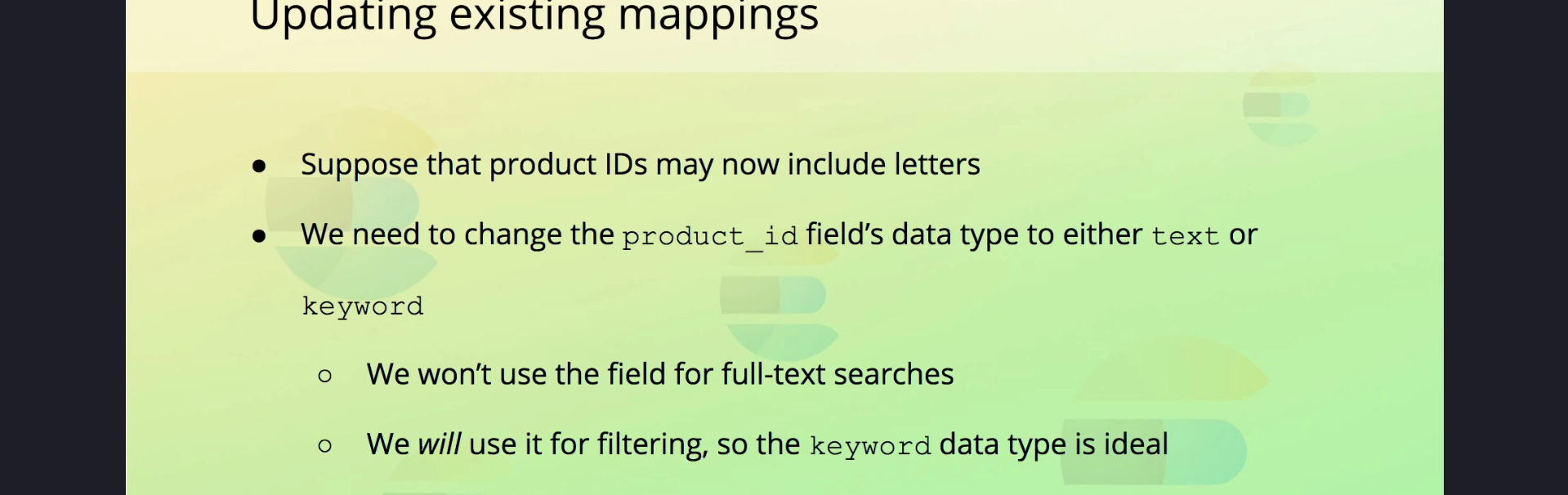
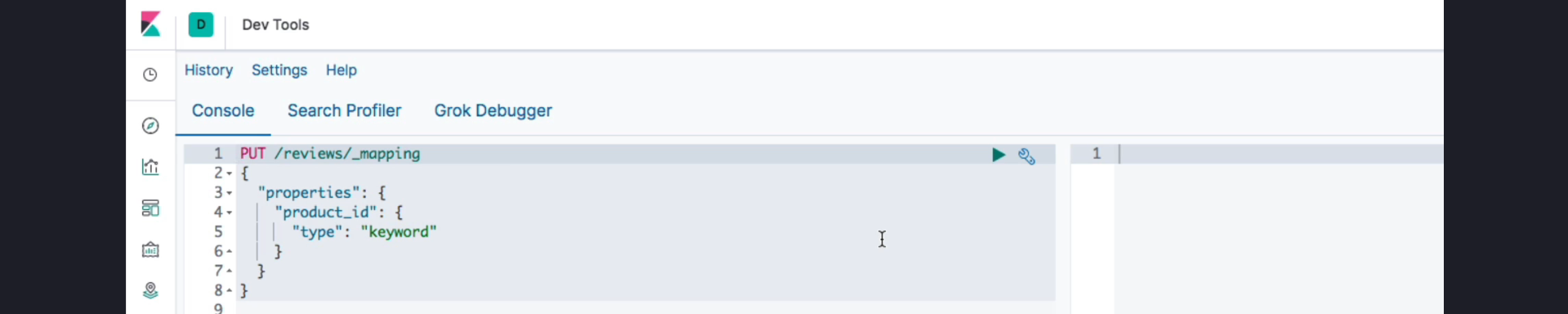
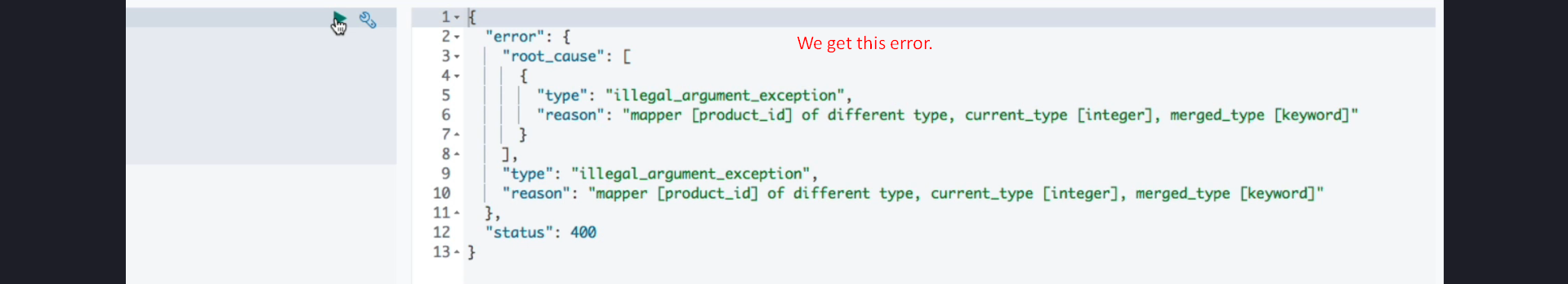
# 

