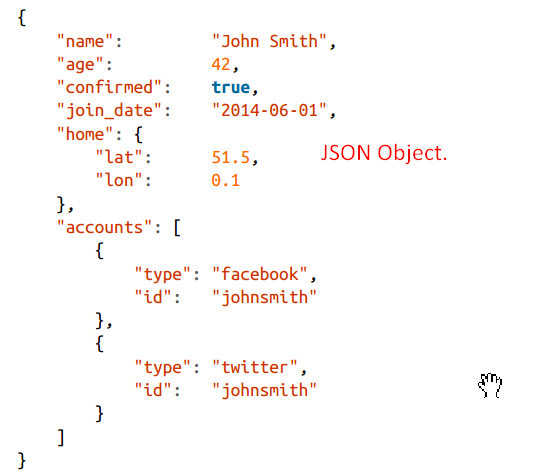
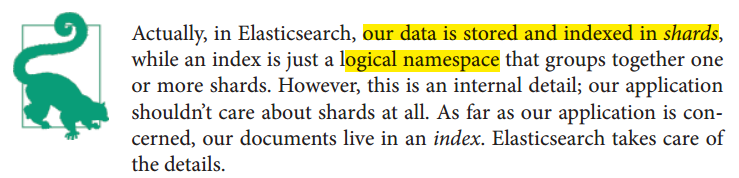
1. **JSON Document**: When an object is serialized into JSON, it’s known as a **JSON Document**.
2. Elasticsearch is a **distributed document store**.
3. Document can be retrieved from any node in the cluster.
4. Which fields require **an index** in order to make data retrieval fast.
5. In Elasticsearch, all data in every field is **indexed by default**. That is, every field has a dedicated **inverted index** for fast retrieval.

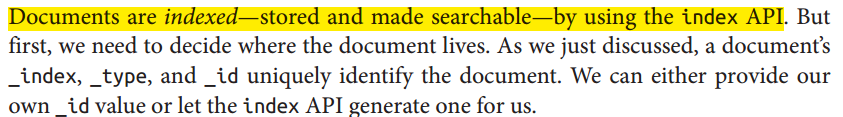
What is a Document?

1. Most entities or objects in most applications can be serialized into a JSON object with keys and values.   
   
2. Object and document interchangeably.
   1. **Object**: JSON Object similar to what is known as a hash, hashmap, dictionary or associative array.
   2. Document: Special meaning in Elasticsearch. It refers to the top-level, or root object that is serialized into JSON and stored in Elasticsearch under a unique ID.

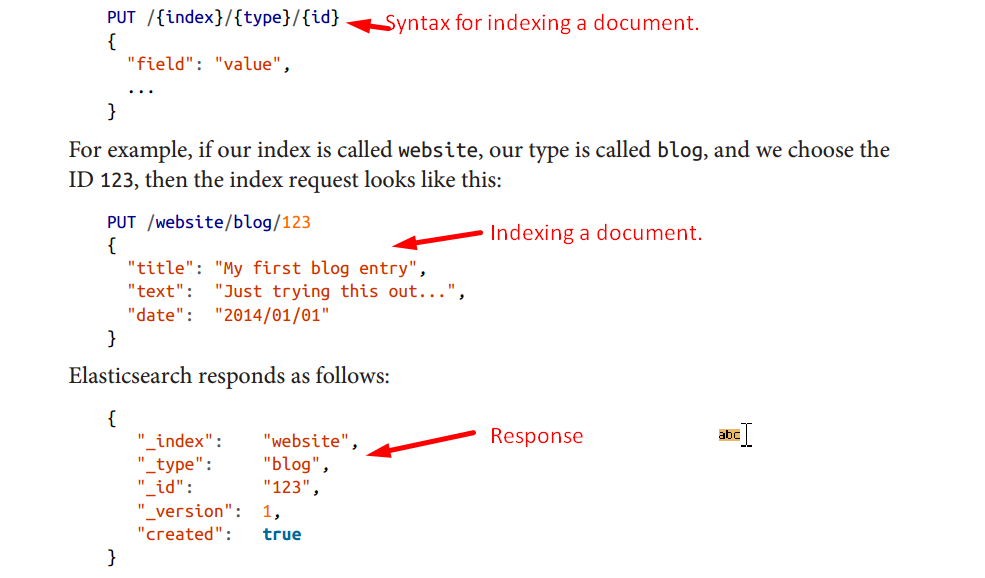
Document Metadata

1. A document also contains metadata about the document (actual data).
   1. **\_index**: Where the document lives
   2. **\_type**: The class of object that the document represents.
   3. **\_id:** The unique identifier for the document.
2. **\_index**:
   1. Like a database where we store and index related data.
   2. 
   3. Rules for index.
      1. Lowercase.
      2. Can’t begin with underscore.
      3. Can’t contain commas.
3. \_**type**:
   1. A type in Elasticsearch is similar like a class in java.
   2. Every type has its own mapping or schema definition, which defines the data structure for documents of that type.
   3. Documents of all types can be stored in the same index but the mapping for the type tells Elasticsearch how the data in each document should be indexed.
   4. For the time being, we will rely on Elasticsearch to detect our document’s data structure (type) automatically.
   5. **Rules**:
      1. In lowercase or uppercase.
      2. Should not being with underscore.
      3. Must not contain commas
4. \_**id**:
   1. The ID is a string that, when combined with \_index, \_type, uniquely identifies a document in Elasticsearch.
   2. When creating a new document, you can either provide your own \_id or let Elasticsearch generate one for you.

