



ICMS 2010 Innovation

# #01 CI세미나 - WEBMAKER3 적용기술

작성 및 발표 : 변종원 (blumine@gmail.com)

CodeIgniter 한국사용자포럼 운영자 (웅파)

NOTICE: Proprietary and Confidential

This material is proprietary to ICMS. It contains trade secret and confidential information which is solely the property of ICMS. This material is for client's internal use only. It shall not be used, reproduced, copied, disclosed, transmitted, in whole or in part, without the express consent of ICMS

© 2010 ICMS INC. All rights reserved.



# 목차

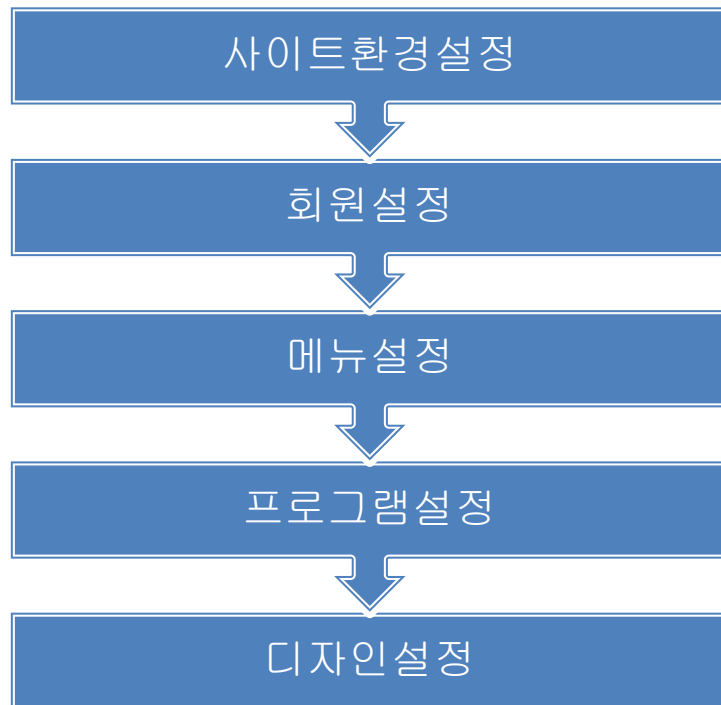
1. 인사말
2. 웹메이커3의 전체구조 및 CI 적용기술 (웹메이커3 시연)
  - 웹메이커3 구조 설명
  - URL (기존 주소와 쿼리 스트링 혼용)
  - MATCHBOX 개념 및 실제 작동 예
  - WIDGET 개념 및 실제 작동 예



