# Using Dot Notation for Object Mapping in Elasticsearch

## 1. Introduction

In Elasticsearch, objects are mapped implicitly using the 'properties' mapping parameter at each level of the hierarchy. This approach can make the mapping verbose, especially for deeply nested objects.

To simplify the process, Elasticsearch provides an alternative approach by using dot notation. This makes mapping definitions more concise and readable.

## 2. Traditional Object Mapping

When defining mappings for objects, each level of the hierarchy is specified explicitly using the 'properties' parameter. For example, the 'author' field in the 'reviews' index was mapped as follows:

PUT /reviews  
{  
 "mappings": {  
 "properties": {  
 "author": {  
 "properties": {  
 "first\_name": { "type": "text" },  
 "last\_name": { "type": "text" },  
 "email": { "type": "keyword" }  
 }  
 }  
 }  
 }  
}

This approach works well for simple mappings but can become cumbersome for more complex, deeply nested objects.

## 3. Dot Notation for Object Mapping

Dot notation provides a simpler way to define mappings for objects. Using this approach, each level of the hierarchy is separated by a dot. For example:

PUT /reviews\_dot\_notation  
{  
 "mappings": {  
 "properties": {  
 "author.first\_name": { "type": "text" },  
 "author.last\_name": { "type": "text" },  
 "author.email": { "type": "keyword" }  
 }  
 }  
}

This format is more compact and easier to read, especially when dealing with deeply nested objects.

## 4. Internal Translation by Elasticsearch

When dot notation is used, Elasticsearch automatically translates the mapping into the traditional 'properties' structure. As a result, retrieving the mapping for an index will always show the explicit 'properties' format, regardless of how the mapping was defined.

For example, retrieving the mapping for the 'reviews\_dot\_notation' index would yield the following result:

GET /reviews\_dot\_notation/\_mapping

{  
 "mappings": {  
 "properties": {  
 "author": {  
 "properties": {  
 "first\_name": { "type": "text" },  
 "last\_name": { "type": "text" },  
 "email": { "type": "keyword" }  
 }  
 }  
 }  
 }  
}

This demonstrates that dot notation is simply a shortcut for defining mappings, and it does not affect the underlying structure.

## 5. Using Dot Notation in Search Queries

Dot notation is not limited to defining mappings; it can also be used within search queries to access nested fields.

For example, to search for documents where the 'author.email' field equals 'john.doe@example.com':

GET /reviews/\_search  
{  
 "query": {  
 "term": { "author.email": "john.doe@example.com" }  
 }  
}

## 6. Summary

1. Dot notation simplifies the definition of object mappings by allowing hierarchical fields to be specified in a concise format.  
2. Elasticsearch translates dot notation into the traditional 'properties' structure internally, so the resulting mapping is unchanged.  
3. Dot notation can also be used in search queries to access nested fields.  
4. While not mandatory, dot notation is a handy shortcut for creating and querying nested mappings.

