

API

- - - - name
 - operator
 - displayOperator
 - uuid
 - possibleValues
 - between
 - list
 - values
 - valueOne
 - valueTwo
 - valueList
 - appliedValues
 - appliedValueOne
 - appliedValueTwo
 - appliedValueList
 - defaultValues
 - defaultValueOne
 - defaultValueTwo
 - defaultvValueList
 - - reset(apply)
 - clear(apply)
 - resetToLastAppliedState()
 - setValueOne(value, apply)ValueOne
 - setValueTwo(value, apply)ValueTwo
 - setValueList(valueList, apply)ValueList
 - setValue(value, apply)
 - applyFilter()
 - setPossibleValues(values)
 - addEventListener(eventName, callbackFunction)
 - removeEventListener(listenedId)
 - trigger(eventName, eventData)
 - - changed
 - applied
 - reset
 - cleared

APIgetFilter

値の入力方法

あらかじめ定義された一覧から、フィルターの値を選択します。

手入力 ☐

一覧から値を選択 ☒

Yellowfin

JavaScript

```
[[
  {
    value: 'AU', //The value that is actually applied to any queries
    description: 'Australia', //The human readable version of the value
  },
  {
    value: 'NZ',
    description: 'New Zealand'
  }
]]
```

valueSQLSQL

```
SELECT Country FROM CountryTable where Country = 'AU'
```

descriptionvaluedescription

name

String -

Yellowfin UI

Yes

operator

String - YellowfinSQL

Yes

- EQUAL - =
- NOTEQUAL -
- GREATER -
- GREATEREQUAL -
- LESS -
- LESSEQUAL -
- BETWEEN -
- NOTBETWEEN -
- INLIST -
- NOTINLIST -

DemographicINLIST

```
let filter = filter.get('Demographic');
console.log(filter.operator); //outputs 'INLIST'
```

displayOperator

String -

Yes

- EQUAL - =
- NOTEQUAL -
- GREATER -
- GREATEREQUAL -
- LESS -
- LESSEQUAL -
- BETWEEN -
- NOTBETWEEN -
- INLIST -
- NOTINLIST -

```
let filter = filter.get('Demographic');
console.log(filter.displayOperator); //outputs 'In List'
```

filter.operator

uuid

String - UUID

Yes

possibleValues

Array[Object]

```
[[
  value: 'AU', //The value that is actually applied to any queries
  description: 'Australia', //The human readable version of the value
},{
  value: 'NZ',
  description: 'New Zealand'
}]
```

null

possibleValues

```
let possibleValues = filter.possibleValues; //Get the possible values out of the filter
let newFilterValues = []; //Create a new array that will be applied to the valueList function
possibleValues.forEach(valueObject => { //Iterate over the possibleValues and then push the value property to the newFilterValues array
  newFilterValues.push(valueObject.value);
});

filter.setValueList(newFilterValues, true);
```

between

Boolean

BetweenNot BetweenBetweentrueBETWEEN

Yes

list

Boolean

In ListNot In Listtrue

Yes

values

- valueOne
- valueTwo
- valueList

valueList

valueOne

betweenvalueOnevalueTwo

Yes

in List

```
let filter = filters.getFilter('Demographic');
filter.setValues(['Adventure']);

console.log(filter.values); //Outputs { valueList: ['Adventure'] }
```

Between

```
let filter = filters.getFilter('Age at Camp');
filter.setValueOne(15);
filter.setValueTwo(35);
console.log(filter.values); //Outputs { valueOne: 15, valueTwo: 35 };
```

```
let filter = filters.getFilter('Age at Camp');  
filter.setValueOne(15);  
console.log(filter.values); //Outputs { valueOne: 15 };
```

valueOne

StringNumber

valueOnenull

True

```
let filter = filters.getFilter('Average Age at Camp');  
filter.setValueOne(35);  
console.log(filter.valueOne); //Outputs '35'
```

valueTwo

StringNumber

valueTwonull

True

```
let filter = filters.getFilter('Average Age at Camp');  
filter.setValueTwo(65);  
console.log(filter.valueTwo); //Outputs '65'
```

valueList

Array{String/Number}

valueListBetweennull

True

```
let filter = filters.getFilter('Demographic');  
filter.setValueList(['Adventure', 'Luxury']);  
console.log(filter.valueList); //Outputs ['Adventure', 'Luxury']
```

appliedValues

- valueOne
- valueTwo
- valueList

valueList

valueOne

betweenvalueOnevalueTwo

Yes

in List

```
let filter = filters.getFilter('Demographic');
filter.setValues(['Adventure'], true); //Set the value and apply

console.log(filter.appliedValues); //Outputs { valueList: ['Adventure'] }
```