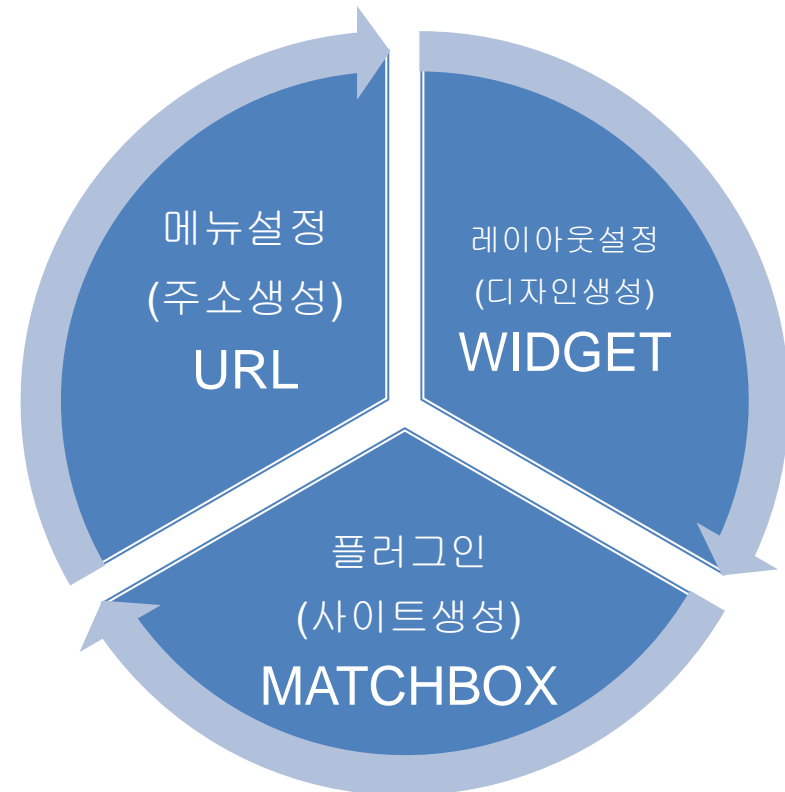
# 인사말

## 웹메이커3 구조

### 운영자 메뉴 구조

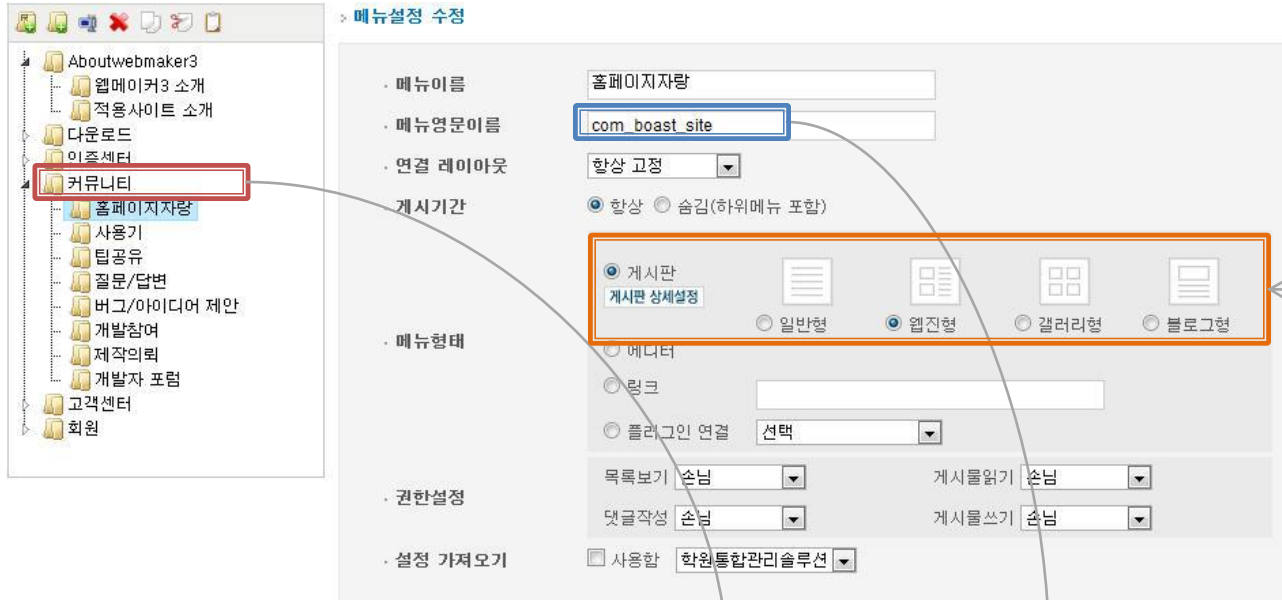


### 핵심 동작 구조



## URL (기존 주소와 쿼리 스트링 혼용) 1

1. 웹메이커3에서 사용할 주소체계를 만들어 낸다.



[http://www.hbuilder.com/community/com\\_boast\\_site](http://www.hbuilder.com/community/com_boast_site)

- 위 화면의 셋팅 결과 주소가 만들어지고 /community/com\_boast\_site 에 웹진형 게시판이 연결
- 연결된 기본 메소드는 list라서 위 주소만으로 게시판의 리스트가 실행

문제! Community라는 디렉토리도 없고 컨트롤러도 없다. 어떻게 주소와 연결됐을까?

