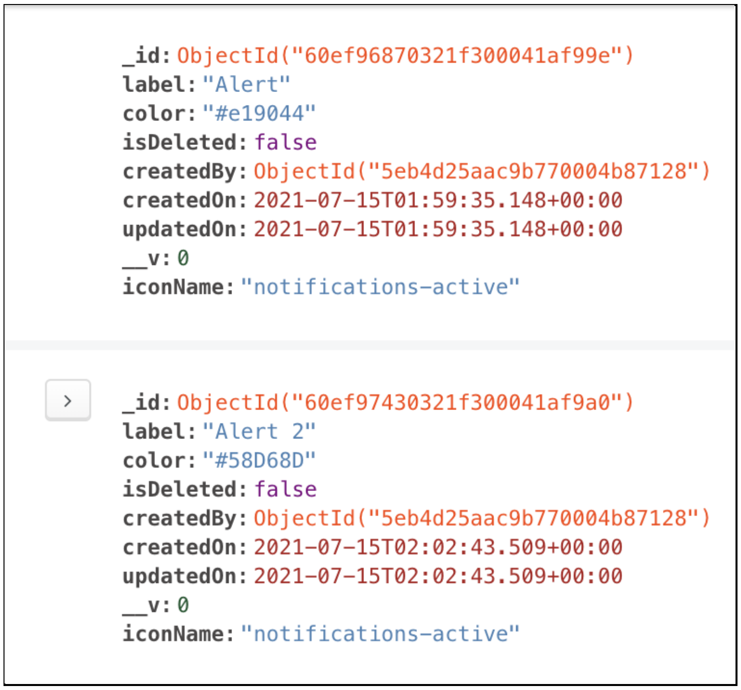
Managing Alert Types

The collection “geofenceareatypes” holds all the different alert types the app manages.
When defining a new alert type you tell it what label to display in the app, the color to use (hex html color code), and the icon name.

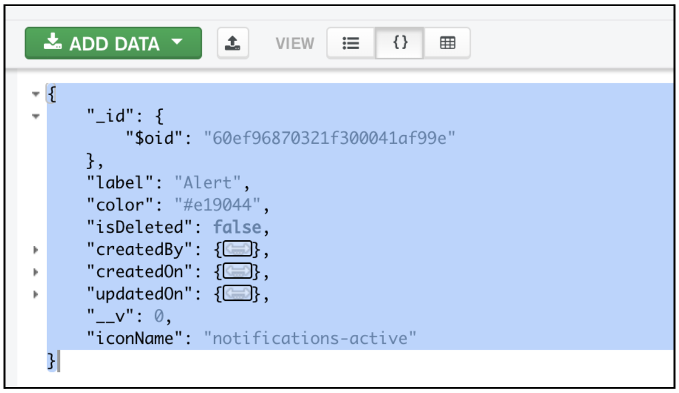


```
_id: ObjectId("60ef96870321f300041af99e")
label: "Alert"
color: "#e19044"
isDeleted: false
createdBy: ObjectId("5eb4d25aac9b770004b87128")
createdOn: 2021-07-15T01:59:35.148+00:00
updatedOn: 2021-07-15T01:59:35.148+00:00
__v: 0
iconName: "notifications-active"
```

```
_id: ObjectId("60ef97430321f300041af9a0")
label: "Alert 2"
color: "#58D68D"
isDeleted: false
createdBy: ObjectId("5eb4d25aac9b770004b87128")
createdOn: 2021-07-15T02:02:43.509+00:00
updatedOn: 2021-07-15T02:02:43.509+00:00
__v: 0
iconName: "notifications-active"
```