Between

```
let filter = filters.getFilter('Age at Camp');
filter.setValueOne(15);
filter.setValueTwo(35);
filter.applyFilter(); //Apply the values
console.log(filter.appliedValues); //Outputs { valueOne: 15, valueTwo: 35 };
```

```
let filter = filters.getFilter('Age at Camp');
filter.setValueOne(15, true);
console.log(filter.appliedValues); //Outputs { valueOne: 15 };
```

appliedValueOne

StringNumber

valueOnenull

True

```
let filter = filters.getFilter('Average Age at Camp');
filter.setValueTwo(35, true); //Set valueTwo and apply it
console.log(filter.appliedValueOne); //Outputs '35'
```

appliedValueTwo

StringNumber

valueTwobetweennull

True

```
let filter = filters.getFilter('Average Age at Camp');
filter.setValueTwo(65, true); //Set valueTwo and apply it
console.log(filter.appliedValueTwo); //Outputs '65'
```

appliedValueList

Array{String/Number}

valueListBetweennull

True

```
let filter = filters.getFilter('Demographic');
filter.setValueList(['Adventure', 'Luxury']);

filter.applyFilter();
console.log(filter.appliedValueList); //Outputs ['Adventure', 'Luxury']
```

defaultValues

- valueOne
- valueTwo
- valueList

betweenvalueOnevalueTwonull

valueList

valueOne

betweenvalueOnevalueTwo

Yes

in List

```
let filter = filters.getFilter('Demographic');
console.log(filter.defaultValues); //Outputs { valueList: ['Adventure'] }
```

Between

```
let filter = filters.getFilter('Age at Camp');  
console.log(filter.defaultValues); //Outputs { valueOne: 15, valueTwo: 35 };
```

```
let filter = filters.getFilter('Age at Camp');  
console.log(filter.defaultValues); //Outputs { valueOne: 15 };
```

defaultValueOne

StringNumber

valueOnenull

True

```
let filter = filters.getFilter('Average Age at Camp');  
console.log(filter.defaultValueOne);
```

defaultValueTwo

StringNumber

valueTwobetweennull

True

```
let filter = filters.getFilter('Average Age at Camp');  
console.log(filter.defaultValueTwo);
```

defaultvValueList

Array{String/Number}

valueListnull

True

```
let filter = filters.getFilter('Demographic');  
console.log(filter.defaultValueList);
```

reset(apply)

applytrue

Apply - Boolean - True

Athlete Country['AU','NZ']

```
filter.setValue(['UK', 'US']);  
console.log(filter.values); //Outputs {valueList: ['UK', 'US'] }  
filter.reset();  
console.log(filter.values); //Outputs {valueList: ['AU', 'NZ']}
```

clear(apply)

applytrue

Apply - Boolean - True

Athlete Country['AU','NZ']

```
filter.setValue(['UK', 'US']);  
console.log(filter.values); //Outputs {valueList: ['UK', 'US'] }  
filter.clear();  
console.log(filter.values); //Outputs {valueList: null}
```

resetToLastAppliedState()

```
filter.setValue(['UK', 'US']);  
filter.apply(); //Set the value on the report  
  
filter.setValue(['DE']);  
console.log(filter.values); //Outputs {valueList: ['DE'] }  
  
filter.resetToLastAppliedState();  
  
console.log(filter.values); //Outputs {valueList: ['UK', 'US']}
```

setValueOne(value, apply)ValueOne

valueOneapplytrue

NullvalueOne

In ListNot In ListSetValueOne

value - String, Number

apply - Boolean - false

```
filter.setValueOne('Relaxation'); //Change the value to Relaxtion but don't immediately run the report  
console.log(filter.valueOne); //Should return 'Relaxation'
```

setValueTwo(value, apply)ValueTwo

valueTwoapplytrue

NullvalueTwo

BetweenNot BetweenSetValueTwo

value - String, Number

apply - Boolean - false

```
//Filter is a between filter (Average Age at Camp)  
filter.setValueOne(15);  
filter.setValueTwo(35);  
  
console.log(filter.valueOne + " to " + filter.valueTwo); //Should output "15 to 35"
```

setValueList(valueList, apply)ValueList

ValueListapplytrue

valueOne

betweenvalueOnevalueTwo

valueList - Array

apply - Boolean - false

```
//Set the value of the filter "Demographic" to be Adventure, Family, Sport
filter.setValueList(['Adventure', 'Family', 'Sport']);

//Set the values and immediately apply the filter
filter.setValueList(['Adventure', 'Relaxation'], true);

//Can also be used to set the between values in Average Age at Camp
filter.setValueList([15, 35]);

filter.setValueList([15]);
```

setValue(value, apply)

