API

-
  -
  -
  -
  -
  -
    - getFilter (filterId)
      -
    - getAllFilters()
    - forEach(fn)
    - hasFilters()
      - clearFilter(filterId)
      -
      -
    - clear()
    - applyFilters()
      -
    - LoadFilters()
    - resetFiltersToDefault()
    - addEventListener(eventName, callbackFunction)
    - removeEventListener(listenerId)
      -
    - trigger(eventName, eventData)
  -
    - changed
    - applied
    - reset
    - cleared

API2

- API
- API

Yellowfin

API

report.filters

dashboard.filters

API

2

APIgetAppliedValue

APIsetValuegetValue

defaultValues

valueOnevalueTwoappliedValueOne

Yellowfin

ValueOneValueOneValueTwo

ValueList

Yellowfin

ID

UUIDUUID

```
report.filters.getFilter('47fe96c2-5101-4b0d-9018-7d12a84d3519');
```

UUID

```
dashboard.filters.getFilter('47fe96c2-5101-4b0d-9018-7d12a84d3519'); //Will return null
```

API

---

getFilter (filterId)

IDfilterIdnull

filterId - (String, Number)

UUID

Demographic

```
let filter = filters.get('Demographic');
console.log(filter.name, filter.uuid); //Output the name of the filter as well as its UUID
```

UUID

```
let filter = filters.get('47fe96c2-5101-4b0d-9018-7d12a84d3519');
console.log(filter.name, filter.uuid); //Output the name of the filter as well as its UUID
```

getAllFilters()

Object - {String, FilterObject}

API UUID

APIgetAllFilters()[uuid]filters.getFilter(filterId)getFiltersUUIDfilters.forEach(fn)

forEach(fn)

APIfn

fn -

```
let appliedFilterValues = [];
filters.forEach(filter => {
        appliedFilterValues.push(Object.assign({
                name: filter.name,
                uuid: filter.uuid,
        }, filter.appliedValues));
```

UUID

hasFilters()

Boolean

truefalse

API

```
if(filters.hasFilters()) {
        generateMyCustomFilterPanel(filters);
}
```

clearFilter(filterId)

filterIdID

filterId - (String)

UUID

clear()

API

applyFilters()

Average Age at CampDemographic

```
let demographic = filters.getFilter('Demographic');
let ageAtCamp = filters.getFilter('Average Age at Camp');

demographic.setValue(['Adventure']);
ageAtCamp.setValue([15, 35]);

filters.applyFilters();
```

```
document.querySelector('div#applyButton').addEventListener('click', function(e) {
        filters.applyFilters();
});
```

LoadFilters()

Promise

Promise

resetFiltersToDefault()

```
filters.resetFiltersToDefault();
```

addEventListener(eventName, callbackFunction)

Number

callbackFunction

IDIDremoveEventListenerID

API API#

.trigger()

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

removeEventListener(listenerId)

ID

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

trigger(eventName, eventData)

API

```
//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 } );
});
```

API

- **eventName** - String -
- **filterEvents** - Array{Object} -
    - **uuid** - UUID
    - **filter** -
    -

changed

API

Event - Object

- eventName - "changed"
- filterEvents - Array{Object} - API

```
let filters = report.filters;

filters.addEventListener('changed', function(event) {
            console.log(event.eventName); //'changed'
            console.log(event.filterEvents); //[{ uuid: 'a-filter-uuid', filter: FilterObject, changed: Object, previous Object }, {....}]
});

filters.setFilterValue('Demographic', ['Adventure']);
```

applied

APIapplyFilters

Event - Object

- eventName - "applied"
- filterEvents - Array{Object} - API

```
let filters = report.filters;

filters.addEventListener('applied', function(event) {
        console.log(event.eventName); //'applied'
        console.log(event.filterEvents); //[{ uuid: 'a-filter-uuid', filter: FilterObject, changed: Object, previous Object }, {....}]
});

filters.setFilterValue('Demographic', ['Adventure']);

filters.applyFilters();
```

reset

API

Event - Object

- eventName - "reset"
- filterEvents - Array{Object} - API

```
let filters = report.filters;

filters.addEventListener('reset', function(event) {
        console.log(event.eventName); //'changed'
        console.log(event.filterEvents); //[{ uuid: 'a-filter-uuid', filter: FilterObject}, {....}]
});

filters.setFilterValue('Demographic', ['Adventure']);
filters.resetFiltersToDefault();

filters.getFilter('Demographic').reset(); //Trigger reset on an individual filter
```

cleared

API

Event - Object

- eventName - "cleared"
- filterEvents - Array{Object} - API

```
let filters = report.filters;

filters.addEventListener('reset', function(event) {
        console.log(event.eventName); //'changed'
        console.log(event.filterEvents); //[{ uuid: 'a-filter-uuid', filter: FilterObject}, {....}]
});

filters.setFilterValue('Demographic', ['Adventure']);

filters.applyFilters();
```