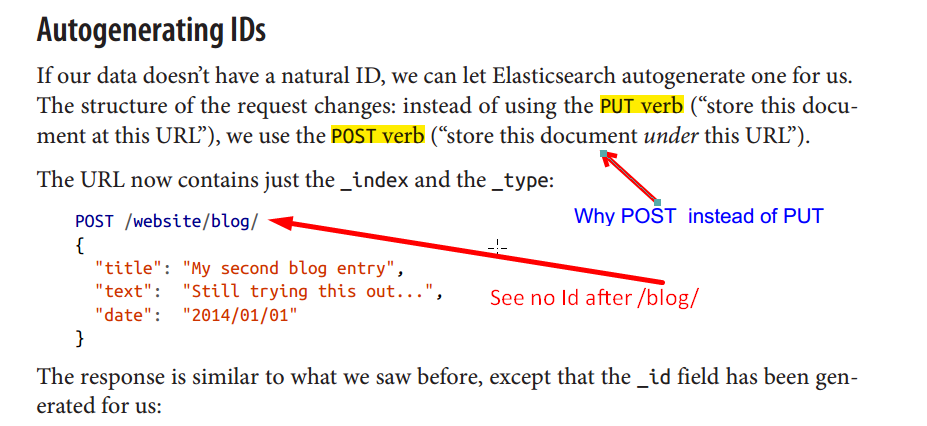
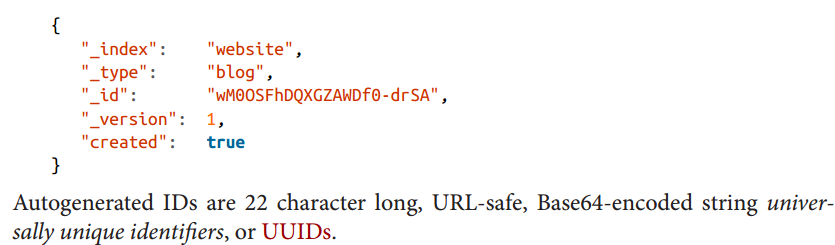
Indexing a Document

1. 

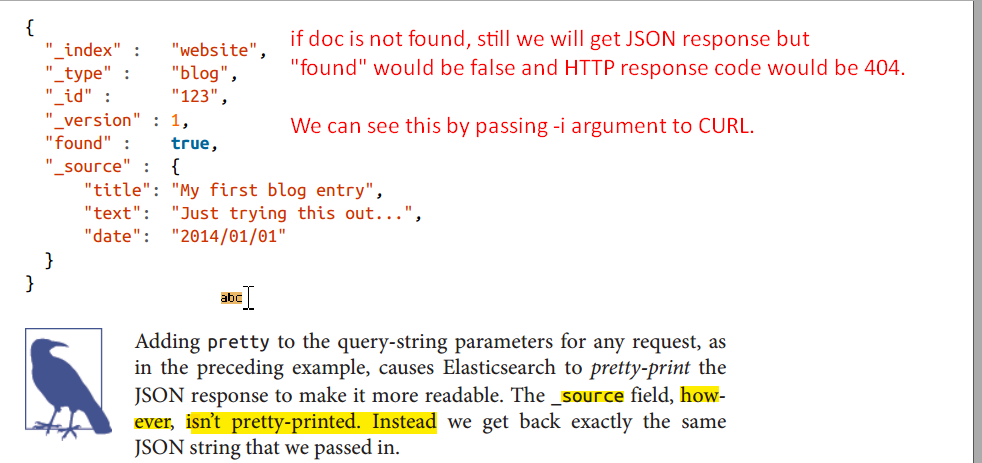
Using Our Own ID

1. If your document has a natural identifier that identifies the document, you should provide your own \_id, using this form of the index API:
2. 
3. \_version: Every time a change is made to a document (including deleting it), the \_version number is incremented.

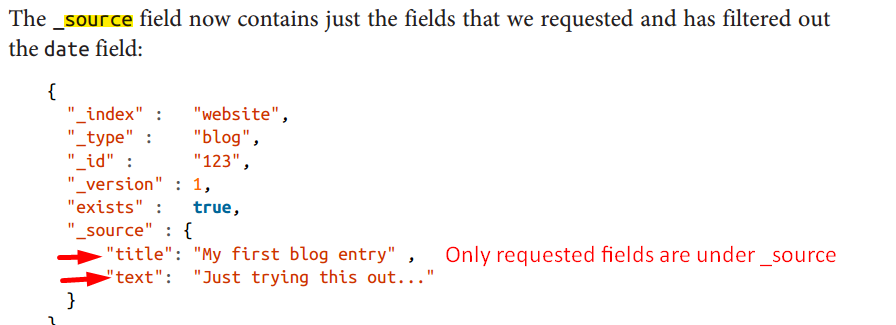
Autogenerating IDs

1. 
2. 

Retrieving a Document

1. To get the document out of Elasticsearch, we use the same **\_index, \_type, \_id** but the HTTP verb changes to **GET.**
2. 
3. 
4. 

Retrieving Part of a Document

1. By Default, a GET request will return the whole document, as stored in the **\_source** field.
2. **Agenda**: But we’re interested only in **title field**.
3. **How**:
   1. Individual fields can be requested by using the **\_source parameter**.
   2. Multiple fields can be specified in a **comma-separated list**.  
      
4. 
5. How to get just \_source field without any metadata.
6. 