applytrue

value

valueNumberString

listfilter

```
filter.setValueList(['single value']);
```

betweenvalueOne

```
filter.setValueOne('first value');
```

valueArray

valueList

betweenvalueOne

betweenvalueTwo

```
filter.setValue([15, 35]);
```

between

```
filter.setValueOne(15);  
filter.setValueTwo(35);
```

value

betweenvalueOnevalueOne

betweenvalueTwovalueTwo

valueListvalueList

value - String, Number, Array, Object

apply - Boolean - false

between

```
let filter = filters.getFilter('Age at Camp');  
filter.setValue({  
  valueOne: 15, //Sets valueOne to 15  
  valueTwo: 35 //Sets valueTwo to 35  
  valueList: [1,2,3,4,5] //Ignored  
});  
  
filter.setValue({  
  valueOne: 15, //Sets valueOne to 15  
  valueTwo: 35 //Sets valueTwo to 35  
});
```

```
let filter = filters.getFilter('Demographic');  
filter.setValue(['Adventure']); //Using an array  
  
filter.setValue({ valueList: ['Adventure'] }); //Using an object
```

applyFilter()

valueOnevalueTwovalueListvalueOneappliedValueOnevalueTwoappliedValueTwovalueListappliedValueListapplied

DemographicAdventure

```
let filter = filters.getFilter('Demographic');  
filter.setValue(['Adventure']);  
filter.applyFilter();
```

setPossibleValues(values)

possiblevaluesYellowfin UI null

valueDescription

valueSQLdescriptiondescription

```
{
  value: {Number, String},
  description: {String}
}
```

DemographicDemographic

```
let possibleValues = filter.possibleValues;
possibleValues.push({
  value: 'Relaxation',
  description: 'Relaxation'
});
filter.setPossibleValues(possibleValues);
```

```
let possibleValues = [];
possibleValues.push({
  value: 'FIRST', //The data that is stored in the table for this filter is upper case.
  description: 'First' //Upper case can be painful to read, so put a more readable version to be displayed in the description
});

possibleValues.push({
  value: 'SECOND',
  description: 'Second'
});

filter.setPossibleValues(possibleValues);
```

addEventListener(eventName, callbackFunction)

Number

callbackFunction

IDIDremoveEventListenerID

API

.trigger()

```
let eventId = filter.addEventListener('changed', function(event) {
  console.log(event.filter.name + ' changed value');
  filter.removeEventListener(eventId);
});
```

removeEventListener(listenedId)

ID

```
let eventId = filters.addEventListener('changed', function(event) {
  console.log('One of my filters changed');
  filters.removeEventListener(eventId);
});
```

trigger(eventName, eventData)

```
//Add a 'userClick' listener to the filter object, which we will set up a trigger for later on.
filters.addEventListener('userClick', function(event) {
  console.log('A user clicked on the element ' + event.element + ' which is tied to this filter');
});
//Get the custom filter list from the DOM and create a click listener on that which will trigger userClicked events on the filter
let myCustomFilterList = document.querySelector('div#customFilterList')
myCustomFilterList .addEventListener('click', function(e) {
  filters.trigger('userClicked', { element: e.currentTarget });
});
```

UUID

```
filter.addEventListener('changed', function(event) {
  console.log(event.uuid); //The filters UUID that the event was triggered from
  console.log(event.filter); //The FilterObject that the event was triggered from
});
```

changed

Event - Object

- uuid - String - UUID
- filter - FilterObject -
- changed - Object - valueOnevalueTwovalueList
- Previous - Object - valueOnevalueTwovalueList

```
filter.addEventListener('applied', function(event) {  
  console.log(event.filter.name + " has just been applied with the following changed values " + JSON.stringify(event.changed));  
});
```

applied

Event - Object

- uuid - String - UUID
- filter - FilterObject -
- changed - Object - valueOnevalueTwovalueList
- Previous - Object - valueOnevalueTwovalueList

```
filter.addEventListener('applied', function(event) {  
  console.log(event.filter.name + " has just been applied with the following changed values " + JSON.stringify(event.changed));  
});
```

```
let filter = filters.getFilter('Demographic');  
filter.setValue(['Adventure']);  
filter.apply();  
  
filter.setValue(['Adventure']);  
filter.apply();
```

setValue

reset

```
filter.addEventListener('reset', function(event) {  
  console.log(event.filter.name + " has just been reset");  
});
```

Event - Object

- uuid - String - UUID
- filter - FilterObject -

cleared

clearedreset

Event - Object

- uuid - String - UUID
- filter - FilterObject -

```
filter.addEventListener(cleared, function(event) {  
  console.log(event.filter.name + " has just had its values cleared");  
});
```