# Updating Field Mappings in Elasticsearch

A bit earlier you saw how to add new field mappings to an existing index. But what about updating an existing field mapping? The “reviews” index currently contains a “product\_id” field of the type “long.” That data type has served us well, but for whatever reason our product IDs may now contain letters, so we need to change it. We won’t be performing full-text searches on the field, so the “text” data type doesn’t make sense. Instead, we will use the field for filtering, i.e., to find the reviews for a given product. That narrows the data type down to “keyword,” which is the one we will use.

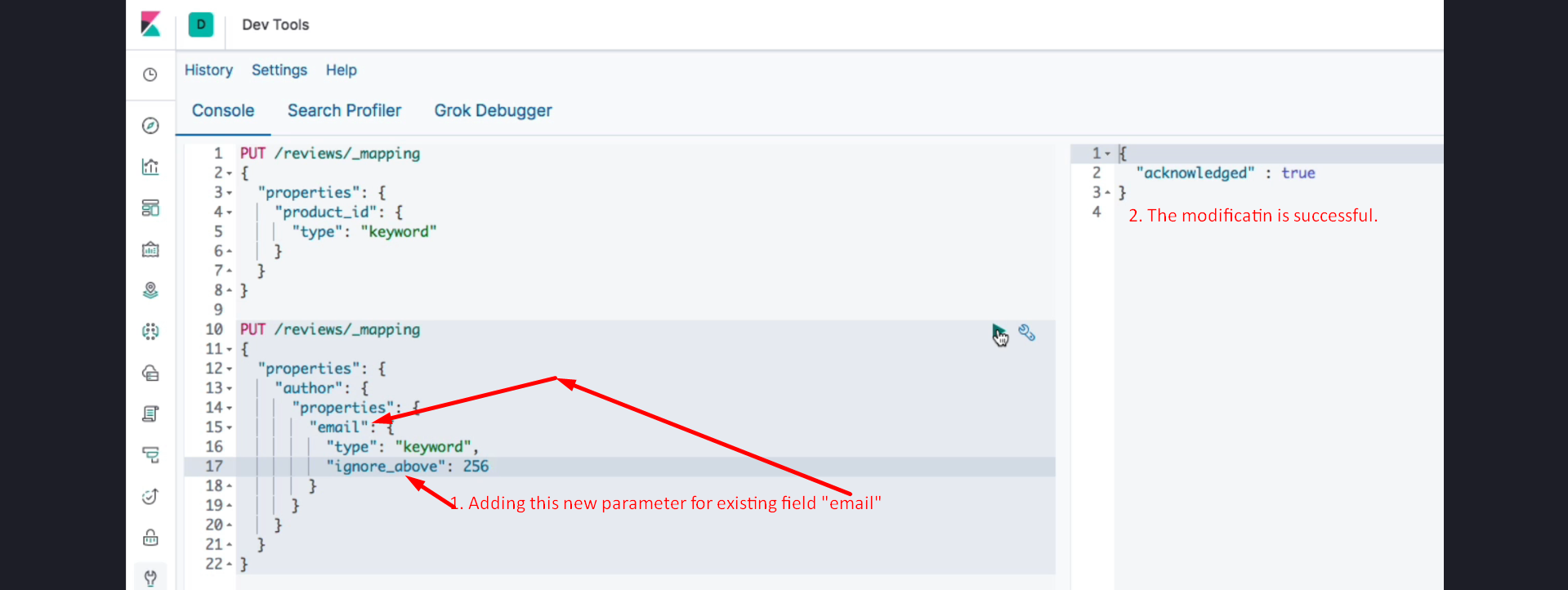


