

```

<?php
/**
 * Maple - PHP Web Application Framework
 * Copyright (C) 2004 TAKAHASHI Kunihiko
 */

require_once 'core/Smarty4Maple.class.php';

/**
 * Viewの実行準備および実行を行うFilter
 *
 * @author TAKAHASHI Kunihiko
 * @package maple.filter
 * @since 3.0.0
 */
class Filter_View extends Filter {
    /**
     * コンストラクター
     *
     * @access public
     * @since 3.0.0
     */
    function Filter_View() {
        parent::Filter();
    }

    /**
     * Viewの処理を実行
     *
     * @access public
     * @since 3.0.0
     */
    function execute() {
        $log =& LogFactory::getLog();
        $log >trace("Filter_Viewの前処理が実行されました", "Filter_View#execute"
);

        $container =& DIContainerFactory::getContainer();
        $filterChain =& $container >getComponent("FilterChain");
        $filterChain >execute();

        $response =& $container >getComponent("Response");
        $view = $response >getView();

        if ($view != "") {
            // entry point から切替部分の取り出し
            preg_match("/\#[\w]*-([\w]*)\.php$/", $_SERVER['SCRIPT_NAME'], $match);

            $template = $this >getAttribute($view);
            // 切替テンプレートが無かった時のために、保持
            $old_template = $template;

            if (isset($match[1])) {
                // テンプレート名の決定
                preg_match("/(.*?)\#(.*?)/", $template, $html);
                $template = $html[1]."-".$match[1].".$html[2];

                // 切替テンプレートの存在確認。無い時は、以前のものに戻す
                if (!file_exists(SMARTY_TEMPLATE_DIR . $template)) {
                    $template = $old_template;
                }
            }

            if ($template == "") {
                $log >error("テンプレートファイルの取得に失敗しました", "Filter_
View#execute");
                exit;
            }

            if (preg_match("/location:/", $template)) {
                $url = preg_replace("/location:/", "", $template);
                $url = trim($url);
                $response >setRedirect($url);
            } else {
                $renderer =& Smarty4Maple::getInstance();

                $actionChain =& $container >getComponent("ActionChain");
                $action =& $actionChain >getCurAction();
            }
        }
    }
}

```

```

    $renderer >setAction($action);

    $errorList =& $actionChain >getCurErrorList();
    $renderer >setErrorList($errorList);

    $token =& $container >getComponent("Token");
    if (is_object($token)) {
        $renderer >setToken($token);
    }

    $result = $renderer >fetch($template);

    if ($result != "") {
        $response >setResult($result);
    }
}

$contentDisposition = $response >getContentDisposition();
$contentType         = $response >getContentType();
$result              = $response >getResult();
$redirect            = $response >getRedirect();

if ($redirect) {
    header("Location: ${redirect}");
} else {
    ob_start();
    if ($contentDisposition != "") {
        header("Content-disposition: ${contentDisposition}");
    }
    if ($contentType != "") {
        header("Content-type: ${contentType}");
    }
    //header("Content-Length: ".strlen($result));
    print $result;
    while (@ob_end_flush());
}

$log >trace("Filter_Viewの後処理が実行されました", "Filter_View#execute"
);
}
?>

```