Autocomplete

- Screenshots
- Settings
- Getting options remotely via Ajax

import Screenshots from '@site/src/components/Screenshots'; import Choice from '../_parts/_choice.md';

The autocomplete field creates a simple text input with autocomplete feature. You can select multiple values from the predefined list.

This field uses jQuery UI library to perform the autocomplete action.

Screenshots

<Screenshots name="autocomplete" col1={[['https://imgur.elightup.com/zvZI8qs.png', 'The autocomplete field interface']]} />

Settings

Besides the common settings, this field has the following specific settings, the keys are for use with code:

Name	Key	Description
Choices	`options`	List of choices, each per line. If you need to set values and labels, use the format "value: Label" for each choice. When using with code, this setting is an array of `'value' => 'Label'`. It can be an URL to a remote resource that returns the array of data in JSON format.
Size of the input box	`size`	Input size. Default `30`. Optional.

This is a sample field settings array for registering this field with code:

```
[
    'name'    => 'Autocomplete',
    'id'      => 'field_id',
    'type'    => 'autocomplete',
    'options' => [
        'java'       => 'Java',
        'javascript' => 'JavaScript',
        'php'        => 'PHP',
        'csharp'     => 'C#',
        'kotlin'     => 'Kotlin',
        'swift'      => 'Swift',
    ],
],
```

Getting options remotely via Ajax

In case you want to use remote data instead of user-defined data for the "Choices" (`options`) settings, you can set this setting as an URL of your remote data source.

For example, you can set the "Choices" with the value: `https://yourdomain.com/wp-admin/admin-ajax.php?action=something`, which will send an ajax request to the `admin-ajax.php` file, and then you can handle it with your function as follows:

```
add_action( 'wp_ajax_something', function() {
    $s = $_REQUEST[ 'term' ];
    // Do some stuff here to find matches.

    $response = [
        [ 'value' => '123', 'label' => 'Some Post' ],
        [ 'value' => '77', 'label' => 'Another Post' ],
    ];

    // Do some stuff to prepare JSON response ( headers, etc ).
    echo wp_json_encode( $response );
    die;
} );
```

Note that the data returned must be in JSON format as above. The ajax request also sends the search term via `$_REQUEST['term']` parameter as you see above.

□□□ #2
□ □□□ □□□ 2026-08-29 15:36:17 UTC
□ □□□ □□□ 2026-08-29 15:36:17 UTC