I attended the Umbraco Examine Course & you won't believe what happened next

#CG18

Oh HAI ! 👋

#NotJustTheFormsGuy

Courses

Lucene, Examine, huh?!

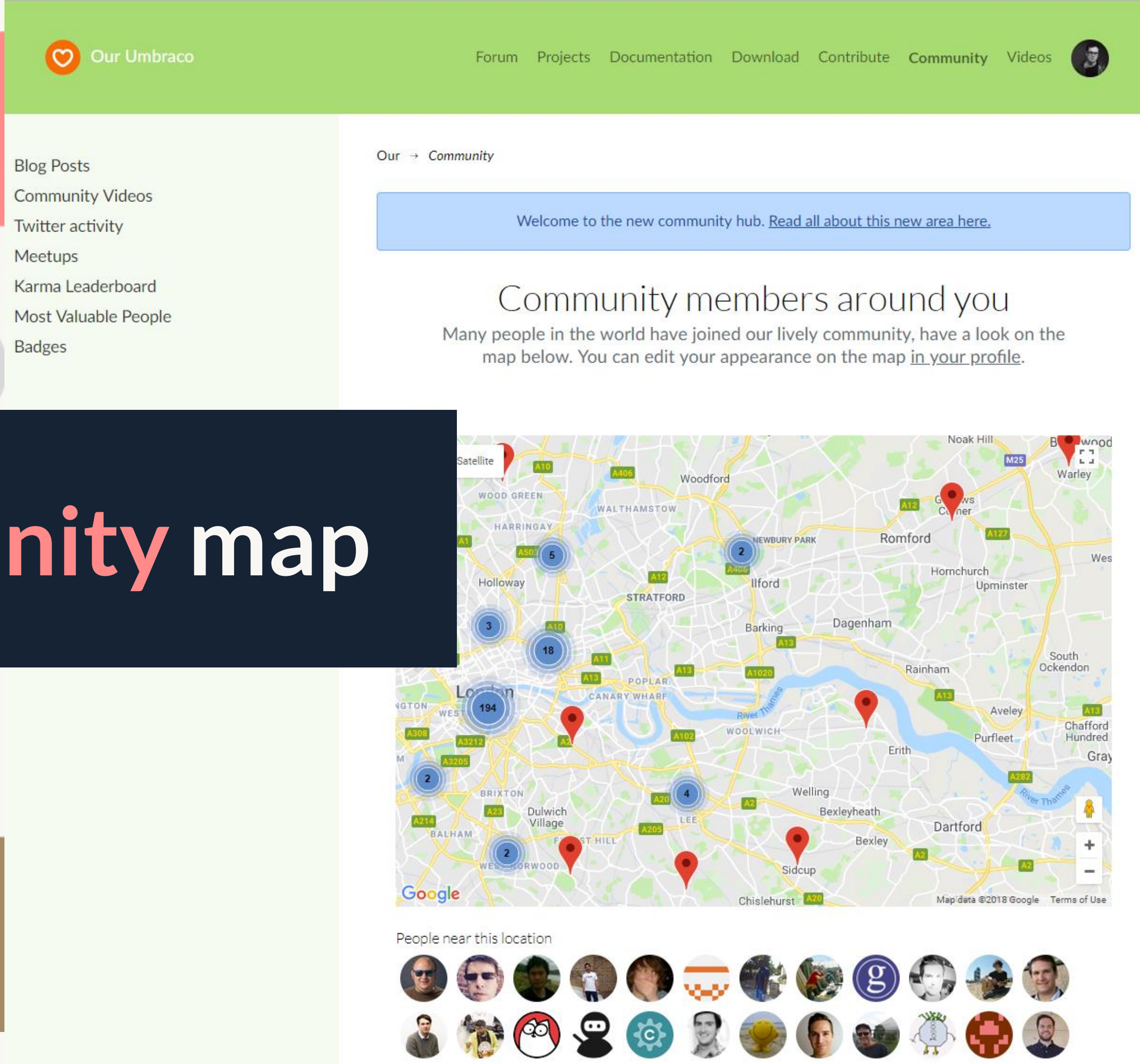
#CG18

- **Lucene**
  The underlying tech that allows us to do powerful searches & queries quickly

- **Examine**
  The .NET Library from Shannon that makes working with Lucene easier with a fluent API - ie easier to use & consume

So what happened?

