

```

<?php
/*
 * Purpose: Dynamic Output of Article List (Main Page always not in list)
 * Note: Currently support four types (new, hot, update, discussion).
 * new => Newly Posted Articles
 * update => Recently Updated Articles
 * hot => Most Popular Articles
 * discussion => Recently Updated Discussion
 * @author: Zeng Ji (zengji@gmail.com)
 *
 * To install, add following to LocalSettings.php
 * require_once ("extensions/DynamicArticleList.php");
 */

$wgExtensionFunctions[] = "wfDynamicArticleList";

function wfDynamicArticleList() {
    global $wgParser;

    $wgParser->setHook( "DynamicArticleList", "DynamicArticleList" );
}

// The callback function for converting the input text to HTML output
function DynamicArticleList( $input ) {
    require_once ('CategoryUtil.php');

    $dbr =& wfGetDB( DB_SLAVE );

    // INVALIDATE CACHE
    global $wgTitle;
    $wgTitle->invalidateCache();

    // Default Values
    $listTitle = false;
    $listType = 'new';
    $listCount = 5;
    $categoryRoot = false;

// ##### PARSE PARAMETERS #####
    $aParams = explode("\n", $input);
    foreach($aParams as $sParam) {
        $aParam = explode("=", $sParam);
        if( count( $aParam ) < 2 )
            continue;
        $sType = trim($aParam[0]);
        $sArg = trim($aParam[1]);

        switch ($sType) {
            case 'title':
                $listTitle = $sArg;

```

```

        break;

    case 'type':
        if ( in_array($sArg, array('new','hot','update', 'discussion')) )
            $listType = $sArg;
        break;

    case 'count':
        $listCount = IntVal( $sArg );
        break;

    case 'categoryRoot':
        $categoryRoot = $sArg;
        break;
    }
}

// ##### CHECKS ON PARAMETERS #####
if ($listTitle!=false && strlen($listTitle)==0)
    $listTitle=false;

// ##### BUILD SQL QUERY #####
$sql = genSQL($listType, $listCount, $categoryRoot);

// ##### PROCESS SQL QUERY #####
global $wgUser;
global $wgLang;
global $wgContLang;

$res = $dbr->query($sql);
$sk =& $wgUser->getSkin();
if ($dbr->numRows( $res ) != 0) {
    while( $row = $dbr->fetchObject ( $res ) ) {
        $title = Title::makeTitle( $row->namespace, $row->title);
        if ($listType == 'discussion')
            $sLink = $sk->makeKnownLinkObj($title, $wgContLang->convertHtml($title->getText()), ", ", 'Discussion: ');
        else
            $sLink = $sk->makeKnownLinkObj($title, $wgContLang->convertHtml($title->getText()));

        // Generate 'View Count' String (for 'hot')
        if ($listType == 'hot')
            $aViews[] = ' - ( ' . $row->count . ' views ) ';
        else
            $aViews = false;

        // Generate 'Time' String (for 'new', 'update', 'discussion')
        if ($listType != 'hot') {
            $aDates[] = ' - [ ' . $wgLang->timeanddate($row->timestamp, true) . ' ] ';
        }
    }
}

