . $this->getFile()
            . ': <b>' . $this->getMessage() . '</b> is not a valid E-Mail address';
        return $errorMsg;
    }
}

$email = "someone@example...com";

try {
    //check if
    if(filter_var($email, FILTER_VALIDATE_EMAIL) === FALSE) {
        //throw exception if email is not valid
        throw new customException($email);
    }
}

catch (customException $e) {
    //display custom message
    echo $e->errorMessage();
}
```


Flera exceptions

```
class customException extends Exception {
    public function errorMessage() {
        //error message
        $errorMsg = 'Error on line ' . $this->getLine() . ' in ' . $this->getFile()
        . ': <b>' . $this->getMessage() . '</b> is not a valid E-Mail address';
        return $errorMsg;
    }
}

$email = "someone@example.com";

try {
    //check if
    if(filter_var($email, FILTER_VALIDATE_EMAIL) === FALSE) {
        //throw exception if email is not valid
        throw new customException($email);
    }
    //check for "example" in mail address
    if(strpos($email, "example") !== FALSE) {
        throw new Exception("$email is an example e-mail");
    }
}

catch (customException $e) {
    echo $e->errorMessage();
}

catch (Exception $e) {
    echo $e->getMessage();
}
```

Kasta vidare exceptions

```
class customException extends Exception {
    public function errorMessage() {
        //error message
        $errorMsg = $this->getMessage().' is not a valid E-Mail address.';
        return $errorMsg;
    }
}

$email = "someone@example.com";

try {
    try {
        //check for "example" in mail address
        if(strpos($email, "example") !== FALSE) {
            //throw exception if email is not valid
            throw new Exception($email);
        }
    }
    catch(Exception $e) {
        //re-throw exception
        throw new customException($email);
    }
}

catch (customException $e) {
    //display custom message
    echo $e->errorMessage();
}
```

Hantera ohanterade exceptions

- Vi kan hantera ohanterade exceptions med funktionen `set_exception_handler()`.

```
function myException($exception) {  
    echo "<b>Exception:</b> " . $exception->getMessage();  
}  
  
set_exception_handler('myException');  
  
throw new Exception('Uncaught Exception occurred');
```

Exception: Uncaught Exception occurred

Övning

- Justera er felhantering till att även logga exceptions som inte hanteras någon annanstans.

Namespace

- I PHP används namespaces för att lösa två problem som kan uppstå när man utvecklar bibliotek eller applikationer i återanvändbar kod som klasser.
 - Namnkollisioner mellan kod du skapar och kod andra skapar, så som klasser/funktioner/konstanter.
 - Möjlighet att använda alias eller förkorta långa namn, något som ibland används för att lösa det första problemet.
- Namespaces ger ett sätt att gruppera relaterade klasser, interface, funktioner och konstanter.

Namespaces

Example #1 Declaring a single namespace

```
namespace MyProject;

const CONNECT_OK = 1;
class Connection { /* ... */ }
function connect() { /* ... */ }
```

Example #2 Declaring a single namespace

```
<html>
<?php
namespace MyProject; // fatal error - namespace must be the first statement in the script
?>
```

Namespaces

Example #1 Declaring a single namespace with hierarchy

```
namespace MyProject\Sub\Level;  
  
const CONNECT_OK = 1;  
class Connection { /* ... */ }  
function connect() { /* ... */ }
```

Ovanstående exempel skapar:

```
constant MyProject\Sub\Level\CONNECT_OK,  
class MyProject\Sub\Level\Connection och  
functionMyProject\Sub\Level\connect.
```

Använda namespaces

file1.php

```
<?php
namespace Foo\Bar\subnamespace;

const FOO = 1;
function foo() {}
class foo
{
    static function staticmethod() {}
}
```

file2.php

```
<?php
namespace Foo\Bar;
include 'file1.php';

const FOO = 2;
function foo() {}
class foo
{
    static function staticmethod() {}

    /* Unqualified name */
    foo(); // resolves to function Foo\Bar\foo
    foo::staticmethod(); // resolves to class Foo\Bar\foo, method staticmethod
    echo FOO; // resolves to constant Foo\Bar\FOO

    /* Qualified name */
    subnamespace\foo(); // resolves to function Foo\Bar\subnamespace\foo
    subnamespace\foo::staticmethod(); // resolves to class Foo\Bar\subnamespace\foo,
                                        // method staticmethod
    echo subnamespace\FOO; // resolves to constant Foo\Bar\subnamespace\FOO