I built a community map



Our Umbraco

- Blog Posts
- Community Videos
- Twitter activity
- Meetups
- Karma Leaderboard
- Most Valuable People
- Badges

Our → *Community*

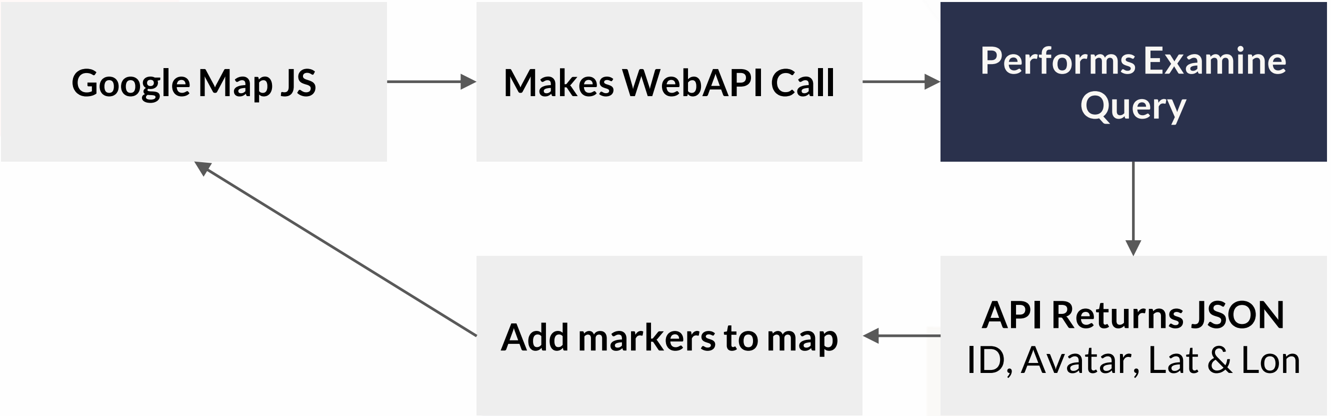
Welcome to the new community hub. Read all about this new area here.

Community members around you

Many people in the world have joined our lively community, have a look on the map below. You can edit your appearance on the map in your profile.

