

Introduction to the Bulk API

- You learned how to index, update, and delete documents
- Let's see how we can perform these actions on *many* documents with a *single* query
 - That's done with the Bulk API
- The Bulk API expects data formatted using the NDJSON specification

```
action_and_metadata\n
optional_source\n
action_and_metadata\n
optional_source\n
```

Things to be aware of (1/3)

- The HTTP `Content-Type` header should be set as follows
 - `Content-Type: application/x-ndjson`
 - `application/json` is accepted, but that's not the correct way
- The Console tool handles this for us
 - The Elasticsearch SDKs also handle this for us
 - Using HTTP clients, we need to handle this ourselves
- You will see how to do this in the next lecture

Things to be aware of (2/3)

- Each line **must** end with a newline character (`\n` or `\r\n`)
 - This **includes** the last line
 - In a text editor, this means that the last line should be empty
 - Automatically handled with the Console tool
 - Typically a script will generate the bulk file, in which case you need to handle this
 - Don't type out `\n` or `\r\n` in a text editor 😊

Things to be aware of (3/3)

- A failed action will **not** affect other actions
 - Neither will the bulk request as a whole be aborted
- The Bulk API returns detailed information about each action
 - Inspect the `items` key to see if a given action succeeded
 - The order is the same as the actions within the request
 - The `errors` key conveniently tells us if any errors occurred

When to use the Bulk API

- When you need to perform lots of write operations at the same time
 - E.g. when importing data or modifying lots of data
- The Bulk API is more efficient than sending individual write requests
 - A lot of network round trips are avoided

Two more things...

- Routing is used to resolve a document's shard
 - The routing can be customized if necessary
- The Bulk API supports optimistic concurrency control
 - Include the `if_primary_term` and `if_seq_no` parameters within the action metadata