**Opensearch indesxer** :- in new version we use opensearch indexer instead of wazuh indexer.

**Opensearch Dashboard** :- in new version we use opensearch dashboard instead of wazuh dashboard.

**OpenSearch** is an open-source search, analytics, and visualization platform designed to handle large volumes of data. It is a community-driven fork of the Elasticsearch and Kibana projects, created after Elastic transitioned these projects to a non-open-source license. OpenSearch provides a suite of tools for log analytics, full-text search, monitoring, and observability.

 **Cluster**:  
A collection of nodes working together to store and process data. The cluster ensures data distribution, fault tolerance, and high availability.

 **Node**:  
An instance of OpenSearch running in a machine (physical or virtual). Each node contributes to the cluster's capacity for storage and computation.

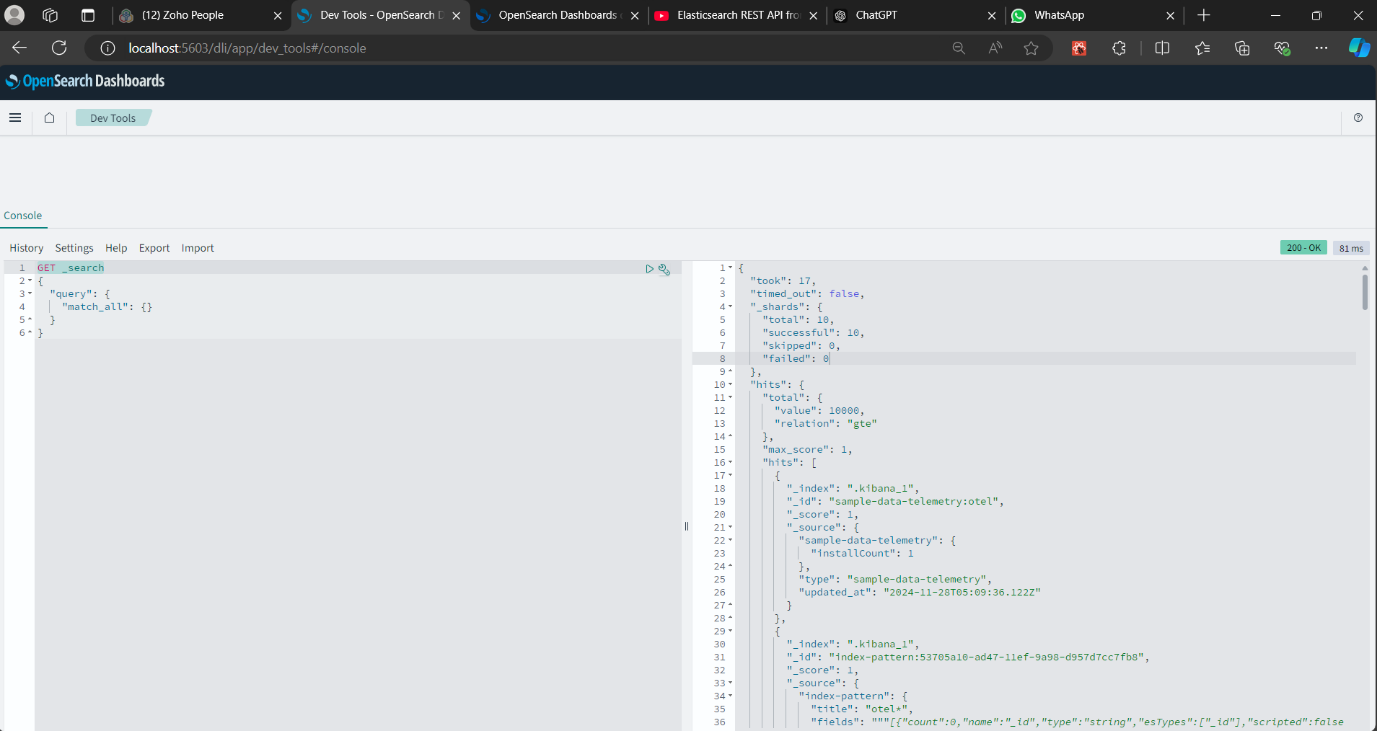
 **Index**:  
A logical structure similar to a table in SQL databases. It organizes documents, providing a way to group and query related data.