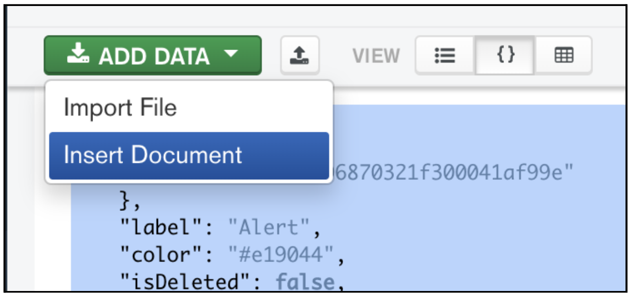
Adding a New Alert Type

1. Copy and paste the JSON for an existing alert type.



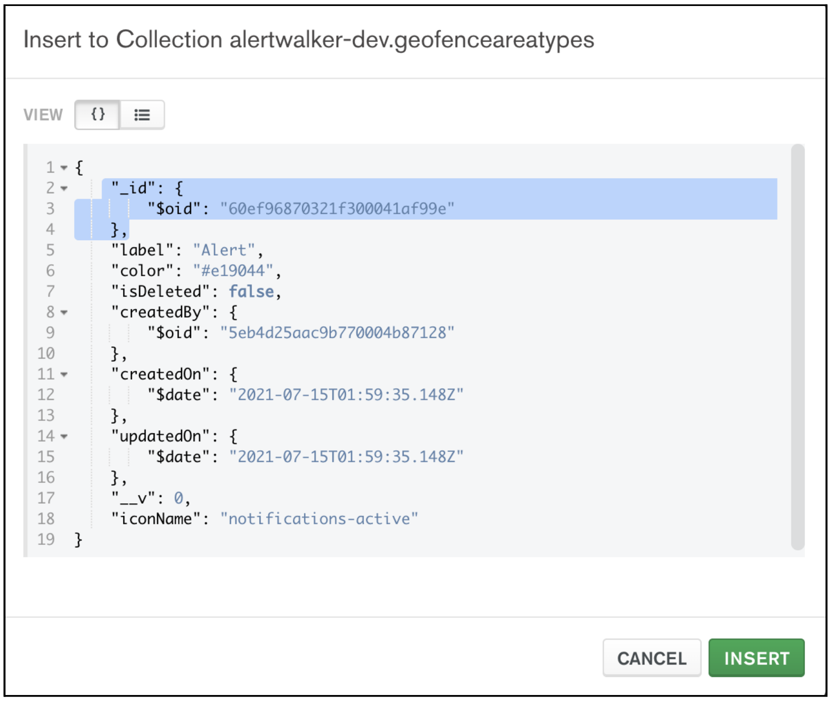
```
{
    "_id": {
        "$oid": "60ef96870321f300041af99e"
    },
    "label": "Alert",
    "color": "#e19044",
    "isDeleted": false,
    "createdBy": {…},
    "createdOn": {…},
    "updatedOn": {…},
    "__v": 0,
    "iconName": "notifications-active"
}
```

2. Insert a new document



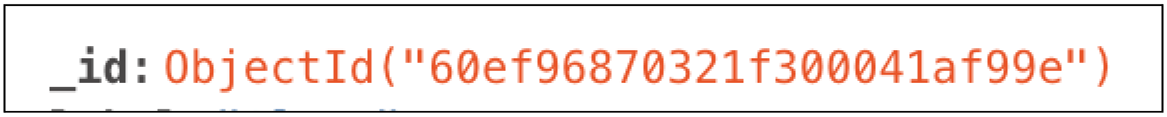
```
6870321f300041af99e"
    },
    "label": "Alert",
    "color": "#e19044",
    "isDeleted": false,
```

3. Delete the _id field completely otherwise you will overwrite the existing document with that ID.


Insert to Collection alertwalker-dev.geofenceareatypes

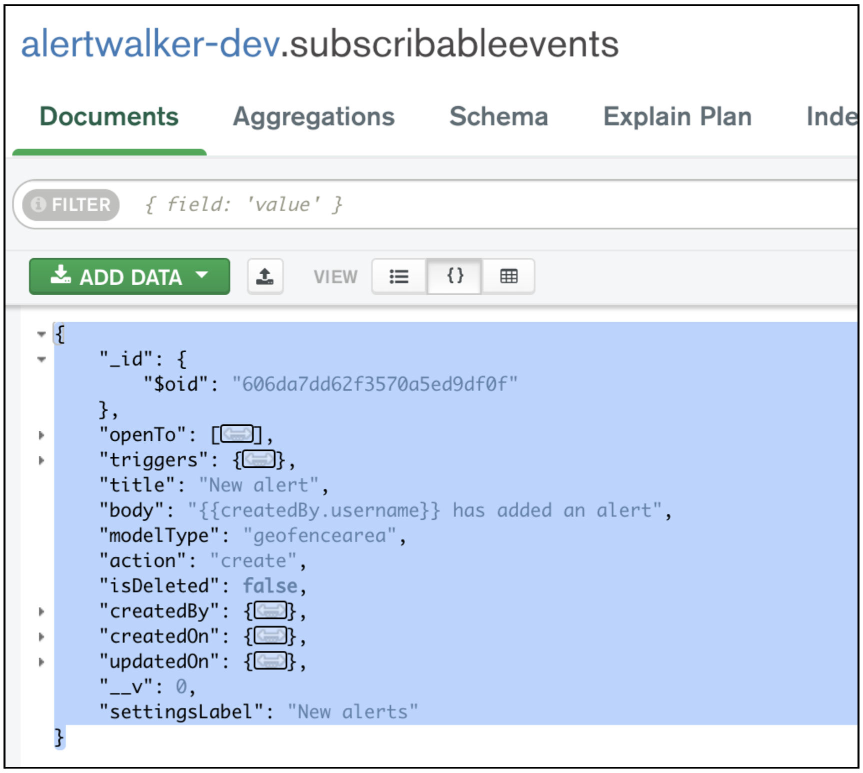
```
{
    "_id": {
        "$oid": "60ef96870321f300041af99e"
    },
    "label": "Alert",
    "color": "#e19044",
    "isDeleted": false,
    "createdBy": {
        "$oid": "5eb4d25aac9b770004b87128"
    },
    "createdOn": {
        "$date": "2021-07-15T01:59:35.148Z"
    },
    "updatedOn": {
        "$date": "2021-07-15T01:59:35.148Z"
    },
    "__v": 0,
    "iconName": "notifications-active"
}
```