## URL (기존 주소와 쿼리 스트링 혼용) 2

### 2. 주소를 만들어 내고 보니 문제가 생겼다.

- 주소구조가 무한 카테고리 구조이기 때문에 /1단/2단/3단/4단... 형식으로 무한 생성이 가능하다.
- 그러다 보니 검색 시, 검색 후 페이징 처리에서 문제가 생겼고, 변수처리에 문제가 생겼다.
- segment 주소방식의 장점을 버릴 수 없어서 기존주소와 쿼리스트링을 혼용

어디가 카테고리이고  
어디가 변수냐? T.T

http://www.hbuilder.com/community/com\_boast\_site/method/view/no/1/page/1

http://www.hbuilder.com/community/com\_boast\_site/?method=view&no=1&page=1

```
$this->seg_exp = $this->common->segment_explode($this->uri->uri_string());
```

```
Array
(
    [0] => community
    [1] => com_boast_site
    [query_string] => Array
        (
            [method] => view
            [no] => 1
            [page] => 1
        )
)
```

기존 주소는 카테고리들  
의미 (무한)

쿼리스트링은 카테고리 이  
외의 모든 변수

## MATCHBOX의 개념 및 실제 작동 예 1

### 1. Application 디렉토리의 한계를 뛰어넘자

- 웹메이커3는 사이트를 레이아웃과 플러그인으로 구성한다.
- 이 조건을 만족시키기 위해서는 application 디렉토리만으로는 구현할 수 없다



**MATCHBOX 적용**

- hmvc 라는 동일한 역할을 하는 것도 있지만 도입시기에는 버그가 상당히 있었다. (현재는 개선됨)
- 그 외 index.php 파일을 다른 이름으로 2개 만들고 application 디렉토리도 2개 만들어 사용하는 방법도 있었다.
- 그 외 여러 가지 방법 존재 (codeigniter.com 검색해보면 개발자들의 needs가 동일한 것을 알 수 있다)



2009-03-18-codeigniter\_1\_7\_1\_matchbox\_0\_9\_4.zip

- MATCHBOX는 원래 codeigniter 1.5에 맞춰서 개발된 것이었다. 누군가 1.7.1 버전에서 사용할 수 있게 수정 후 배포
- 1.7.2 버전에서도 이상 없이 돌아갈 것으로 판단됨
- 아래 파일만 1.7.2 버전 설치된 곳에 복사 후 설정 수정

applicaion/config/matchbox.php

application/libraries/Loader.php, Matchbox.php, My\_Config.php, My\_Language.php, Router.php

## 플러그인

- 📁 plugins 플러그인 디렉터리 (application과 같은 레벨)
- 📁 .svn
- 📁 banner 플러그인(이름으로 구분)
- 📁 board 플러그인
  - 📁 .svn
  - 📁 controllers 컨트롤러 (MVC 에서 C)
  - 📁 libraries 사용자 library나 core 확장 library
  - 📁 models 모델 위치 (MVC 에서 M)
  - 📁 views 뷰 위치 (MVC 에서 V)

## 웹메이커3 Core 위치함

- 📁 application
  - 📁 config 각종 설정
  - 📁 controllers 컨트롤러 위치 (MVC 에서 C)
  - 📁 errors
  - 📁 helpers 헬퍼함수
  - 📁 hooks 후킹 프로그램 위치
  - 📁 language 예러메시지(언어별 셋팅)
  - 📁 libraries 사용자 library나 core 확장 library
  - 📁 models 모델 위치 (MVC 에서 M)
  - 📁 views 뷰 위치 (MVC 에서 V)

```
<?php
class Board extends Plugin {

    function __construct()
    {
        $this->CI = & get_instance();
        $this->CI->load->module_model('board', 'board_model'); //플러그인의 모델 사용
        $this->CI->load->model('admin_menu_m', 'menu'); // 웹메이커3의 모델 사용
        $this->CI->load->helper('ckeditor');
    }
    function run($data = array())
    {
        $data['module_type'] = "plugin";
        $data['module_directory'] = "boards";
        $data['module_name_en'] = "board";
        $data['module_info'] = $this->CI->admin_program_m->module_info($data);
        $data['detail_xml'] = unserialize($data['module_info']['detail_xml']);
        $data['skin'] = $data['detail_xml']['default_skin'];
        $data['skin_dir'] = "plugins/" . $data['module_directory'] . "/views/" . $data['skin'];
        $data['CI'] = $this->CI;
        return $this-> $data['function'] ($data);
    }

    function lists($data = array())
    {
        //리스트처리
        //뷰파일 선언
        $data['view'] = "joinstart_v";
        return $this->render($data);
    }

}
?>
```

```
<?php
class Board extends Controller {

    function Board()
    {
        //필요없음
        $this->load->module_model('board', 'board_model');
        $this->load->model('admin_program_m', 'program');
        $this->load->helper('ckeditor');
    }
    function index()
    {
        $data['mfo'] = $this->program->module_info($data);
        $this->load->view('view_file', $data);
    }

}
?>
```

