# 

# Adding Mappings to an Existing Index in Elasticsearch

## 1. Introduction

In Elasticsearch, mappings define the structure and data types of fields within an index. Previously, we discussed defining mappings for a new index. In this lecture, we focus on adding mappings to an existing index.

For instance, we might realize that a 'reviews' index should also store a timestamp for when a review was written. Since Elasticsearch does not store timestamps for indexed documents by default, we need to explicitly add a 'created\_at' field to the index.

## 2. Adding a Field to an Existing Index

To add a new field to an existing index, we use the **Mapping API**. The process involves sending a PUT request with the new field definition. Here’s how we can add a 'created\_at' field to the 'reviews' index:

PUT /reviews/\_mapping  
{  
 "properties": {  
 "created\_at": { "type": "date" }  
 }  
}

Key points to note:  
- Use the PUT HTTP verb.  
- Invoke the Mapping API by specifying the '\_mapping' endpoint.  
- Define the 'properties' key at the top level of the request body, without the 'mappings' key.

## 3. Verifying the Mapping

After adding the mapping, it is good practice to retrieve the updated mapping to ensure the changes have been applied successfully. To do this, use the following request:

GET /reviews/\_mapping

The results should show the newly added 'created\_at' field with its specified data type.

## 4. Summary

1. The Mapping API allows you to add new fields to an existing index in Elasticsearch.  
2. Use the PUT HTTP verb and specify the '\_mapping' endpoint.  
3. Define the 'properties' key at the top level of the request body to add new field mappings.  
4. Always verify the updated mapping to ensure the changes have been applied correctly.