4. Fill in the new label, color, and icon name. Then click insert.
   Make a note of the object ID for this new alert type.



```
_id: ObjectId("60ef96870321f300041af99e")
```

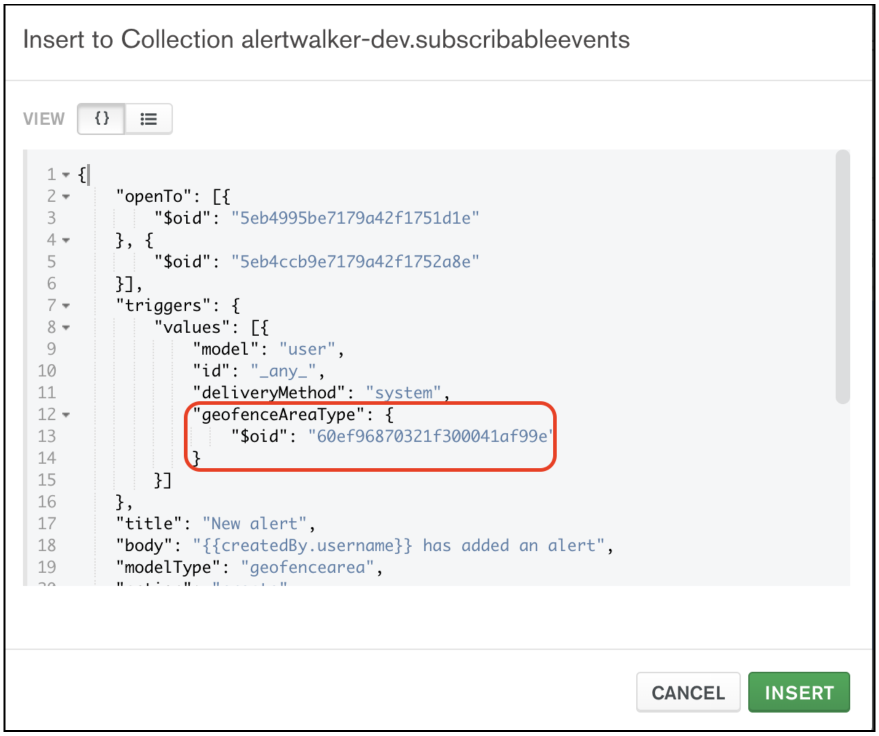
5. Next you need to create a new subscribable event for this new alert type.
   Navigate to the subscribableevents collection and copy the JSON for an existing document. Now add a new document like you did previously.
   Paste in this JSON and remove the _id field.



alertwalker-dev.subscribableevents

```
{
    "_id": {
        "$oid": "606da7dd62f3570a5ed9df0f"
    },
    "openTo": [⋯],
    "triggers": {⋯},
    "title": "New alert",
    "body": "{{createdBy.username}} has added an alert",
    "modelType": "geofencearea",
    "action": "create",
    "isDeleted": false,
    "createdBy": {⋯},
    "createdOn": {⋯},
    "updatedOn": {⋯},
    "__v": 0,
    "settingsLabel": "New alerts"
}
```

6. Replace the triggers.values.geofenceAreaType field value with the object ID of the new geofenceareatype you created in step 4.