 **Field**:  
Similar to a column in a table, a field represents a single attribute of a document and stores specific data, like name, age, or price.

 **Document**:  
Similar to a row in a table, a document is a unit of data stored in JSON format. It contains fields (columns) with associated values and metadata.

**Dev tool**

1. GET \_search :- gives all data and all indexes

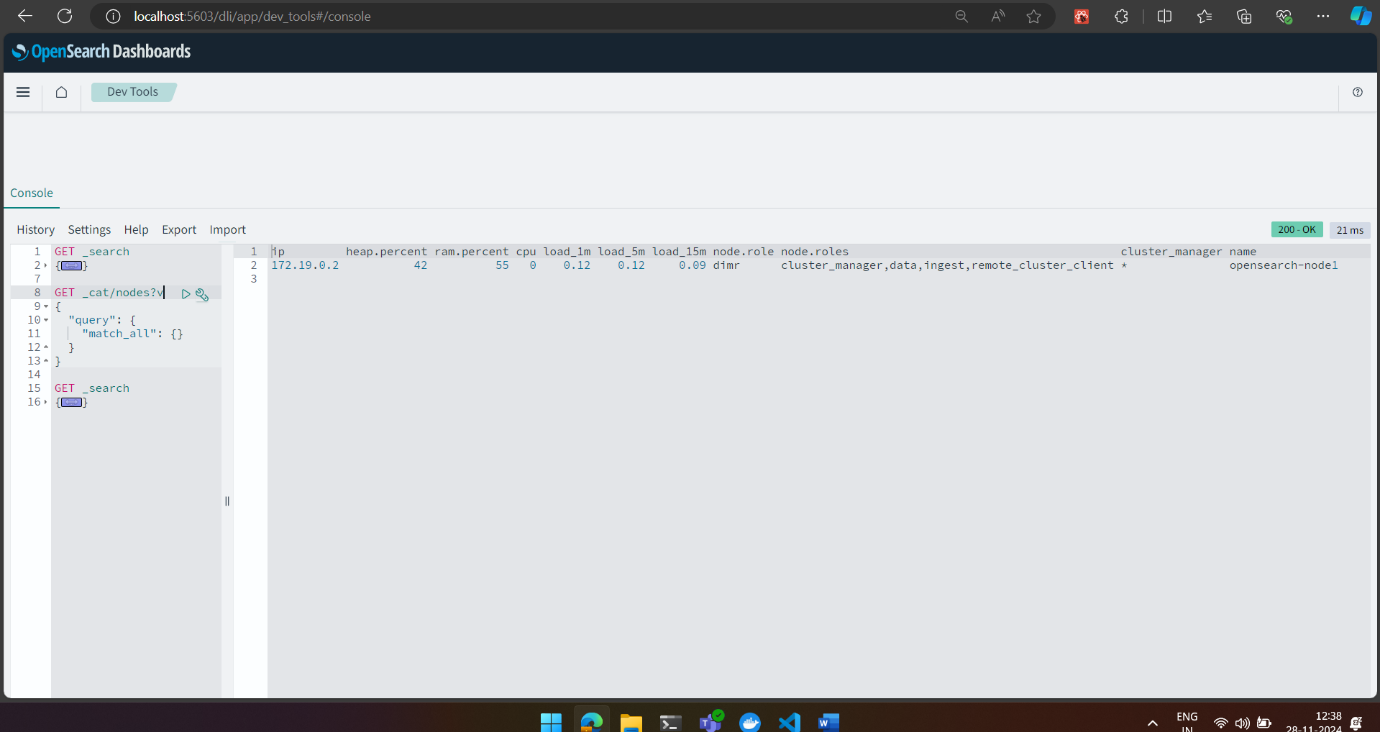


