



# CODEIGNITER SEMINERİ

KÜTÜPHANE YAZMA  
GÜVENLİK ÖNLEMLERİ  
CODEIGNITER 2.0

Özgür Web Teknolojileri Günleri 2010  
Yeditepe Üniversitesi, Airties Salonu  
16 Ekim 2010 (14:00 – 14:45)

Fatih BAZMAN (<http://codeigniter.gen.tr>)



## CI->BASLIKLAR

- CodeIgniter'ı nasıl zenginleştiririz?
- CodeIgniter'da güvenlik önlemleri
- CodeIgniter 2.0 ile neler değişecek?





# CI->NASIL\_ZENGİNLESTİRİRİZ

- Kendi yazdığımız kütüphane dosyalarını ekleyerek:
  - Örnek: Hata mesajlarını gösteren kütüphane yazılması
    - Görevi : Session kütüphanesi ile gelen hata mesajı dizisini ekrana css bilgileriyle bastırmak
    - Dosya adı: Error.php
    - Yeri : application/libraries/error.php





# CI->NASIL\_ZENGİNLESTİRİRİZ

```
<?php
if (!defined('BASEPATH')) exit('No direct script access allowed');
```

```
/**
 * Error flashing library
 * @author Fatih Bazman : fatihbazman at gmail.com
 */
```

```
class Error
{
    var $CI;

    function Error()
    {
        $this->CI =& get_instance();
        log_message('debug', "Error Class Initialized");
    }
}
```

..... Sonraki sayfada devam ediyor





# CI->NASIL\_ZENGİNLESTİRİRİZ

..... Önceki sayfadan devam ediyor

```
function error_view()
{
    if(is_array($this->CI->session->flashdata('msg')))
    {
        $text = '<blockquote>';
        foreach($this->CI->session->flashdata('msg') as $row):
            $text .= '<p>'.$row.'</p>';
        endforeach;
        $text .= '</blockquote>';
        return $text;
    }
    else if($this->CI->session->flashdata('msg'))
    {
        $text = '<blockquote><p>'.$this->CI->session->flashdata('msg').'</p></blockquote>';
        return $text;
    }
    else return FALSE;
}
```





# CI->NASIL\_ZENGİNLESTİRİRİZ

- Kütüphanenin kullanımı:
  - Yüklenmesi: application/config/autoload.php  
`$autoload['libraries'] = array( 'session','error');`
  - Çağırılması:
    - application/controller/deneme.php  
`$this->session->set_flashdata('msg', $this->form_validation->_error_array);`
    - application/views/deneme\_view.php  
`<?php echo $this->error->error_view();?>`





# CI->NASIL\_ZENGİNLESTİRİRİZ

## ○ Helper Dosyaları Yazarak:

- Yeri : application/helpers/error\_helper.php
- Yüklenmesi : application/config/autoload.php  
`$autoload['helper'] = array('error');`





# CI->NASIL\_ZENGİNLESTİRİRİZ

## ○ Helper Dosyası Yazılması:

- İçeriği:

```
function error_view()  
{  
    $CI =& get_instance();  
    return $CI->error->error_view();  
}
```

- Kullanılması

- application/views/deneme\_view.php  
<?php echo error\_view()?>







## CI->GUVENLIK\_ONLEMLERI

- Kod yazarken alınacak önlemler
  - CI güncel sürümünün takibi
  - Kullanıcı girdilerinin kontrolü
  
- Yayın öncesi alınacak önlemler
  - Varsayılan dizin yapısının ve adının değiştirilmesi
  - Kodlamaya yardımcı fonksiyonların kapatılması
  - Model, library ve diğer sonradan eklenen dosyaların başına güvenlik satırı ekleme
  - Hata mesajlarının yönetimi





# CI->GUVENLIK\_ONLEMLERI

- Kod yazarken alınacak önlemler
  - CI güncel sürümünün takibi:
    - Geçerli sürüm: 1.7.2 (upload library patched)  
<http://codeigniter.com/downloads/>





# CI->GUVENLIK\_ONLEMLERI

- Kod yazarken alınacak önlemler
  - Kullanıcı girdilerinin kontrolü:
    - \$\_GET, \$\_POST, \$\_COOKIE, \$\_SERVER fonksiyonları yerine CI Input sınıfının kullanılması.  
(int) \$this->input->get('page',TRUE);
  - Xss önlemek amacıyla form girdilerinde kontrol.  
\$this->form\_validation->set\_rules('key', 'Arama kriteri',  
'trim|required|xss\_clean|min\_length[3]');  
\$this->input->get('page',TRUE)





# CI->GUVENLIK\_ONLEMLERI

## ○ Kod yazarken alınacak önlemler

- Kullanıcı girdilerinin kontrolü:

- SQL injection önlemek amacıyla:

- Active Record kullanımı

- ```
$this->db->get_where('posts',array('user_id' => $user_id));
```