```

```

        $editorID = $row->user;
        if ($editorID == 0) {
            $editor = wfMsg('anonymous');
            $aEditors[] = ' ( ' . $editor . ' )';
        } else {
            $editor = User::whoIs($editorID);
            $aEditors[] = ' ( ' . $sk->makeLink($wgContLang->getNsText(NS_USER) . ':' . $editor,
htmlspecialchars($editor)) . ' )';
        }
    } else {
        $aDates = false;
        $aEditors = false;
    }
}

$aArticles[] = $sLink;
}
}
$dbr->freeResult( $res );

// ##### GENERATE OUTPUT #####
return OutputList($aArticles, $aEditors, $aDates, $aViews, $listTitle);
}

function genSQL( $type, $count, $categoryRoot=false ) {
    $dbr =& wfGetDB( DB_SLAVE );
    $sPageTable = $dbr->tableName( 'page' );
    $sRevisionTable = $dbr->tableName( 'revision' );
    $sRecentChangesTable = $dbr->tableName( 'recentchanges' );
    $sCategoryLinksTable = $dbr->tableName( 'categorylinks' );

    $categoryUtil = new CategoryUtil();

    $sql = "";
    if ($type == 'hot') {

        if ($categoryRoot != false) {
            $cNameList = $categoryUtil->getCNameList($categoryRoot);
            $sql = "
                SELECT DISTINCT
                page_title as title,
                page_namespace AS namespace,
                page_counter as count
                FROM $sPageTable
                INNER JOIN $sCategoryLinksTable ON page_id=cl_from
                WHERE page_namespace=".NS_MAIN." AND page_is_redirect=0 AND
page_id!=1 AND cl_to IN ".$cNameList."
                ORDER by count DESC
                LIMIT $count
            ";
        }
    }
}

```

```

    } else {
        $sql = "
            SELECT DISTINCT
            page_title as title,
            page_namespace AS namespace,
            page_counter as count
            FROM $sPageTable
            WHERE page_namespace=".NS_MAIN." AND page_is_redirect=0 AND
page_id!=1
            ORDER by count DESC
            LIMIT $count
        ";
    }

} elseif ($type == 'update') {

// Step 1: Get revision list order by rev_id
if ($categoryRoot != false) {
    $cNameList = $categoryUtil->getCNameList($categoryRoot);
    $sql = "
        SELECT
        page_id,
        MAX(rev_id) AS max_rev_id
        FROM $sPageTable
        INNER JOIN $sRevisionTable ON page_id=rev_page
        INNER JOIN $sCategoryLinksTable ON page_id=cl_from
        WHERE page_is_new!=1 AND page_namespace=0 AND page_is_redirect=0
AND page_id!=1 AND cl_to IN ".$cNameList."
        GROUP BY page_id
        ORDER by max_rev_id DESC
        LIMIT $count
    ";
} else {
    $sql = "
        SELECT
        page_id,
        MAX(rev_id) AS max_rev_id
        FROM $sPageTable
        INNER JOIN $sRevisionTable ON page_id=rev_page

        WHERE page_is_new!=1 AND page_namespace=0 AND page_is_redirect=0
AND page_id!=1

        GROUP BY page_id
        ORDER by max_rev_id DESC
        LIMIT $count
    ";
}

// Step 2: According to revision list, generate SQL to retrieve article page information.
$res = $dbr->query($sql);

```

```

$inClause = "";
if ($dbr->numRows( $res ) == 0) {
    $inClause = "-1";
} else {
    while( $obj = $dbr->fetchObject( $res ) ) {
        if( isset( $obj->max_rev_id ) ) {
            $inClause .= $obj->max_rev_id . ',';
        }
    }
    $inClause = substr($inClause, 0, strlen($inClause)-1);    //delete trailing ','
}
$dbr->freeResult( $res );

```

```

$sql = "
    SELECT
        page_title AS title,
        page_namespace AS namespace,
        rev_user AS user,
        rev_timestamp AS timestamp
    FROM $sRevisionTable, $sPageTable
    WHERE rev_page=page_id AND rev_id IN (". $inClause . ")
    ORDER BY rev_id DESC";

```

```

} elseif ($type == 'discussion') {

```

```

    // Step 1: Get revision list order by rev_id.