People near this location

The basic flow



#CG18

Finding members with location data

Problem

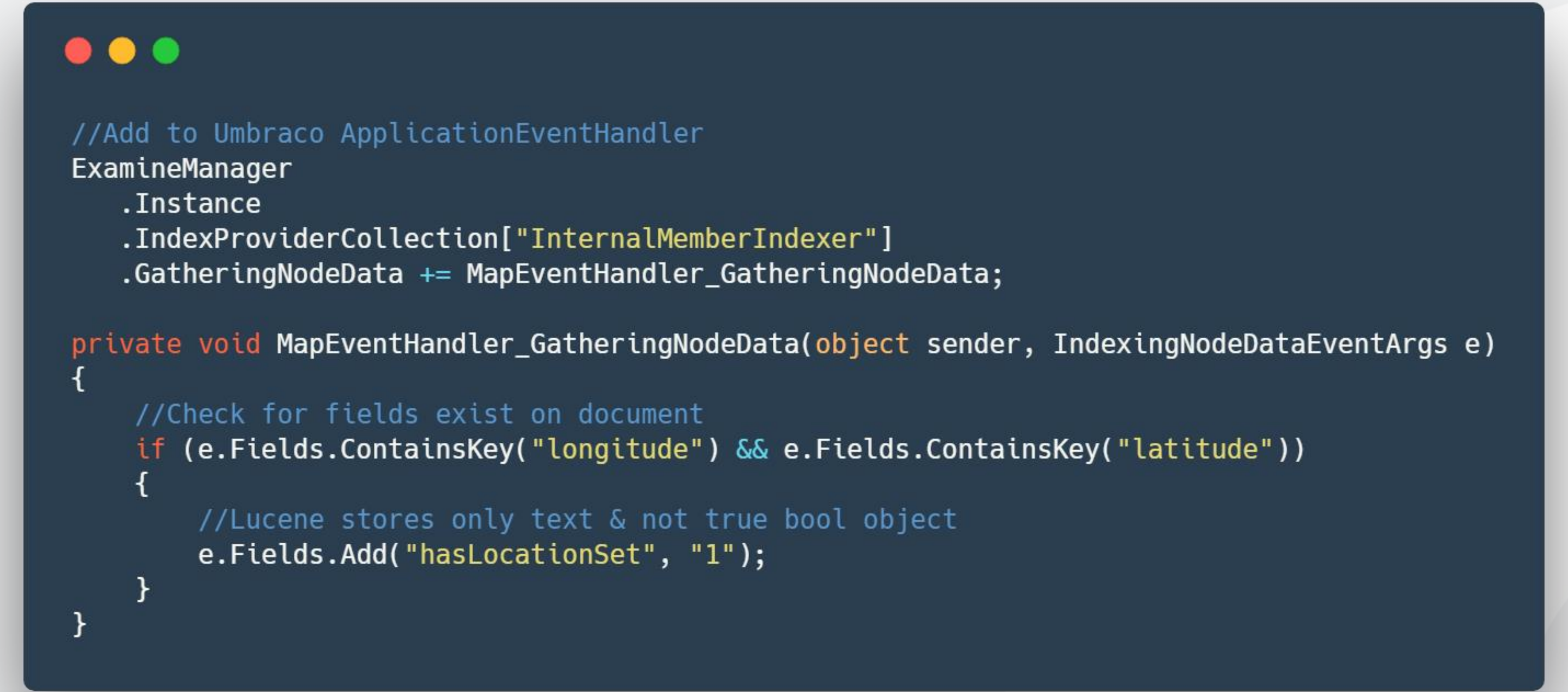
It’s not possible to search for members who have both Latitude & Longitude fields exist on the document.

Solution

Use Examine’s GatheringNodeData event to modify the Lucene document to dynamically add a new field that we can search for.

#CG18

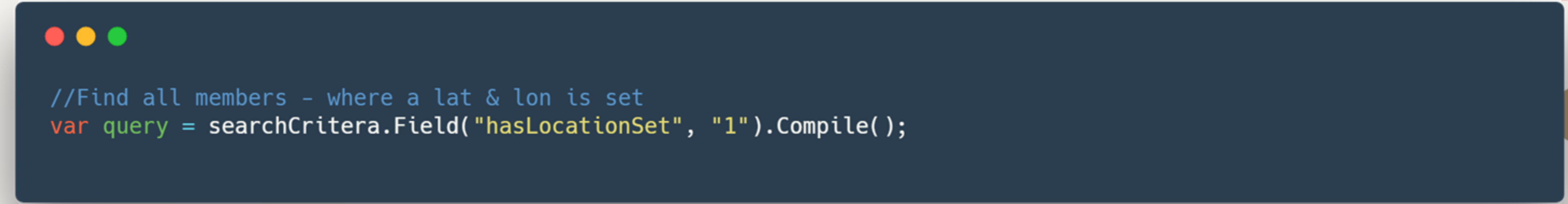
Gathering Node Data - Example



```
//Add to Umbraco ApplicationEventHandler
ExamineManager
    .Instance
    .IndexProviderCollection["InternalMemberIndexer"]
    .GatheringNodeData += MapEventHandler_GatheringNodeData;

private void MapEventHandler_GatheringNodeData(object sender, IndexingNodeDataEventArgs e)
{
    //Check for fields exist on document
    if (e.Fields.ContainsKey("longitude") && e.Fields.ContainsKey("latitude"))
    {
        //Lucene stores only text & not true bool object
        e.Fields.Add("hasLocationSet", "1");
    }
}
```

Query

#CG18



```
//Find all members - where a lat & lon is set
var query = searchCritera.Field("hasLocationSet", "1").Compile();
```

#CG18

Too much data !

