

Wordpress plugin

- 1.
- 2.
- 3.
- 4.

1.

- [synapeditor_plugin.zip](#)

2.

1. .
2. .
 - synapeditor.min.js ()
 - synapeditor.min.css ()
 - license.json ()
 - synapeditor.config.js ()
 - sedocConverter ()
3. (synapeditor.min.js, synapeditor.min.css, synapeditor.config.js) resource .
4. (sedocConverter) resource/sedocConverter .

3.

1. (license.json) resource .
2. resource/synapeditor.config.js . . .

/synapeditor_plugin/resource/synapeditor.config.js

```
var synapEditorConfig = {
  "editor.license": "http://localhost/wordpress/wp-content/plugins/synapeditor/resource/license.json",
  "editor.import.api": "admin-ajax.php",
  "editor.import.param": {"action": "import_doc"},
  "editor.upload.image.api": "admin-ajax.php",
  "editor.upload.image.param": {"action": "upload_file"},
  "editor.upload.video.api": "admin-ajax.php",
  "editor.upload.video.param": {"action": "upload_file"},
  "editor.upload.file.api": "admin-ajax.php",
  "editor.upload.file.param": {"action": "upload_file"},
  ...
}
```

3. wordpress_plugin "wordpress/wp-content/plugins" .

4.

/synapeditor_plugin/index.php

```
<?php
/*
Plugin Name: synapeditor
Plugin URI: http://www.synapeditor.com
Description: Unlimited Rich Text Editor
Version: 0.1
Author: synapsoft
Author URI: http://www.synapsoft.co.kr
License: GPL 2
*/
/*
```

```

* remove current editor
*/
function wp_remove_editor() {
    remove_post_type_support( 'post', 'editor' );
}

/*
* add Synap Editor
*/
function wp_add_synapeditor() {
    wp_enqueue_script('synapeditor', plugin_dir_url(__FILE__) . 'resource/synapeditor.js', array( 'jquery'
));
    wp_enqueue_script('synapeditorConfig', plugin_dir_url(__FILE__) . 'resource/synapeditor.config.js');
    wp_enqueue_style('synapeditor', plugin_dir_url(__FILE__) . 'resource/synapeditor.min.css');

    $post = get_post(get_the_ID());
    $content = apply_filters('the_content', $post->post_content);

    echo "<textarea id=\"content\" name=\"content\" style=\"display: none;\">$content</textarea>";
    echo "<script>";
    echo "window.onload = function () {";
    echo "var html = document.getElementById('content').innerHTML;";
    echo "window.editor = new SynapEditor('content', synapEditorConfig, html);";
    echo "};";
    echo "</script>";
}
add_action('init', 'wp_remove_editor');
add_action('edit_form_after_title', 'wp_add_synapeditor' );

/*
* upload file
*/
function upload_file()
{
    $response = array();
    $uploaded_file = wp_handle_upload($_FILES['file'], array('test_form' => false));
    if ($uploaded_file && !isset($uploaded_file['error'])) {
        $response['uploadPath'] = parse_url($uploaded_file['url'])['path'];
    }

    echo json_encode($response);
    die();
};
add_action( 'wp_ajax_upload_file', 'upload_file' );
add_action( 'wp_ajax_nopriv_upload_file', 'upload_file' );

/*
* import document
*/
function import_doc()
{
    $response = array();
    $uploaded_file = wp_handle_upload($_FILES['file'], array('test_form' => false));
    if ($uploaded_file && !isset($uploaded_file['error'])) {
        $uploadDir = wp_upload_dir();
        $path_parts = pathinfo($uploaded_file['file']);
        $uploadFileName = $path_parts['filename'];

        // convert document
        $outputFilePath = join(DIRECTORY_SEPARATOR, array($uploadDir['path'], $uploadFileName));
        executeConverter($uploaded_file['file'], $outputFilePath);

        // serialized datas
        // v2.3.0 document.word.pb document.pb
        $pbFilePath = join(DIRECTORY_SEPARATOR, array($outputFilePath, "document.pb"));
        $serializedData = readPBData($pbFilePath);
        $outputFileUrl = $uploadDir['url'] . "/"{$uploadFileName}";
    } else {
        $outputFileUrl = $uploaded_file;
        $serializedData = !isset($uploaded_file['error']);
    }
}

