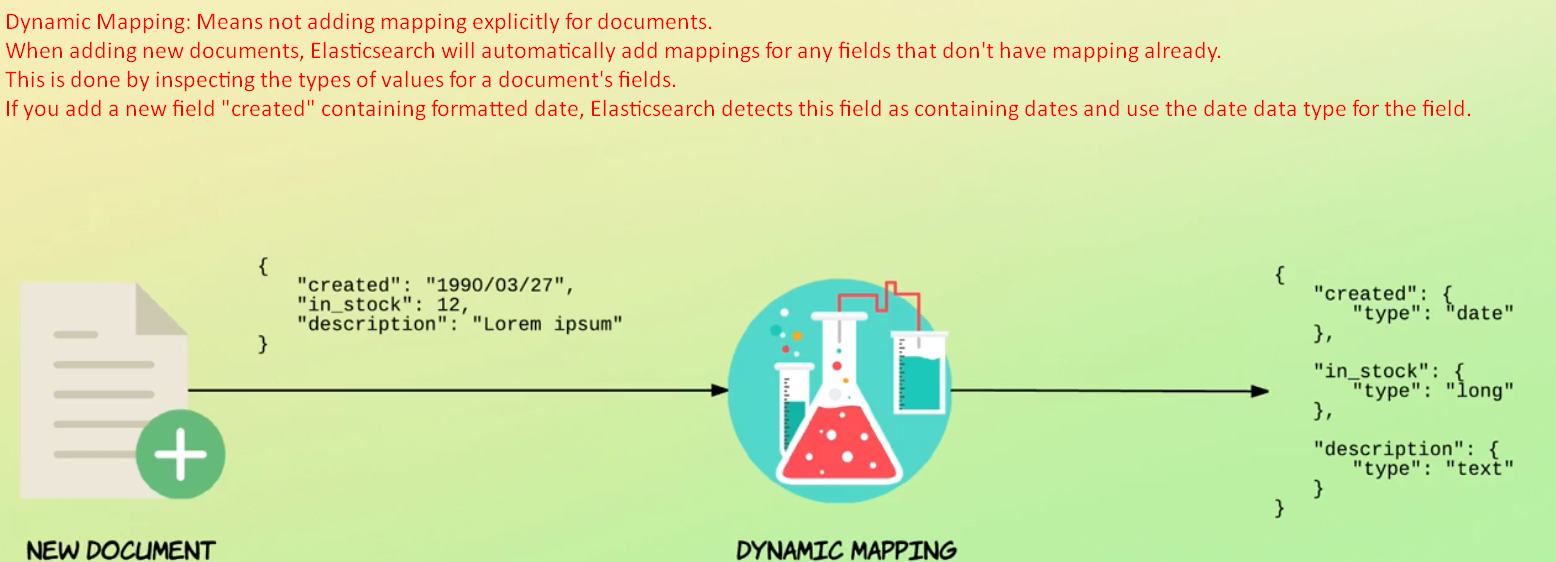
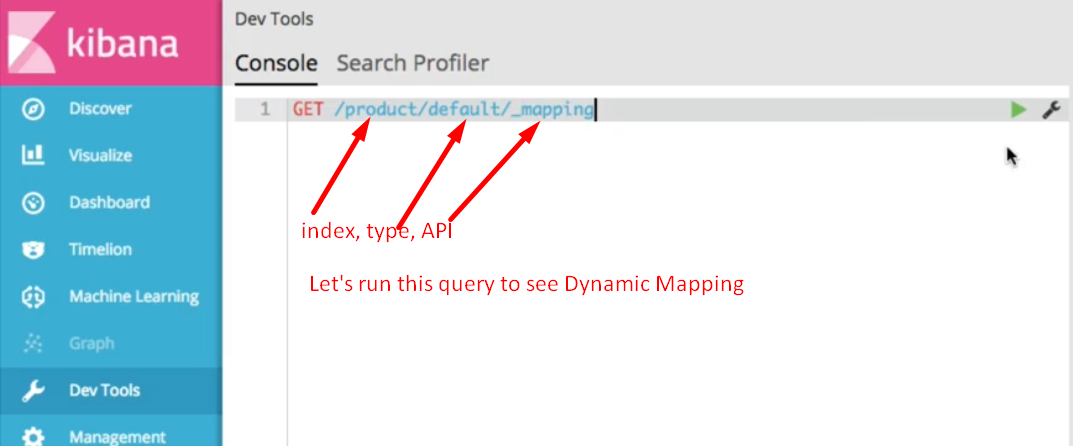
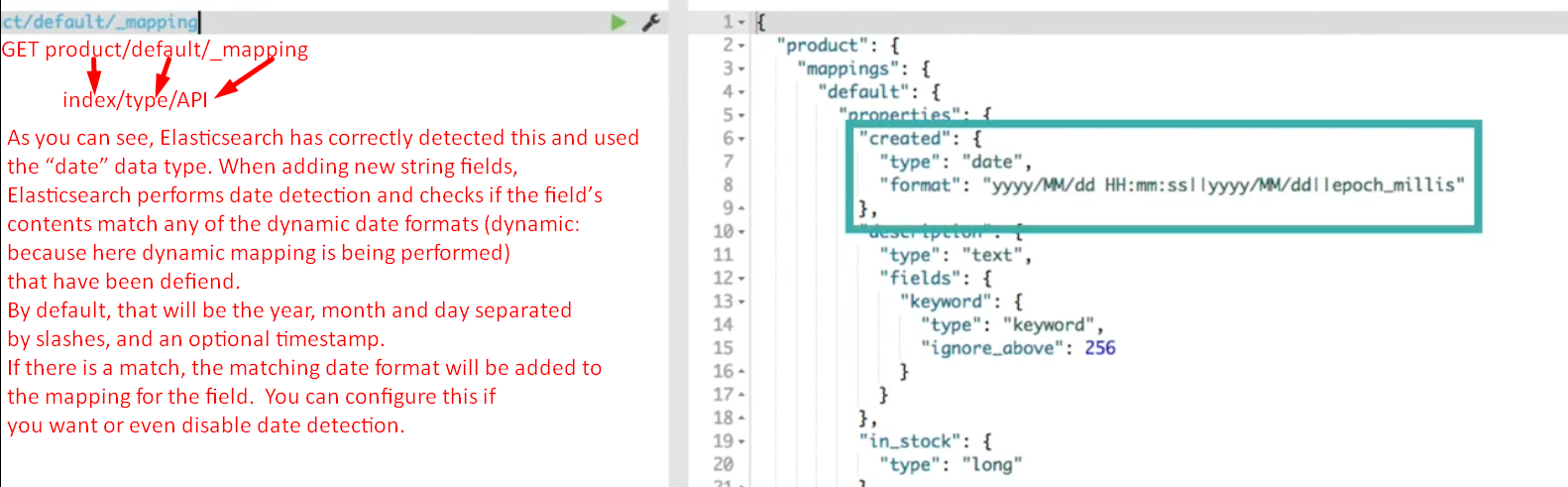
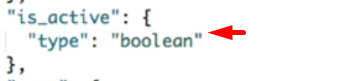
1. 
2. So, mapping can be defined explicitly which is referred to as **Explicit Mapping.**
3. That is when we tell Elasticsearch what to do with our data when we add new documents or update existing ones.  
   Or, as you know, documents are immutable, so updating existing documents will technically add them again.
4. Anyways, there is a convenient alternative to explicitly defining the data types for all fields; something called **Dynamic mapping.**
5. Dynamic mapping means that no mapping is defined explicitly, or at least not for some fields.  
   