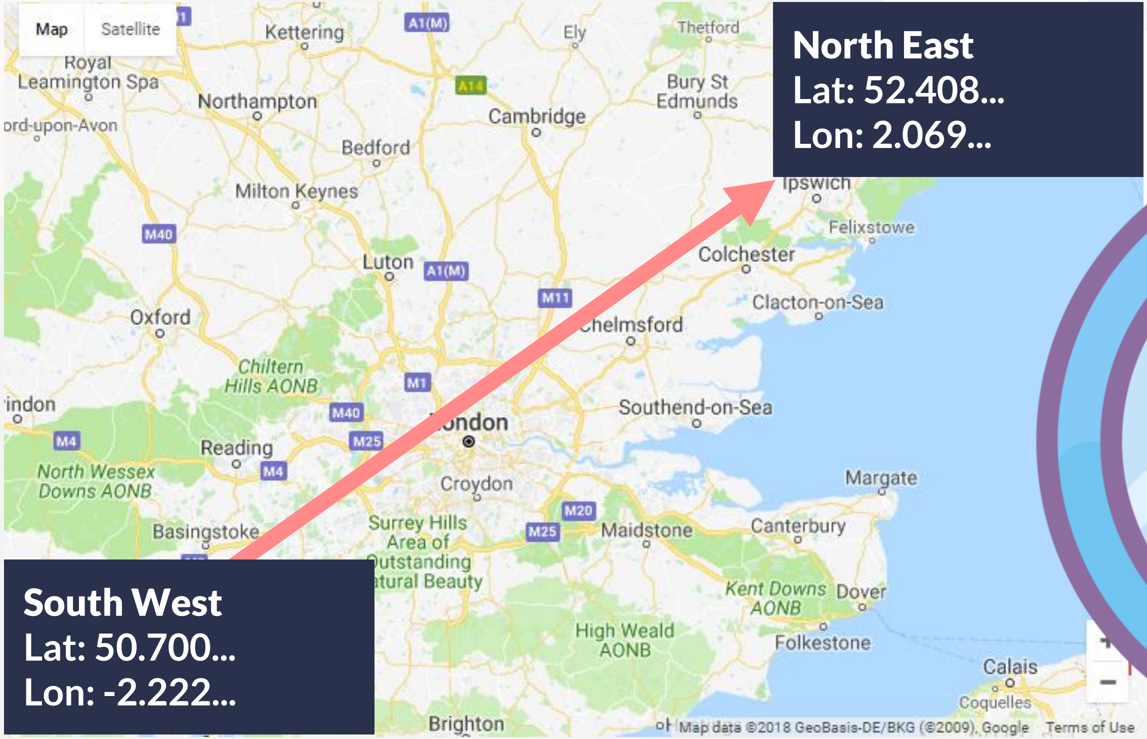
Problem

Querying for all members will give us a GIANT JSON payload, thus adding all these markers/pins to the Google Map will be slooooow

Solution

Search for members who are only visible on the current map view, by searching members lat & lon within a range & get only the data we need

