

# Frontend Internalization Instruction

For the Data Filter Web Application  
from Team Gabriel

Daniel Jarvis; Youhai Li; Qiwen Liu;  
Kenneth Wong

UNIVERSITY OF TORONTO – CSC301 TEAM GABRIEL

# INDEX

Mapbox API Token .....	2
Connection with Server .....	2
Error Handling .....	2
Adding and Changing Time Range Presets .....	2 - 3

## Mapbox API Token

On line 25 in “frontend/src/components/Map/Mapbox.jsx”, there is a mapboxgl access token. You must change this token to a valid mapbox token when internalizing the app.

## Connection with Server

Once the backend is internalized and deployed, please modify line 7 in “frontend/src/actions/actions.js” to the new backend server link. Also, if you wish to test it locally, please modify line 8 to the correct localhost port.

## Error Handling

In most cases if an error occurred, that means there are some invalid or missing required fields in the data query. Please see the error message, which is directly returned from the backend to debug. However, since some actions have interactions with the Mapbox component, the source of error may also be some invalid operations on the Mapbox. If you wish to figure out the error, please modify line 14 in “frontend/src/actions/actions.js” the if-clause to remove “|| !err.response” and add “console.log(err)”.

## Adding and Changing Time Range Presets

In the folder “frontend/src/components/Layout/Datetime” there is a file called Presets.js, which contains the presets that are used in the application.

1. To change an existing preset you can simply change the value stored in its fields.
2. To rename an existing preset, change the name that acts as the key to the preset parameters and the value of the preset field to match it. Then, in the getParams function, change the name of the preset from the old one to the new one in both the ‘case’ and ‘this[]’ sections. Finally, in the getPresets function, change the preset from the old name to the new name.

3. To add a new preset, after the last preset, create a name for the new preset and enclose it in single quotes (e.g. 'New Preset'). Then, in braces, include the parameters that you would like to set with the preset, separated by commas, valid parameters are `startTime` and `endTime`. The start and end times must be set as `new Date( your date time )`, where 'your date time' is formatted as 'yyyy-mm-dd hh:mm:ss', the date portion must be included, however the value itself is irrelevant so the current date may be used. Also in the braces you must include the name of the preset, with the key 'preset and the name in single or double quotes'. In the `getPresets` function add a new case before the default and follow it with the name of the new preset. On the line below that write `'return this["name"]'`, where "name" is the name of the new preset. Finally, in the `getPresets` function, add the name of the new preset in the list of preset names enclosed by square brackets.

Example:

```
'New Preset': {
    startTime: new Date("2020-01-01 00:00:00"),
    endTime: new Date("2020-01-01 23:59:59"),
    preset: 'New Preset'
},

getParams(preset){
    switch(preset){
        ...
        case 'New Preset':
            return this['New Preset']
    }
},

getPresets(){
    return [..., 'New Preset']
}
```