2.GET \_cat/nodes?v :-

cat = view data in human redable format

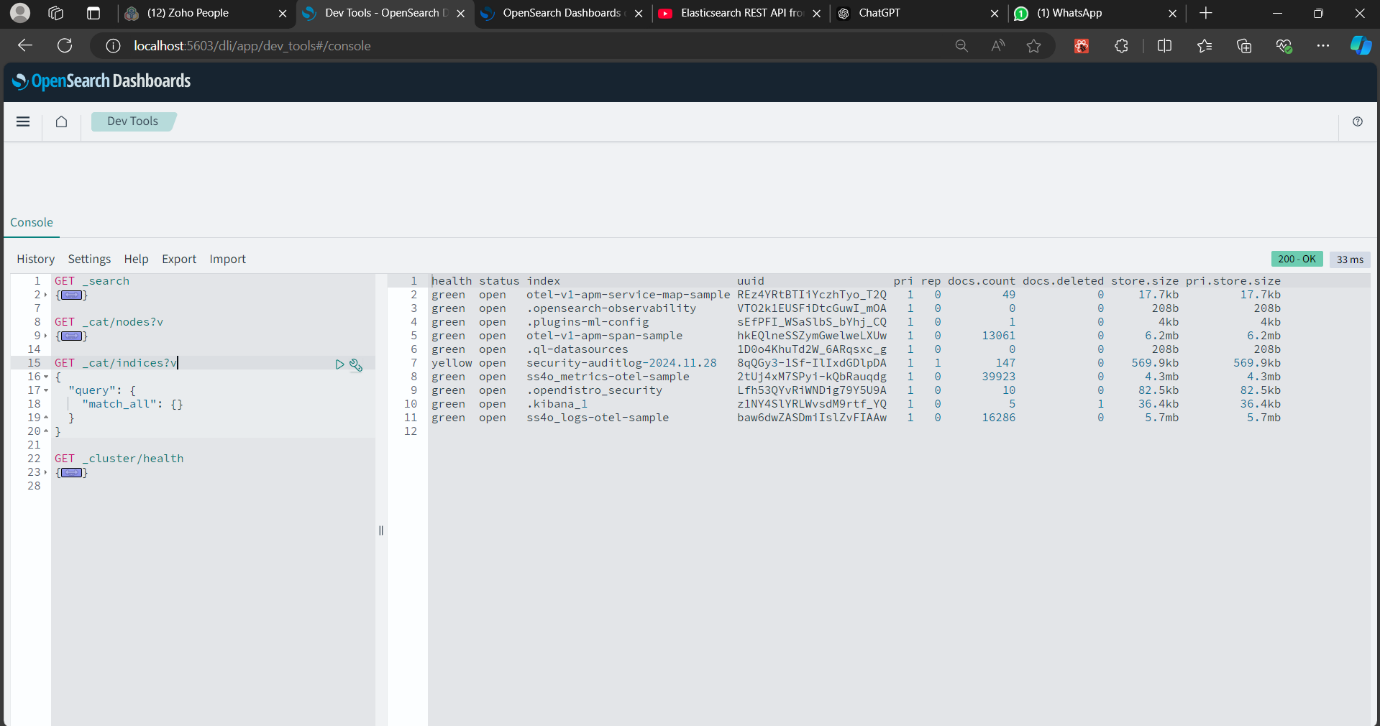
V = gives header

Node = number of nodes



3. \_cat/indices?v

Indices :- details of indices