```

```

    if ($categoryRoot != false) {
        $cNameList = $categoryUtil->getCNameList($categoryRoot);
        $sql = "

```

```

            SELECT
                p1.page_id,
                MAX(rev_id) AS max_rev_id
            FROM $sPageTable AS p1
            INNER JOIN $sPageTable AS p2 ON p1.page_title=p2.page_title
            INNER JOIN $sRevisionTable ON p1.page_id=rev_page
            INNER JOIN $sCategoryLinksTable ON p2.page_id=cl_from
            WHERE p1.page_is_redirect=0 AND p1.page_namespace=1 AND
p2.page_namespace=0 AND cl_to IN ".$cNameList."
            GROUP BY rev_page
            ORDER by max_rev_id DESC
            LIMIT $count

```

```

";

```

```

    } else {
        $sql = "
            SELECT
                p1.page_id,
                MAX(rev_id) AS max_rev_id
            FROM $sPageTable AS p1
            INNER JOIN $sRevisionTable ON p1.page_id=rev_page

```

```

        WHERE p1.page_is_redirect=0 AND p1.page_namespace=1
        GROUP BY rev_page
        ORDER by max_rev_id DESC
        LIMIT $count
    ";
}

// Step 2: According to revision list, generate SQL to retrieve discussion page information.
$res = $dbr->query($sql);
$inClause = "";
if ($dbr->numRows( $res ) == 0) {
    $inClause = "-1";
} else {
    while( $obj = $dbr->fetchObject( $res ) ) {
        if( isset( $obj->max_rev_id ) ) {
            $inClause .= $obj->max_rev_id . ',';
        }
    }
    $inClause = substr($inClause, 0, strlen($inClause)-1); //delete tailing ','
}
$dbr->freeResult( $res );

$sql = "
    SELECT
        page_title AS title,
        page_namespace AS namespace,
        rev_user AS user,
        rev_timestamp AS timestamp
    FROM $sRevisionTable, $sPageTable
    WHERE rev_page=page_id AND rev_id IN (". $inClause . ")
    ORDER BY rev_id DESC";
} else { // default type is 'new'

if ($categoryRoot != false) {
    $cNameList = $categoryUtil->getCNameList($categoryRoot);
    $sql = "
        SELECT DISTINCT
            page_title AS title,
            page_namespace AS namespace,
            rc_user AS user,
            rc_timestamp AS timestamp
        FROM $sPageTable
        INNER JOIN $sRecentChangesTable ON page_id=rc_cur_id
        INNER JOIN $sCategoryLinksTable ON page_id=cl_from
        WHERE rc_new=1 AND rc_namespace=0 AND page_is_redirect=0 AND
page_id!=1 AND cl_to IN ".$cNameList."
        ORDER by rc_id DESC
        LIMIT $count
    ";
}
}

```

```

    } else {
        $sql = "
            SELECT DISTINCT
                page_title AS title,
                page_namespace AS namespace,
                rc_user AS user,
                rc_timestamp AS timestamp
            FROM $sPageTable
            INNER JOIN $sRecentChangesTable ON page_id=rc_cur_id
            WHERE rc_new=1 AND rc_namespace=0 AND page_is_redirect=0 AND
page_id!=1

            ORDER by rc_id DESC
            LIMIT $count
        ";
    }
}

return $sql;
}

function OutputList ( $aArticles, $aEditors, $aDates, $aViews, $listTitle ) {
    if ($listTitle != false) {
        $r .= " <h3>" . $listTitle . "</h3>\n";
        $r .= " <hr/>\n";
    }

    $sStartList = '<ul>';
    $sEndList = '</ul>';
    $sStartItem = '<li>';
    $sEndItem = '</li>';

    $r .= $sStartList . "\n";
    for ($i=0; $i<count($aArticles); $i++) {
        $editorString = "";
        $dateString = "";
        $viewString = "";

        if ($aEditors != false)
            $editorString = "<font size=-2>" . $aEditors[$i] . "</font>";
        if ($aDates != false)
            $dateString = "<font size=-2>" . $aDates[$i] . "</font>";
        if ($aViews != false)
            $viewString = "<font size=-2>" . $aViews[$i] . "</font>";

        $r .= $sStartItem . $aArticles[$i] . $editorString . $dateString . $viewString . $sEndItem . "\n";
    }
    $r .= $sEndList . "\n";

    return $r;
}

```

?>