

## Wysyłanie emaila

Wpisany przez Patryk yarpo Jar  
sobota, 26 września 2009 15:35

---

Wysyłanie emaila jest czymś co robi się dosyć często, za kazdym razem to samo, 4 zmienne [albo więcej] trzymający dane o nagłówku, o tytule, o autorze... Bleh. Czy nie lepiej jest napisać sobie raz klasę odpowiedzialną za wysyłkę e-maili, a potem tylko jej używać. Oto bardzo prosta klasa mojego autorstwa:

```
{codecitation class='brush: php'}// autor Patryk yarpo Jar 26 IX 2009, klasa pozwalająca wysyłać maile
```

```
class yMailer {  
  
    private $to      = false; // do kogo wyslac  
  
    private $subject = false; // temat wiadomosci  
  
    private $content = false; // tresc wiadomosci  
  
    public function recipient( $to ) {  
  
        $this->to = $to;  
  
    }  
}
```

## Wysyłanie emaila

Wpisany przez Patryk yarpo Jar  
sobota, 26 września 2009 15:35

---

```
public function subject( $subject ) {
```

```
    $this->subject = $subject;
```

```
}
```

```
public function content( $content ) {
```

```
    $this->content = $content;
```

```
}
```

```
public function send() {
```

```
    return mail($this->to, $this->subject, $this->content);
```

```
}
```

```
}}{/codecitation}
```

## Wysyłanie emaila

Wpisany przez Patryk yarpo Jar  
sobota, 26 września 2009 15:35

---

Oraz wykorzystanie:

```
{codecitation class='brush: php'}$mail = new yMailer();
```

```
$mail->recipient('adres(a)serwer.pl');
```

```
$mail->subject('Cześć kolego!');
```

```
$mail->content('Działa!');
```

```
echo ($mail->send()) ? &quot;Wysłało!&quot;; : &quot;Nie wysłało&quot;;{/codecitation}
```

Ktoś mógłby sobie pomyśleć - po co metody yMailer::recipient(), yMailer::subject(), yMailer::content() skoro tylko ustawiają zmienne? mogłoby być tak:

## Wysyłanie emaila

Wpisany przez Patryk yarpo Jar  
sobota, 26 września 2009 15:35

---

```
{codecitation class='brush: php'} $mail->recipient = 'adres(a)serwer.pl'; // zamien (a) na znak  
malpy
```

```
$mail->subject = 'Cześć kolego!';
```

```
$mail->content = 'Działa!';{/codecitation}
```

Jednak wtedy jesteśmy zmuszeni na zawsze w ten a nie w inny sposób używać naszego interfejsu. Bardzo ciężko wtedy jest rozwijać klasę. A ja już mam pomysł na [artykuł pokazujący jak to rozwinąć](#) ...