## Attempting to Change Field Mapping

So how do we actually change the field’s mapping? Can we just use the same query that we used to add a new field mapping to an existing index? I have it typed out already, so let’s try.  
  
  


As you can see, we get an error stating that the provided data type is different from the existing one. The reason is that, in general, we cannot update field mappings in Elasticsearch.

## Why Updating Field Mappings Is Restricted

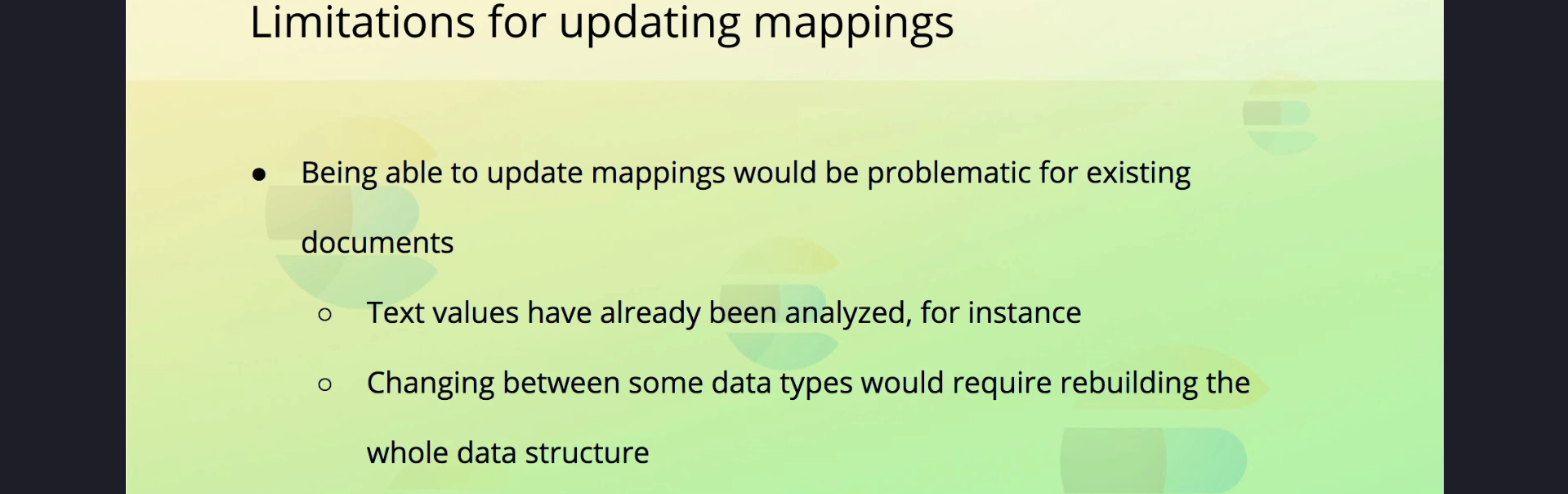
We can add new field mappings as we did earlier, but once you have added a field mapping, you are pretty much stuck with it. There is one exception, though; some **mapping parameters** can be updated, but only a few of them. Let’s see an example.  
  