Finding members on the map

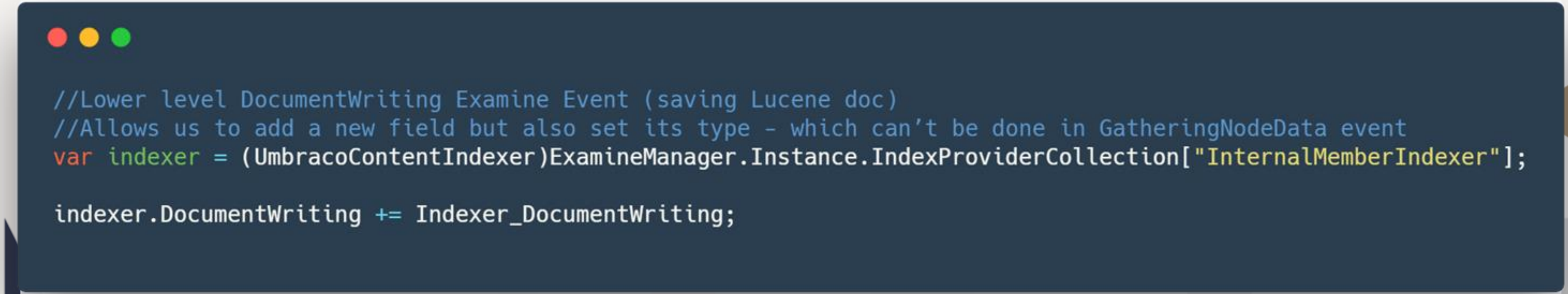


North East
Lat: 52.408...
Lon: 2.069...

South West
Lat: 50.700...
Lon: -2.222...

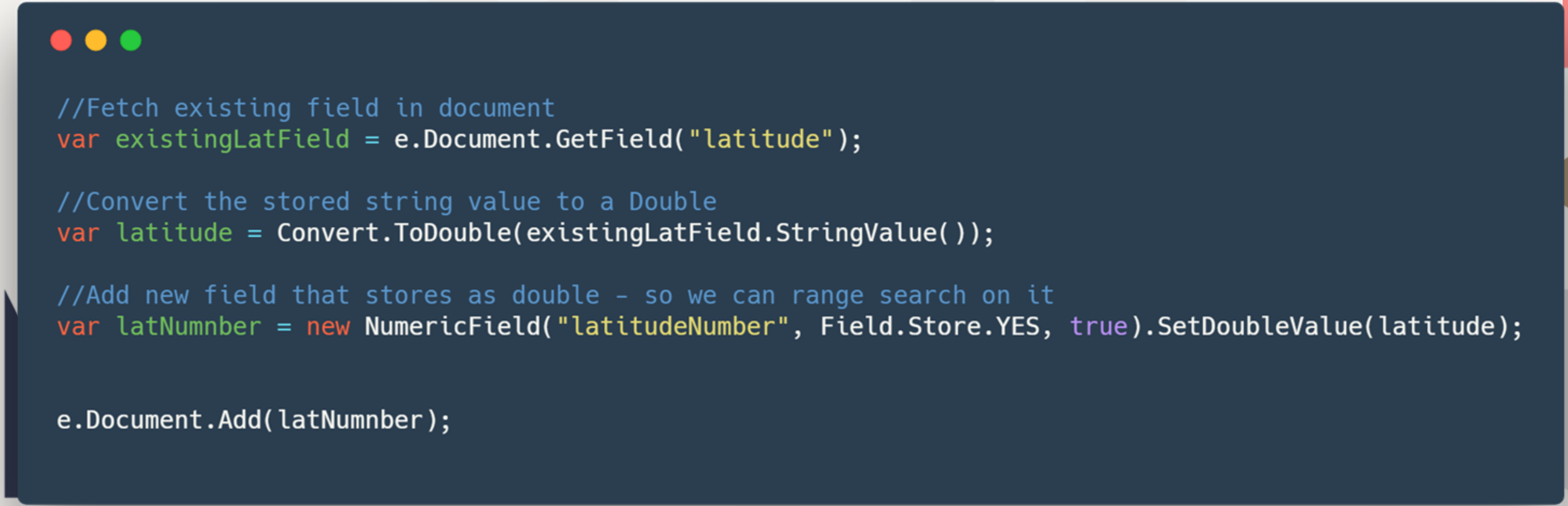
Find members in range of **South West** to **North East**

