

The Country Field Type

This Field Type represents one or multiple countries.

Name	Internal name	Expected input
Country	ezcountry	array

Table of contents:

- [Description](#)
- [Input expectations](#)
- [Validation](#)
- [Settings](#)
- [To / From Hash format](#)
- [Value object](#)
 - [Properties](#)
 - [Constructor](#)

Description

This Field Type makes possible to store and retrieve data representing countries.

Input expectations

Type	Example
array	<pre>array("JP" => array("Name" => "Japan", "Alpha2" => "JP", "Alpha3" => "JPN", "IDC" => 81));</pre>

Note: When you set an array directly on Content field you don't need to provide all this information, the Field Type will assume it is a hash and in this case will accept a simplified structure described below under [To / From Hash format](#).

Validation

This Field Type validates if the multiple countries are allowed by the field definition, and if the Alpha2 is valid according to the countries configured in eZ Platform.

Settings

The field definition of this Field Type can be configured with one option:

Name	Type	Default value	Description
isMultiple	boolean	false	This setting allows (if true) or denies (if false) the selection of multiple countries.

Country FieldType example settings

```
$settings = array(  
    "isMultiple" => true  
);
```

To / From Hash format

The format used for serialization is simpler than the full format, this is also available when setting value on the content field, by setting the value to an array instead of the Value object. Example of that shown below:

Value object content example

```
$content->fields["countries"] = array( "JP", "NO" );
```

The format used by the toHash method is the Alpha2 value, however the input is capable of accepting either Name, Alpha2 or Alpha3 value as shown below in the Value object section.

Value object

Properties

The Value class of this field type contains the following properties:

Property	Type	Description
\$countries	array[]	This property will be used for the country selection provided as input, as its attributes.

Value object content example

```
$value->countries = array(  
    "JP" => array(  
        "Name" => "Japan",  
        "Alpha2" => "JP",  
        "Alpha3" => "JPN",  
        "IDC" => 81  
    )  
);
```

Constructor

The Country\Value constructor will initialize a new Value object with the value provided. It expects an array as input.

Constructor example

```
// Instantiates a Country Value object
$countryValue = new Country\Value(
    array(
        "JP" => array(
            "Name" => "Japan",
            "Alpha2" => "JP",
            "Alpha3" => "JPN",
            "IDC" => 81
        )
    )
);
```