**Create one employee index**

POST .kibana/\_doc/index-pattern-log-pattern

{

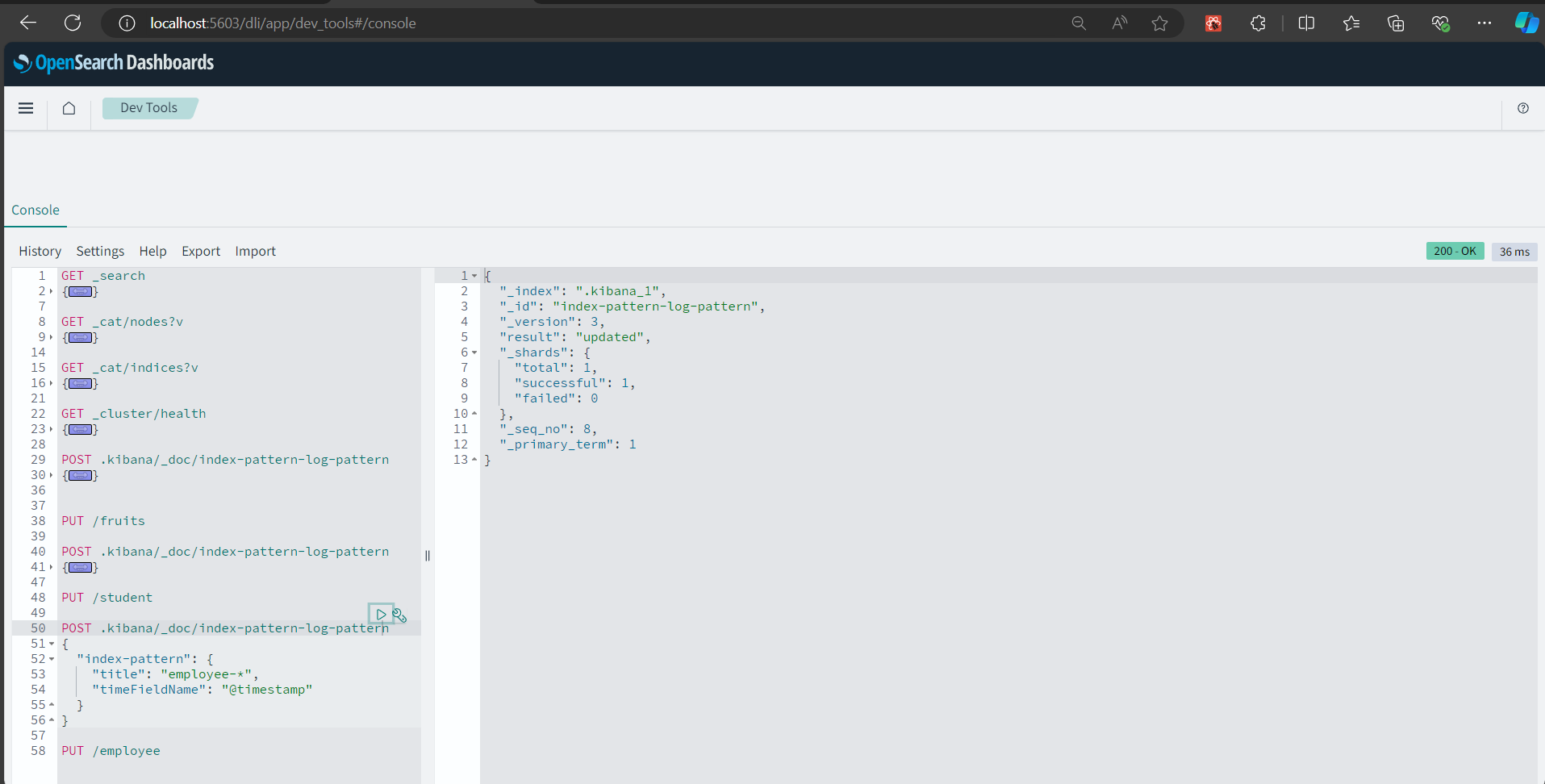
"index-pattern": {

"title": "employee-\*",

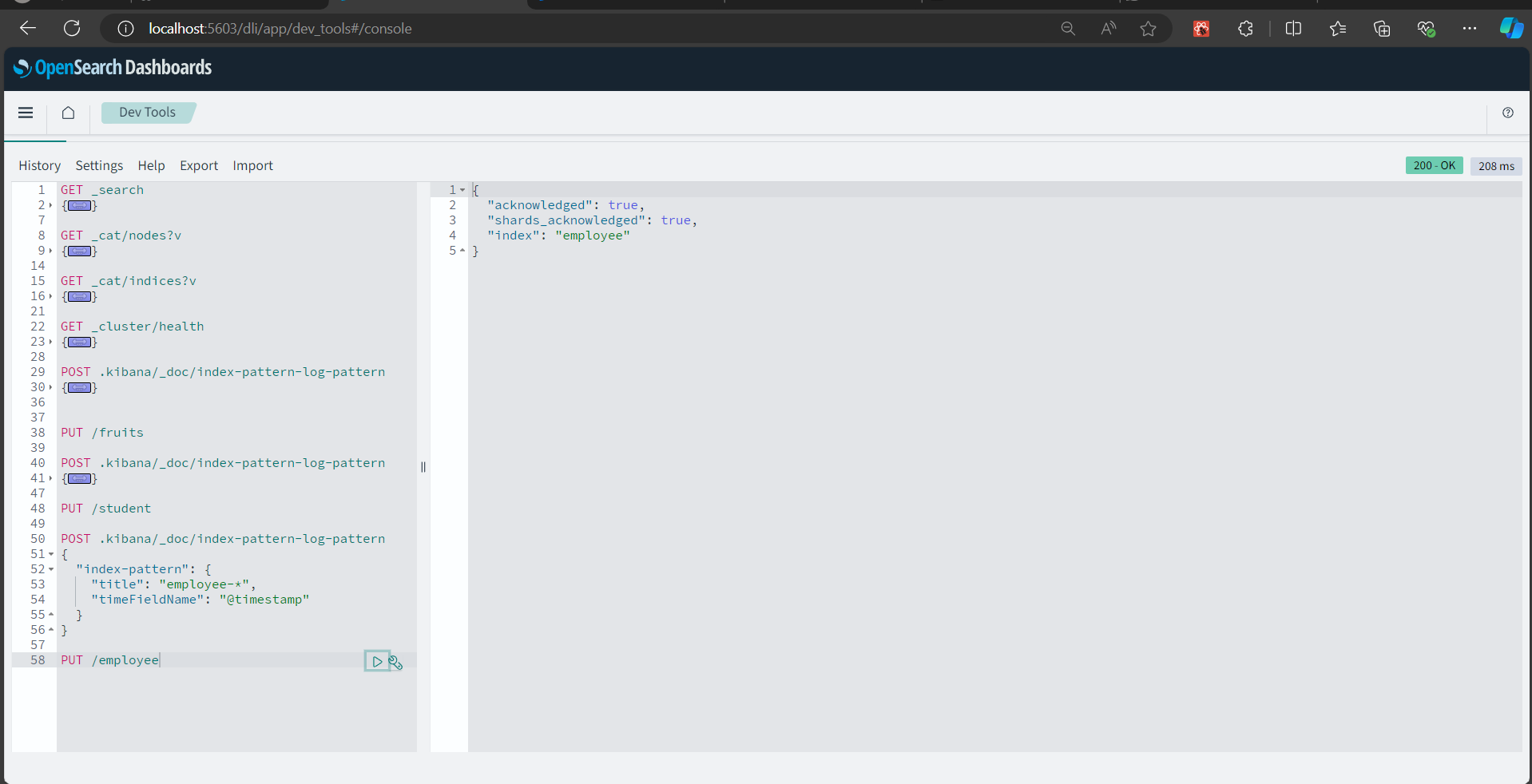
"timeFieldName": "@timestamp"