Store fields as doubles



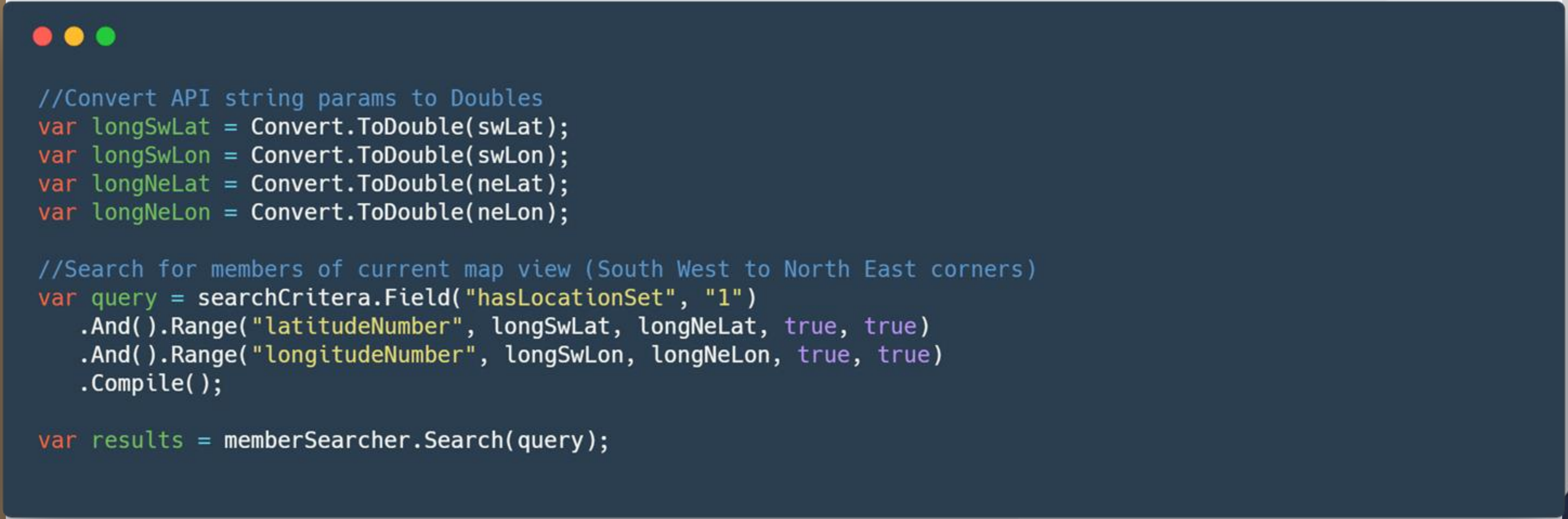
```
//Lower level DocumentWriting Examine Event (saving Lucene doc)
//Allows us to add a new field but also set its type - which can’t be done in GatheringNodeData event
var indexer = (UmbracoContentIndexer)ExamineManager.Instance.IndexProviderCollection["InternalMemberIndexer"];

indexer.DocumentWriting += Indexer_DocumentWriting;
```

Store fields as doubles



```
//Fetch existing field in document
var existingLatField = e.Document.GetField("latitude");

//Convert the stored string value to a Double
var latitude = Convert.ToDouble(existingLatField.StringValue());

//Add new field that stores as double - so we can range search on it
var latNumnber = new NumericField("latitudeNumber", Field.Store.YES, true).SetDoubleValue(latitude);


e.Document.Add(latNumnber);
```

Range Search Query using Doubles



```
//Convert API string params to Doubles
var longSwLat = Convert.ToDouble(swLat);
var longSwLon = Convert.ToDouble(swLon);
var longNeLat = Convert.ToDouble(neLat);
var longNeLon = Convert.ToDouble(neLon);

//Search for members of current map view (South West to North East corners)
var query = searchCritera.Field("hasLocationSet", "1")
    .And().Range("latitudeNumber", longSwLat, longNeLat, true, true)
    .And().Range("longitudeNumber", longSwLon, longNeLon, true, true)
    .Compile();

var results = memberSearcher.Search(query);
```