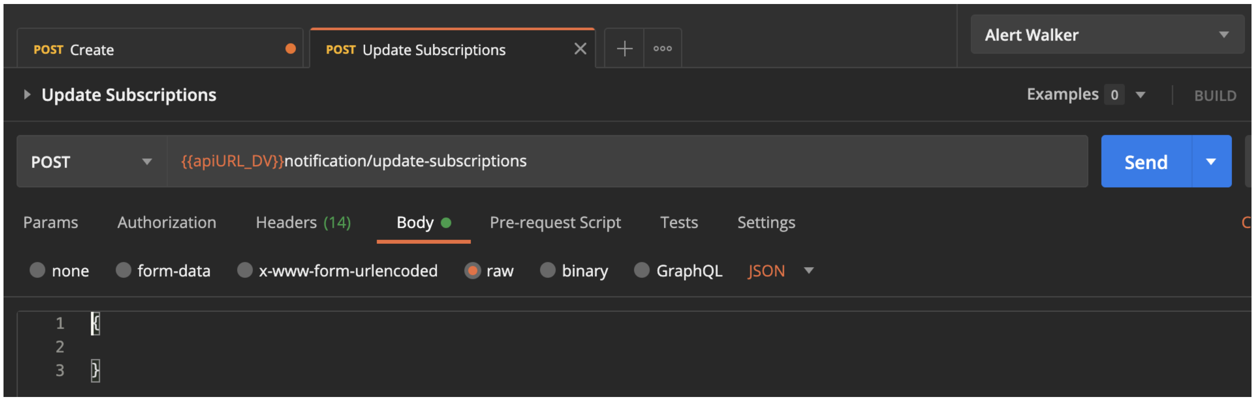


Insert to Collection alertwalker-dev.subscribableevents

```
{
    "openTo": [{
        "$oid": "5eb4995be7179a42f1751d1e"
    }, {
        "$oid": "5eb4ccb9e7179a42f1752a8e"
    }],
    "triggers": {
        "values": [{
            "model": "user",
            "id": "_any_",
            "deliveryMethod": "system",
            "geofenceAreaType": {
                "$oid": "60ef96870321f300041af99e"
            }
        }]
    },
    "title": "New alert",
    "body": "{{createdBy.username}} has added an alert",
    "modelType": "geofencearea",
    "action": "create"
```

7. Now that we have a new subscribable event we need to have an event subscription created for this for each user. Rather than doing it manually just invoke the route notification/update-subscriptions as an admin user and it will do it for you.

   I have a PostMan collection setup to do this which makes it much easier.