- Bind query kullanımı:

- ```
$sql = 'select * from tabloadi where sutun1=? and sutun2=?';
```

- ```
$this->db->query($sql, array($degisken1,$degisken2));
```

- Doğrudan Query fonksiyonu kullanımında girdinin temizliği:

- ```
mysql_real_escape_string($this->input->get('key',TRUE));
```

- ya da,

- ```
"INSERT INTO table (title) VALUES(".$this->db->escape($key).")";
```





# CI->GUVENLIK\_ONLEMLERI

## ○ Yayın öncesi alınacak önlemler

- Varsayılan dizin yapısının ve adının değiştirilmesi :
  - root
    - system
      - application
  - root
    - CI(eski system dizini)
    - uygulama (eski system/application dizini)

*Root/index.php dosyası içindeki iki satır:*

```
$system_folder = "CI";// system_dizini_adı
```

```
$application_folder = "uygulama"; //application_dizini_adı
```

- Çekirdeği değiştirdiğimiz dosyaların MY\_ ön eklerinin değiştirilmesi:

Application/config/config.php'de `$config['subclass_prefix'] = 'CI_';`





# CI->GUVENLIK\_ONLEMLERI

- Yayın öncesi alınacak önlemler
  - Kodlamaya yardımcı fonksiyonların kapatılması
    - Scaffolding kütüphanesi kullanıldıysa devre dışı bırakılmalı  
~~`$this->load->scaffolding('table_name');`~~
    - Profiler kullanıldıysa kapatılmalı  
~~`$this->output->enable_profiler(TRUE);`~~
  - Model, library ve diğer sonradan eklenen dosyaların başına güvenlik satırı ekleme:  
`<?php if (!defined('BASEPATH')) exit('No direct script access allowed');`





# CI->GUVENLIK\_ONLEMLERI

- Yayın öncesi alınacak önlemler
  - Hata mesajlarının yönetimi
    - Tüm hata mesajlarının ekrana bastırılması önlenmelidir:
      - Root/index.php :  
error\_reporting(0);
      - application/config/config.php:  
\$config['log\_threshold'] = 1;  
(system/logs dizinine yazma yetkisinin verilmesi gereklidir.)
      - application/config/database.php:  
\$db['default']['db\_debug'] = FALSE;





## CI->VERSIYON\_2.0

- Versiyon 2.0 ana değişiklikleri
  - PHP4 desteği kalkıyor:
    - PHP5'in tüm kullanışlı opsiyonları geliyor ( \_\_construct, \_\_destruct, \_\_autoload vs)
  - Bazı kütüphaneler kullanımdan kalkıyor:
    - Scaffolding ve pluginler kaldırılıyor  
(CAPTCHA plugin'i helper dosyasına dönüşüyor).
    - Validation sınıfı tamamen kalkıyor, form\_validation sınıfına destek genişleyerek sürüyor.







## CI->VERSIYON\_2.0

- Versiyon 2.0 ana değişiklikleri
  - Çekirdek sınıfların değiştirildiği dosyaların yeri değişiyor:
    - MY\_ ön ekli dokümanları application/core dizini altına kopyalanacak.
  - Application paketleri eklenebiliyor:
    - application/\_common dizini altına farklı uygulamalarda ortak kullanılan dosyalar (model dosyaları gibi) kopyalanabilir.





## CI->VERSIYON\_2.0

- Versiyon 2.0 ana değişiklikleri
  - Driver sınıfları yazma opsiyonu geliyor:

```
$this->load->driver('api');
```

```
$this->api->twitter->call('statuses/update', array('update' => 'Nice status!'));
```

```
$this->api->facebook->get_friends();
```

```
$this->api->vimeo->get_videos($user);
```

*Jquery için yazılmış driver dosyası , CI 2.0 standart dağıtım paketine dahil edilmiştir.*





## CI->VERSIYON\_2.0

- Versiyon 2.0 ana değişiklikleri
  - Veritabanına daha fazla kontrol geliyor:

```
$db['default']['swap_pre'] = '';
```

```
$db['default']['autoinit'] = TRUE;
```

```
$db['default']['stricton'] = FALSE;
```

```
$db['default']['port'] = 5432;
```





## CI->VERSION\_2.0

- Versiyon 2.0 ana değişiklikleri
  - CSRF kontrolünün de eklendiği yeni Security sınıfı geliyor:
    - Input, Upload ve XML-RPC sınıfları Security sınıfı ile daha güçlendiriliyor.
    - XSS filtresinin yeri Security sınıfına kaydırılıyor.





## CI->SONUC

- Kullanım kolaylığı, hızı, anlaşılır kullanım kılavuzu ve PHP5 destekli yeni versiyonuyla yola devam eden CodeIgniter, PHP uygulama çatıları arasındaki yerini gelecekte de güçlendirmeyi istiyor.



# CODEIGNITER SEMINERİ



Teşekkürler.

Fatih BAZMAN  
[codeigniter.gen.tr](http://codeigniter.gen.tr)

