

KONFIGURACJA ZABEZPIECZENIA FORMULARZY PRZED SPAMEM

Przy realizacji zagadnienia przydatne są następujące metody:

- `Security::getModulesData($selected = NULL)`
- `Sett::getUnserialized($setting_key, $array_key, $exception = TRUE)`
- `Sett::serialize($setting_key, $array_key, $value)`
- `Sett::update(array $setting_data[key, value])`

Aby korzystać z klasy `Security`, należy utworzyć instancję jej obiektu:

```
1.      $_security->Security($_pdo)
```

2. Można zauważyć, że wymagane jest przesłanie instancji obiektu obsługującego bazę danych.

Zakładam, że wiesz jak utworzyć obiekt szablonu.
Jeżeli nie, to odsyłam Cię do początkowych rozdziałów tego e-booka.

Aby przesłać do szablonu dane istniejących modułów zabezpieczeń należy skorzystać z metody `Security::getModulesData()`. Przy współpracy z obiektem szablonu można to zrealizować w następujący sposób:

```
1.      $_tpl->assign('validation', $_security->getModulesData());  
2.      $_tpl->template('szablon.html');
```

W linii 2. znajduje się wywołanie pliku szablonu o nazwie `szablon.html`

W pliku szablonu wyświetlenie listy wyboru może odbyć się przez poniższy kod:

```
1.      <select name="validation">  
2.          {section=validation}  
3.              <option value="{ $validation.id }"{ $validation.name}</option>  
4.          {/section}  
5.      </select>
```

Jeżeli posiadamy w bazie zapisaną ostatnio wybraną opcję, możemy nieco zmodyfikować nasz kod do następującej postaci:

```
1.      $_tpl->assign('validation',  
2.          $_security->getModulesData(  
3.              $_sett->getUnserialized('validation', 'register')  
4.          )  
5.      );
```

Nowością jest linia 3. Jako parametr przesłano ustawienie *validation* dla rejestracji (*register*). To jednak nie wystarczy, by na liście zaznaczona była określona wartość. Musimy wprowadzić atrybut *selected*.

```
1.     <select name="validation">
2.         {section=validation}
3.         <option value="{ $validation.id}" {if
$validation.selected}selected="selected" {/if}>{ $validation.name}</option>
4.         {/section}
5.     </select>
```

Do zapisu w bazie nowej wartości opisywanego ustawienia służy metoda *Sett::update()*.

```
1.     $_sett->update(array(
2.         'validation' => $_sett->serialize('validation', 'register', $_request->post('validation')->strip())
3.     ));
```

Sett::serialize() jako 1. parametr przyjmuje *nazwę ustawienia*. Drugim jest *indeks w tablicy tego ustawienia*, gdy jest ono rozserializowane. Trzeci zaś, to wartość, jaką chcemy zapisać.

W przypadku, gdy ustawienia nie muszą być serializowane, aktualizacji można dokonać przez:

```
1.     $_sett->update(array(
2.         'validation' => $_request->post('validation')->strip()
3.     ));
```