#CG18

Spam overload

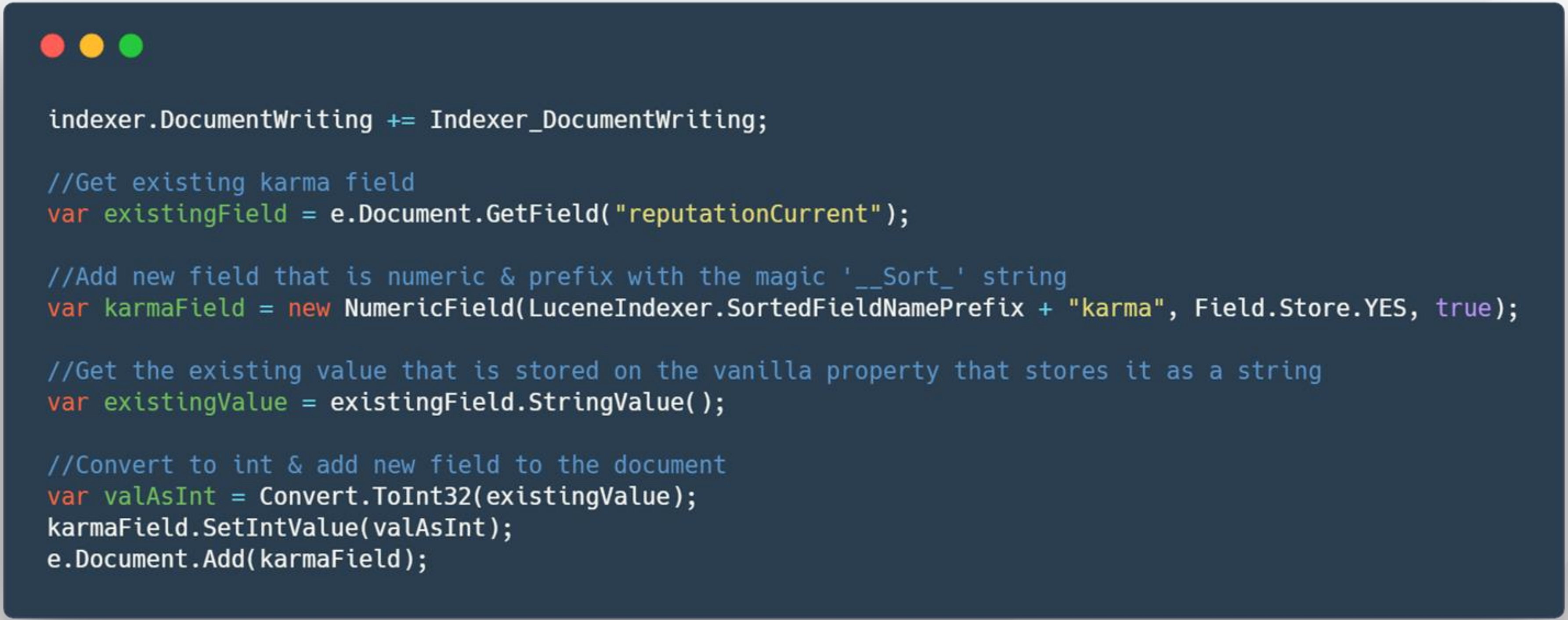
Problem

After querying for members we still have loads of spam accounts or accounts that never post, skewing data on the map

Solution

Filter members by Karma; a point reward system on our.umbraco who have greater than 70 points and order API results by highest points