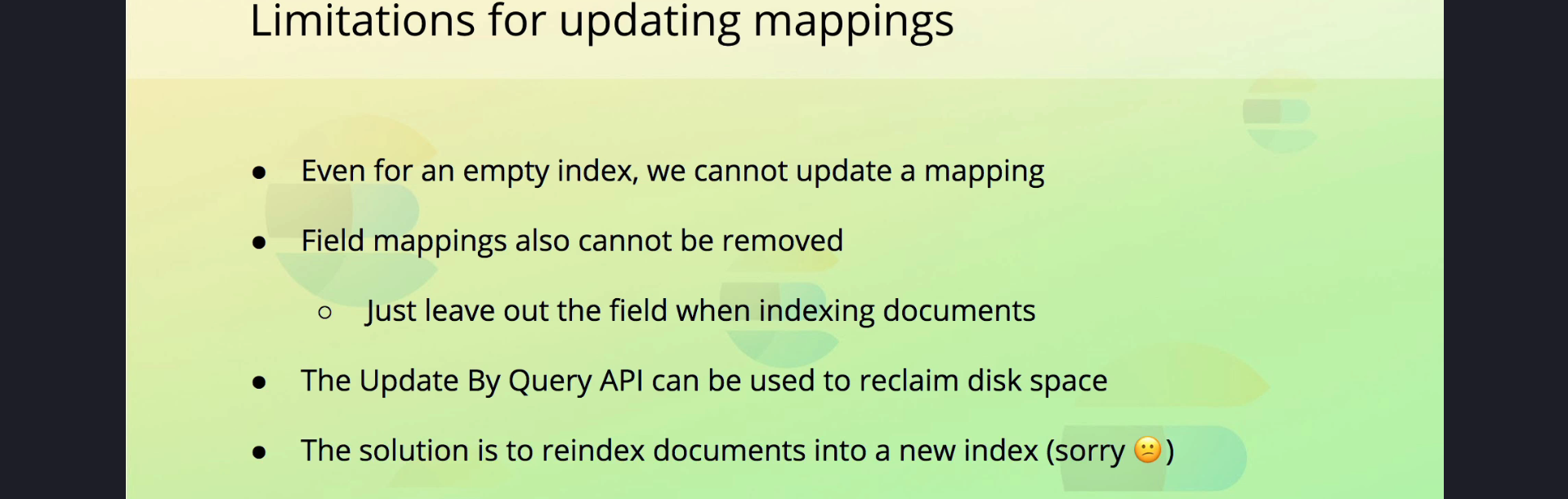
Consider the second query on your screen above. It contains the current mapping for the “email” field that is nested within the “author” field. I will add one of the parameters that can be added to existing field mappings; namely, the “ignore\_above” parameter. This parameter ignores strings longer than the specified value such that they will not be indexed or stored.

As you can see, we were indeed able to modify the field mapping to include this parameter. Chances are, though, that you will not be able to update a field mapping once you have created it.

## Challenges in Changing Field Mappings

There are a number of reasons for that, with one of the main ones being that documents might already have been indexed with the existing field mapping. If we could just change the mapping, we would have to reindex any existing documents. For instance, changing a “text” field to a “keyword” field is not possible because values have already been analyzed.

Our failed attempt to change the “long” data type to “keyword” also didn’t work. Remember how numeric values are indexed into a BKD tree. Switching to the “keyword” data type would require that data structure to be converted to an inverted index, which is essentially the same as reindexing all of the field’s values. If you have indexed many millions of documents, that is going to take a while.  
  




## What to Do Instead

So if we cannot update the data type for our “product\_id” field, then what do we do? We create a new index with the updated mapping and reindex any existing documents into it.

Let’s see how to do that in the next lecture.