```

```

    }

    echo json_encode(array(
        'serializedData' => $serializedData,
        'importPath' => $outputFileUrl
    ));
    die();
}

/*
 * convertor document
 */
function executeConverter($inputFilePath, $outputFilePath)
{
    $sedocConverterPath = plugin_dir_path(__FILE__) . 'resource\sedocConverter\sedocConverter.exe';
    $fontsDir = plugin_dir_path(__FILE__) . 'resource\sedocConverter\fonts';
    $tempDir = plugin_dir_path(__FILE__) . 'resource\sedocConverter\tmp';

    $cmd = "${sedocConverterPath} -f ${fontsDir} ${inputFilePath} ${outputFilePath} ${tempDir}";
    exec($cmd);
}

/*
 * serialized data
 */
function readPBData($pbFilePath)
{
    $fb = fopen($pbFilePath, 'r');
    $data = stream_get_contents($fb, -1, 16);
    fclose($fb);

    $byteArray = unpack('C*', zlib_decode($data));
    $serializedData = array_values($byteArray);

    return $serializedData;
}

add_action( 'wp_ajax_import_doc', 'import_doc' );
add_action( 'wp_ajax_nopriv_import_doc', 'import_doc' );

/**
 * Admin page setting
 */
function me_add_to_admin_menu() {
    add_menu_page( 'Synap Editor Configuration Page', 'Synap Editor Config',
        'manage_options', 'synap-editor-config', 'me_admin_menu' );
}

function me_admin_menu() {
    require( 'admin/synap-admin.php' );
}

function register_my_settings() {
    register_setting( 'synap-editor-conf', 'width' );
    register_setting( 'synap-editor-conf', 'height' );
    register_setting( 'synap-editor-conf', 'default_lang' );
    register_setting( 'synap-editor-conf', 'lang' );
    register_setting( 'synap-editor-conf', 'toolbar' );
}

if (is_admin()) {
    add_action('admin_menu', 'me_add_to_admin_menu');
    add_action('admin_init', 'register_my_settings');
}

```

/synapeditor_plugin/admin/synap-admin.php

```
<?php
/**
 * Created by synapeditor.
 */

if ( !current_user_can( 'manage_options' ) ) {
    wp_die( __( 'You do not have sufficient permissions to access this page.' ) );
}

?>

<div class="wrap">
    <h1>Synap Editor Configuration</h1>

    <form method="post" action="options.php">
        <?php settings_fields( 'synap-editor-conf' ); ?>
        <?php do_settings_sections( 'synap-editor-conf' ); ?>
        <label for="width"> </label>
        <input id="width" name="width" type="text" value="<?php echo esc_attr( get_option('width') ); ?>"/>
        <br/>
        <label for="height"> </label>
        <input id="height" name="height" type="text" value="<?php echo esc_attr( get_option('height') ); ?>"
    />

    <hr/>
    <label for="default_lang"> </label>
    <input id="default_lang" name="default_lang" type="text" value="<?php echo esc_attr( get_option
('default_lang') ); ?>"/>
    <br/>
    <label for="lang"> </label>
    <input id="lang" name="lang" type="text" value="<?php echo esc_attr( get_option('lang') ); ?>"/>
    <hr/>
    <fieldset>
        <legend> </legend>

        <?php
        $options = get_option( 'toolbar' );
        $toolbars = [
            ['bold', ''], ['italic', ''], ['underline', ''], ['strike', ''],
            ['superScript', ''], ['subScript', ''], ['fontColor', ''], ['fontBackgroundColor', ''],
            ['align', ''], ['copyRunStyle', ''], ['pasteRunStyle', ''], ['removeRunStyle', ''],
            ['link', ''], ['image', ''], ['backgroundImage', ''], ['video', ''],
            ['file', ''], ['horizontalLine', ''], ['specialCharacter', ''], ['emoji', ''],
            ['insertDiv', ''], ['bulletList', ''], ['numberedList', ''], ['multiList', '']
        ];

        $idx = 0;
        foreach ( $toolbars as $toolbar ) {
            echo "<input name='toolbar[$toolbar[0]]' type='checkbox' value='1'", checked( isset(
$options[$toolbar[0]] ) ) ,"/> $toolbar[1] ";
            if ( ++$idx % 4 == 0 ) {
                echo "<br/>";
            }
        }
    ?>
    </fieldset>

    <?php submit_button(); ?>
</form>
</div>
```

-
- [Java Spring Framework](#)
 - [Java Servlet](#)
 - [ASP.NET \(C#\)](#)
 - [ASP\(Classic\)](#)
 - [PHP](#)
 - [PHP4](#)
 - [Django](#)
 - [Ruby On Rails](#)
 - [Wordpress plugin](#)