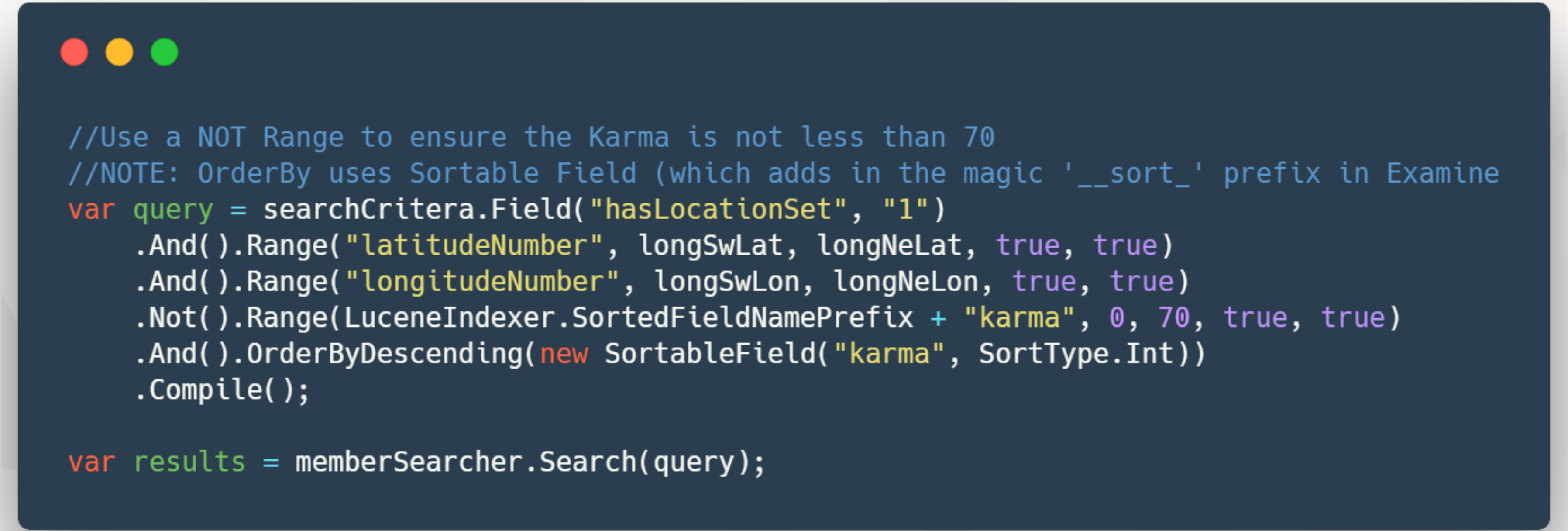
Add sortable field



```
indexer.DocumentWriting += Indexer_DocumentWriting;

//Get existing karma field
var existingField = e.Document.GetField("reputationCurrent");

//Add new field that is numeric & prefix with the magic '__Sort_' string
var karmaField = new NumericField(LuceneIndexer.SortedFieldNamePrefix + "karma", Field.Store.YES, true);

//Get the existing value that is stored on the vanilla property that stores it as a string
var existingValue = existingField.StringValue();

//Convert to int & add new field to the document
var valAsInt = Convert.ToInt32(existingValue);
karmaField.SetIntValue(valAsInt);
e.Document.Add(karmaField);
```

Query

#CG18



```
//Use a NOT Range to ensure the Karma is not less than 70
//NOTE: OrderBy uses Sortable Field (which adds in the magic '__sort_' prefix in Examine
var query = searchCritera.Field("hasLocationSet", "1")
    .And().Range("latitudeNumber", longSwLat, longNeLat, true, true)
    .And().Range("longitudeNumber", longSwLon, longNeLon, true, true)
    .Not().Range(LuceneIndexer.SortedFieldNamePrefix + "karma", 0, 70, true, true)
    .And().OrderByDescending(new SortableField("karma", SortType.Int))
    .Compile();

var results = memberSearcher.Search(query);
```

Plant an idea

- Explore the possibilities of using Examine & Lucene for something more powerful than showing search results
- Open your mind there is always something new to learn
- Get on the courses & be inspired

#CG18

Thank you

@warrenbuckley

#CG18

Questions?