    /* Fully qualified name */
    \Foo\Bar\foo(); // resolves to function Foo\Bar\foo
    \Foo\Bar\foo::staticmethod(); // resolves to class Foo\Bar\foo, method staticmethod
    echo \Foo\Bar\FOO; // resolves to constant Foo\Bar\FOO
}
```


Importera/använda namespaces

Importing/aliasing with the use operator

```
namespace foo;
use My\Full\Classname as Another;

// this is the same as use My\Full\NSname as NSname
use My\Full\NSname;

// importing a global class
use ArrayObject;

// importing a function (PHP 5.6+)
use function My\Full\functionName;

// aliasing a function (PHP 5.6+)
use function My\Full\functionName as func;

// importing a constant (PHP 5.6+)
use const My\Full\CONSTANT;

$obj = new namespace\Another; // instantiates object of class foo\Another
$obj = new Another; // instantiates object of class My\Full\Classname
NSname\subns\func(); // calls function My\Full\NSname\subns\func
$a = new ArrayObject(array(1)); // instantiates object of class ArrayObject
// without the "use ArrayObject" we would instantiate an object of class foo\ArrayObject
func(); // calls function My\Full\functionName
echo CONSTANT; // echoes the value of My\Full\CONSTANT
```

Composer

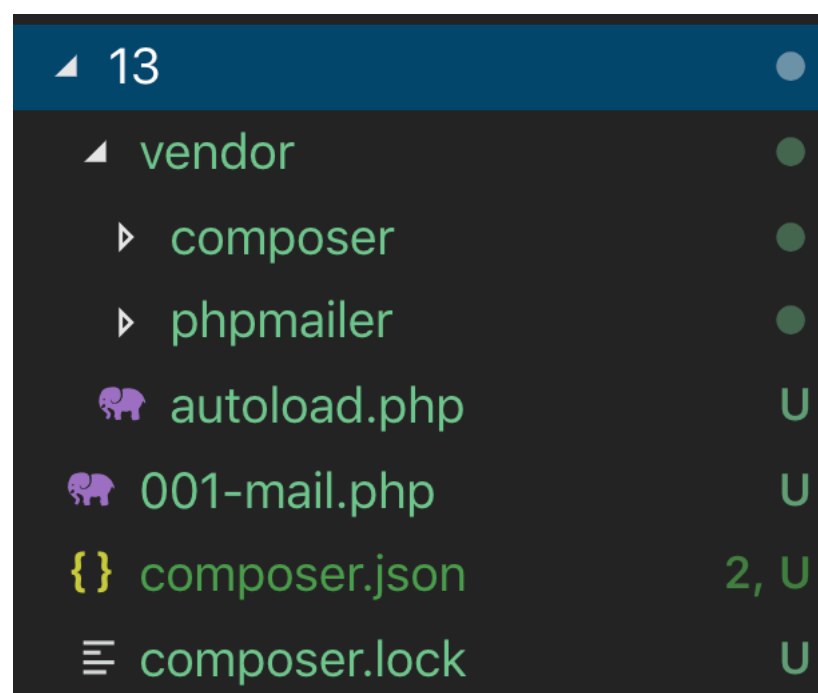
- Composer är ett verktyg som används för dependency management i PHP.
- Det låter dig deklarerera biblioteken du använder dig av i ditt projekt och hanterar installation och uppdatering av dem åt dig.
 - Installera för Windows:
<https://getcomposer.org/doc/00-intro.md#installation-windows>
 - Installera för Mac:
`brew install composer`
(Förutsatt att ni har installerat homebrew som vi använde sist:
<https://brew.sh/>)

Composer

- Navigera till mappen där ni har ert projekt.
- Installera PHPMailer:
 - `composer require phpmailer/phpmailer`

Composer

- Nu har vi ett gäng nya filer.



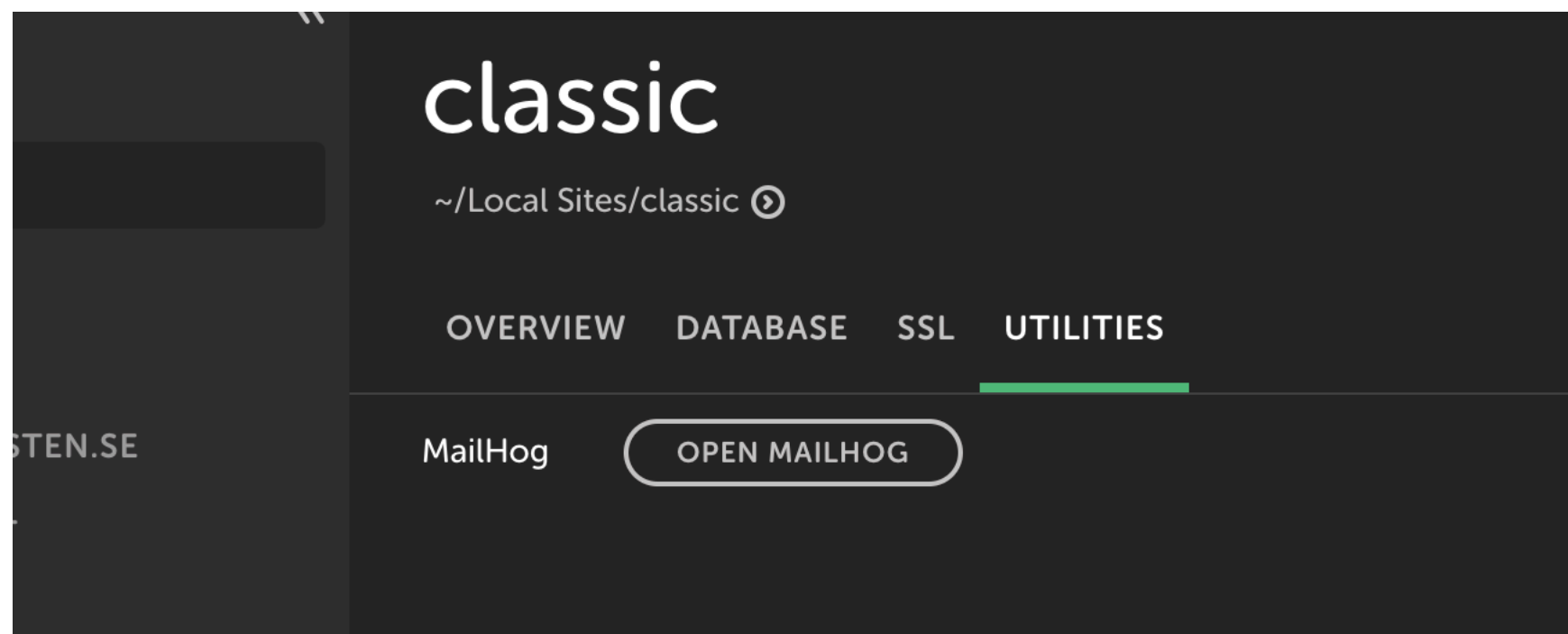
Composer

- Låt oss kolla lite närmare på `composer.json`.

