

The Parameter Class

Parameter is an inner class of `UserInputParameter` and is a user interface definition object which tells Yellowfin how to display and persist data for a single configuration field.

Class Properties

Some properties and methods for the `Parameter` class are outlined below:

Property	Description
UniqueKey	A parameter's unique key is the key that is used by both Yellowfin and the plug-in to retrieve data and metadata for the parameter. It can be any String, as long as it is unique within your own plugin (does not have to be universally unique). This property has a getter and setter method.
DataType	This is one of the UIP data types which will be the storage type of the data value that this parameter holds. E.g. A parameter which holds a number should be stored as <code>TYPE_NUMERIC</code> , a checkbox parameter is likely to be <code>TYPE_BOOLEAN</code> , etc. This property has a getter and setter method. Refer to the appendix to see more data types.
DisplayType	This property tells Yellowfin how to display the Parameter's data. This value is one of the UIP display types. Different display types require different parameter setups. This property has a getter and setter method.
DefaultValue	Defines a default value for this parameter. If the parameter is 'reset', it will revert to this value, and if a user does not set a value for this parameter, its value will be stored as this value. This property has a getter and setter method.
DisplayName	This is the user-facing name of the parameter which in most cases will be shown next to the parameter UI element. This property has a getter and setter method.
Description	This is the user-facing short description of the parameter which in most cases will be shown next to the parameter UI element, near the name. This property has a getter and setter method.
MinAllowedValue	For use with numeric fields. This defines the minimum numeric value for this input. If the user attempts to enter something lower, it will not be accepted. This property has a getter and setter method.
MaxAllowedValue	For use with numeric fields. This defines the maximum numeric value for this input. If the user attempts to enter something higher, it will not be accepted. This property has a getter and setter method.
Options	Some parameter display types require additional options, such as dropdown boxes and URL buttons. These need to be added to the parameter object after instantiation. This property has getter and setter methods, as well as add methods which allow adding options with and without different display, value, and description text. This guide's appendix also includes documentation for required and optional options for each display type. Included below is an example of this.
ViewOptions	Some parameter display types have additional display configuration options called <code>ViewOptions</code> which are specific to the display type. It is recommended not to use these unless necessary, as they occasionally change and the documentation may not be fully up-to-date. These can also be found in the appendix .

Options Example

The example below refers to display types options property.

```
// Select dropdown parameter
Parameter p = new Parameter();
p.setUniqueKey("PUPPY_BREED");
p.setDisplayName("Select Puppy Breed");
p.setDescription("Which breed of puppy would you like?");
p.setDataType(TYPE_TEXT);
p.setDisplayType(DISPLAY_SELECT);
...
p.addOption("CHIHUAHUA"); // display text will also be CHIHUAHUA
p.addOption("SCHNAUZER", "Schnauzer"); // display text will be Schnauzer
...
// Button parameter
p = new Parameter();
p.setUniqueKey("URL");
p.setDisplayName("Access PIN");
p.setDescription("Connect to twitter to receive a PIN for data access");
p.setDataType(TYPE_UNKNOWN);
p.setDisplayType(DISPLAY_URLBUTTON);
...
p.addOption("BUTTONTEXT", "Request URL");
p.addOption("BUTTONURL", "http://google.com");
```

Previous topic: [UIP overview](#)

Next topic: [Basic UI set up](#)