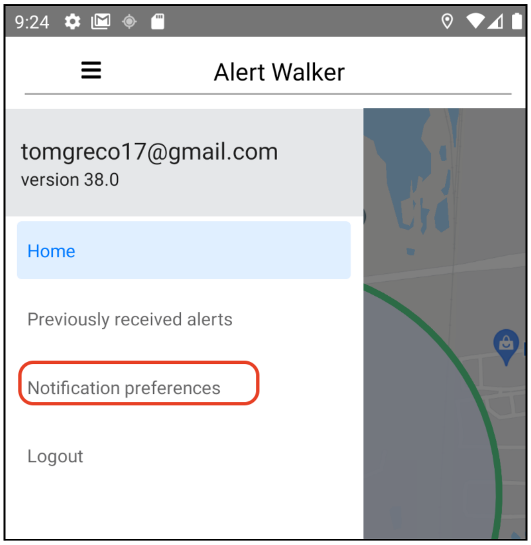


POST {{apiURL_DV}}notification/update-subscriptions

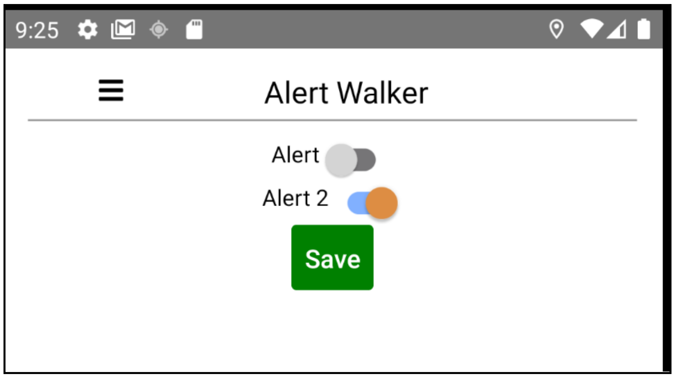
```
{

}
```

Important Information

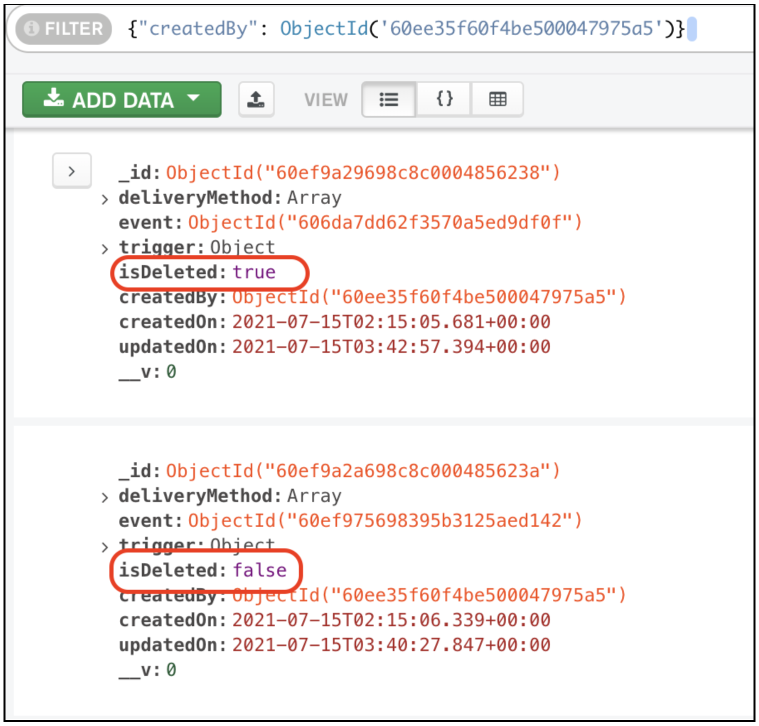
Now in the app there is a new notification preferences page which will allow a user to subscribe/unsubscribe to certain alert types.



The switches that appear on this page are populated from the user’s eventsubscription documents.

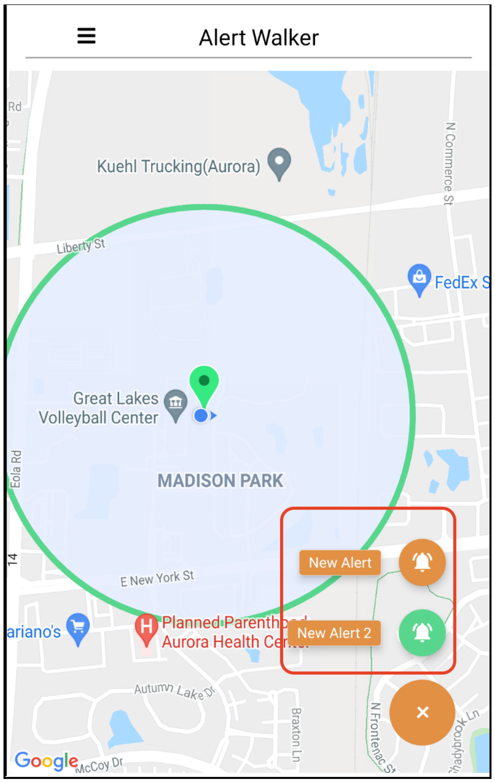


For each document for a different event subscription you will see a switch appear in the list.
When they set it to disabled and save all this does it update the isDeleted field on the document to true so that it no longer appears as a valid event subscription to the backend.
Yet the record still exists so we can properly display the switch on this page.
Enabling the switch and saving updates the isDeleted field to false so the system honors this event subscription and sends out notifications for it.



```
{"createdBy": ObjectId('60ee35f60f4be500047975a5')}

_id: ObjectId("60ef9a29698c8c0004856238")
> deliveryMethod: Array
event: ObjectId("606da7dd62f3570a5ed9df0f")
> trigger: Object
isDeleted: true
createdBy: ObjectId("60ee35f60f4be500047975a5")
createdOn: 2021-07-15T02:15:05.681+00:00
updatedOn: 2021-07-15T03:42:57.394+00:00
__v: 0

_id: ObjectId("60ef9a2a698c8c000485623a")
> deliveryMethod: Array
event: ObjectId("60ef975698395b3125aed142")
> trigger: Object
isDeleted: false
createdBy: ObjectId("60ee35f60f4be500047975a5")
createdOn: 2021-07-15T02:15:06.339+00:00
updatedOn: 2021-07-15T03:40:27.847+00:00
__v: 0
```

The alert type floating action button items are now populated dynamically from the geofenceareatypes collection.



When a user creates a new geofence alert by invoking the data/create route.
The subscription preferences are honored and it also filters by only users in the area to notify them via websocket and populate the map with that new alert.

When a user moves around and their location updates the location/geofence route is invoked.
This will now also honor the event subscription preferences.