```
1  {  
2      "require": {  
3          "phpmailer/phpmailer": "^6.0"  
4      }  
5  }  
6
```


Skicka mail


- Ni som har installerat *local by flywheel* har ett inbyggt system för att hantera mail.
- Mailhog kan även installeras utan local.



Mailhog


- Mailhog fångar upp mail och visar dem i webbläsaren istället för att faktiskt skicka dem.

 MailHog



Connected

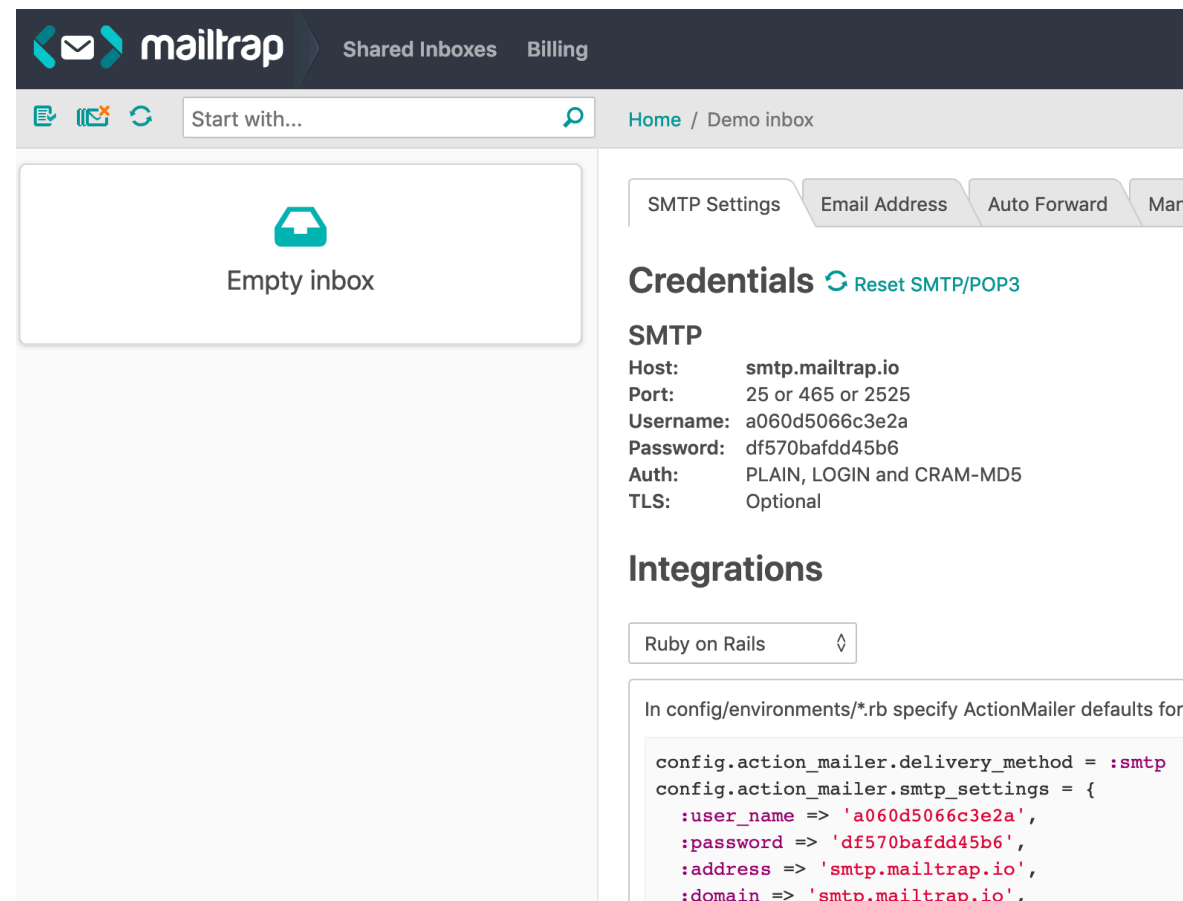
Inbox (2)

 Delete all messages

Mikael Olsson Mikael Olsson	PHPMailer SMTP test
Mikael Olsson Mikael Olsson	PHPMailer SMTP test

Skicka mail

- Det finns även system som inte behöver installeras.
- <https://mailtrap.io> - gratisalternativ finns



Testa PHPMailer