6. By default Dynamic Mapping is enabled and working for all the documents to be indexed.
7. It is possible to define rules and defaults for dynamic mapping, but this is usually not something you will need to do, so I will not get into that now.  
   Instead, I want to show you which mappings have been added for us automatically through Dynamic Mapping.
8. To do that, we just need to run a simple query in Kibana.
9. 
10. **GET product/default/\_mapping **
11. Elasticsearch uses the “long data type for integers by default, because it has no way of knowing how large you number intend to store.  
    If it had used a “short” because the first document had a small value, and a document with a large number was added later, than would be a problem.



1. See, the “**is\_active**” field has been mapped to a Boolean field because it contains Boolean values.



1. One thing that is a bit special, though, is how text fields are treated.  
   For example, notice how the “description” field is mapped as the type “text” which is what you would expect. But also notice that it has a property named “fields” which contains a property named “keyword” which contains key value 🡺 “type”:”keyword”. It means this field has two mappings.  
   We will get back to the details of this, but the point is that a field may have multiple mappings.  
   By default, each text field is mapped using both the “text” type and the “keyword” type.  
   The difference b/w the two is that the **“text” type is used for full-text searches** and the “**keyword**” type for exact matches, aggregations and such.  
   To facilitate both use cases, Elasticsearch automatically maps each text field in both ways.  
   This is very convenient, because then you will not be surprised down the road that you can’t run a given query as you would expect.  
   We will get back to all of this in the next couple of lectures, so don’t worry too much.
2. So to sum up:
   1. Elasticsearch has mapped the fields for us automatically based on their values.