MATCHBOX와 WIDGET이 따로 적용된 것이 아니어서  
원래 사용구조와는 조금 틀리다. (MATCHBOX와 WIDGET의 결합)



# WIDGET의 개념 및 실제 작동 예 1

## 1. 컨트롤러 안에서 다른 컨트롤러를 호출하고 싶다.... (나만 그런 게 아냐.. ^^;;)

- 레이아웃의 각 해당 위치에 플러그인을 배치하여 사이트를 보여주는 구조이다 보니 각 플러그인을 레이아웃의 <div></div> 안에 넣어야 한다.
- 플러그인은 기존 CI와 동일한 구조라 컨트롤러를 호출하여 사용한다.
- 그러기 위해서는 컨트롤러 안에서 다른 컨트롤러를 호출하여 사용할 수 있어야 한다.

The screenshot shows the ICMS WebMaker 3.0 interface. The main banner area features the text '웹메이커 3.0' and 'Version Up'. Below the banner, there are several sections: '공지사항' (Notice), 'PORTFOLIO', and a list of services. Callouts point to specific widgets:

- 로고 플러그인**: Points to the ICMS logo in the top left.
- 바로가기 플러그인**: Points to the 'HOME', 'SITEMAP', and 'CONTACT US' links in the top right.
- 메뉴 플러그인**: Points to the navigation menu in the top right.
- 상단이미지 플러그인**: Points to the main banner area.
- 메인 공지사항 플러그인**: Points to the '공지사항' (Notice) section on the left.
- 배너 플러그인**: Points to the 'PORTFOLIO' section in the middle.
- HTML입력 플러그인**: Points to the list of services on the right.
- 카피라이트 플러그인**: Points to the footer area at the bottom.

## WIDGET의 개념 및 실제 작동 예 2

### 2. 실제 작동 예

- 실제 컨트롤러에서 위젯 호출 시 view의 순서 문제가 생긴다. \$this->load->view()가 항상 제일 마지막에 실행되기 때문에..
- 아래와 같이 ob\_start(), ob\_get\_contents(), ob\_end\_clean() 로 처리를 하면 제 순서대로 가능

```
class Plugin
{

    function Plugin() {
        $this->_assign_libraries();
    }

    function run($data = array()){
        require_once PLUGIN_ROOT.'/'.$data['module_directory'].'/controllers/'.$data['controller'].EXT;
        $module =& new $data['controller']();
        return call_user_func_array(array(&$module, 'run'), array($data));
    }

    function render($data = array()) {
        extract($data); //$data['name'] => $name 형으로 변경
        ob_start();
        include PLUGIN_ROOT.'/'.$data['module_directory'].'/views/'.$data['skin'].'/'.$data['view'].EXT;
        $return = ob_get_contents();
        ob_end_clean();
        return $return;
    }

    function load($Object) {
        $this->$Object =& load_class(ucfirst($Object));
    }

    function _assign_libraries() {
        $ci =& get_instance();
        foreach (get_object_vars($ci) as $key => $Object) {
            $this->$key =& $ci->$key;
        }
    }
}
```