```
1  <?php
2
3  use PHPMailer\PHPMailer\PHPMailer;
4  use PHPMailer\PHPMailer\Exception;
5
6  require 'vendor/autoload.php';
7
8  $mail = new PHPMailer();
9
10 $mail->isSMTP();
11 $mail->Host = 'smtp.mailtrap.io';
12 $mail->SMTPAuth = true;
13 $mail->Username = 'a060d5066c3e2a';
14 $mail->Password = 'df570bafdd45b6';
15 $mail->SMTPSecure = 'tls';
16 $mail->Port = 25;
17
18 $mail->setFrom('mikael.olsson@emmio.se', 'Mikael Olsson');
19 $mail->addReplyTo('mikael.olsson@emmio.se', 'Mikael Olsson');
20
21 $mail->addAddress ('mikael.olsson@emmio.se', 'Mikael Olsson');
22
23 $mail->Subject = 'PHPMailer SMTP test';
24
25 $mail->AltBody = "Line 1\r\nLine 2\r\nLine 3";
26 $mail->Body = "Line 1\r\nLine 2\r\nLine 3";
27
28 if(!$mail->send()){
29     echo 'Message could not be sent.';
30     echo 'Mailer Error: ' . $mail->ErrorInfo;
31 }else{
32     echo 'Message has been sent';
33 }
```

13/001-mail.php - extern SMTP

```
1  <?php
2
3  use PHPMailer\PHPMailer\PHPMailer;
4  use PHPMailer\PHPMailer\Exception;
5
6  require 'vendor/autoload.php';
7
8  $mail = new PHPMailer();
9
10 $mail->setFrom('mikael.olsson@emmio.se', 'Mikael Olsson');
11 $mail->addAddress ('mikael.olsson@emmio.se', 'Mikael Olsson');
12
13 $mail->Subject = 'PHPMailer SMTP test';
14
15 $mail->Body = "Hello";
16
17 if(!$mail->send()){
18     echo 'Message could not be sent.';
19     echo 'Mailer Error: ' . $mail->ErrorInfo;
20 }else{
21     echo 'Message has been sent';
22 }
```

13/001-mail.php - local by flywheel

Livekodning

- Filtrering
- Sökning

Sammanfattning

- Deploy
- Felhantering
- Composer
- Mail

Utvärdering

- Prata i grupper om 2-3 personer i två minuter.
- Vad har varit bra idag?
- Vad skulle kunna förbättras?

Tack för idag!

Mikael Olsson
mikael.olsson@emmio.se
076-174 90 43