}

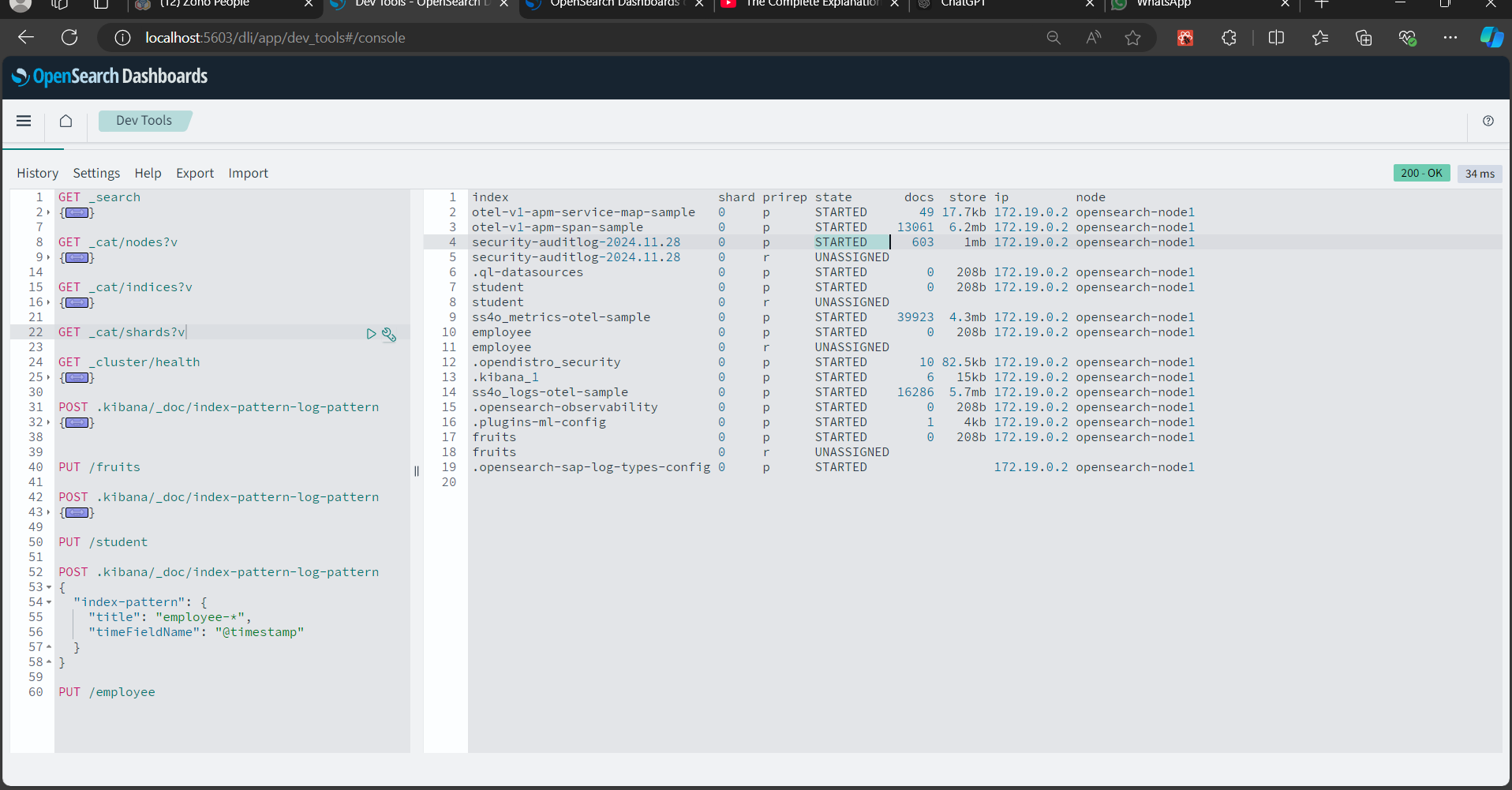
}

****

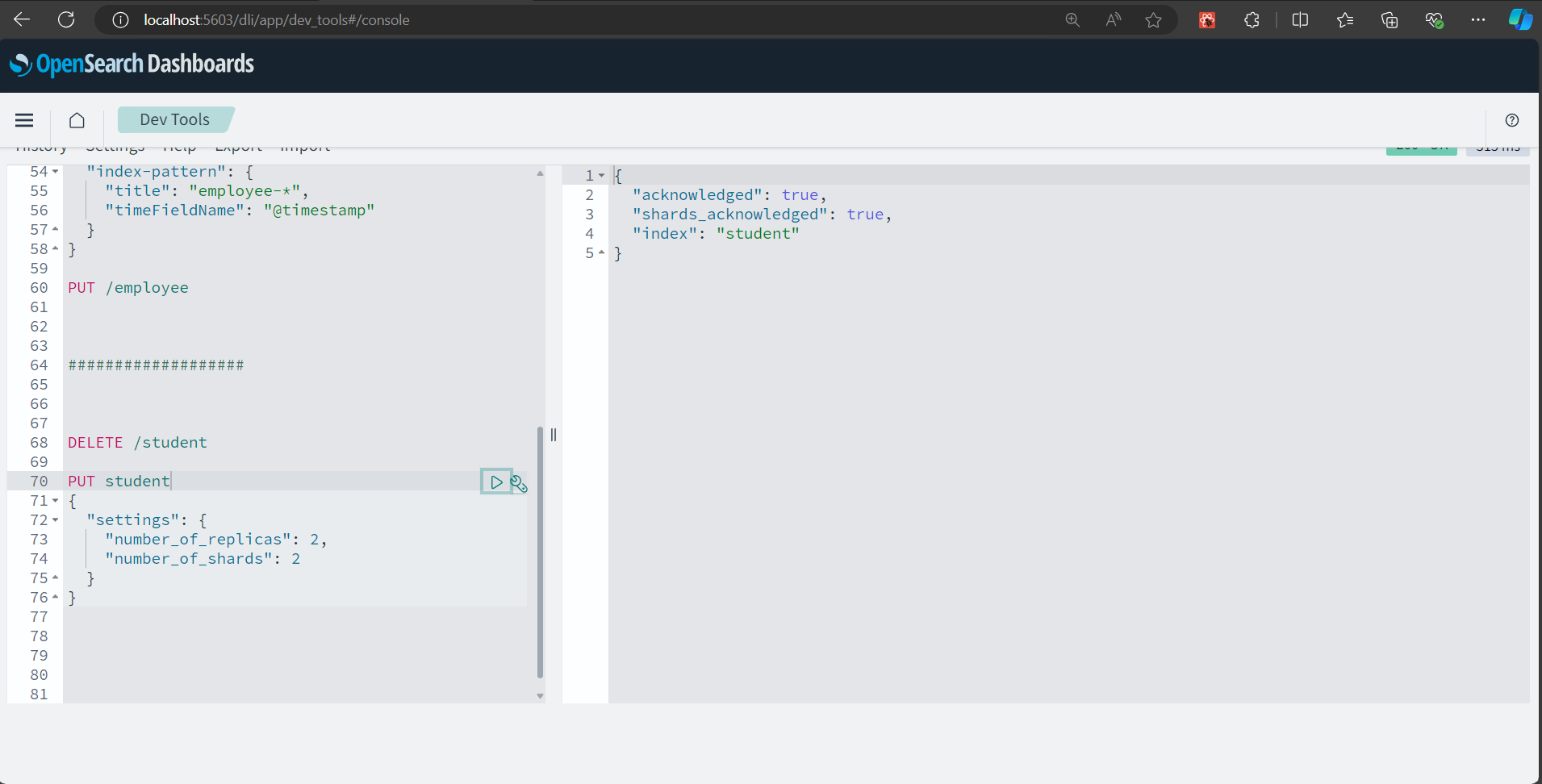
**PUT /employee**

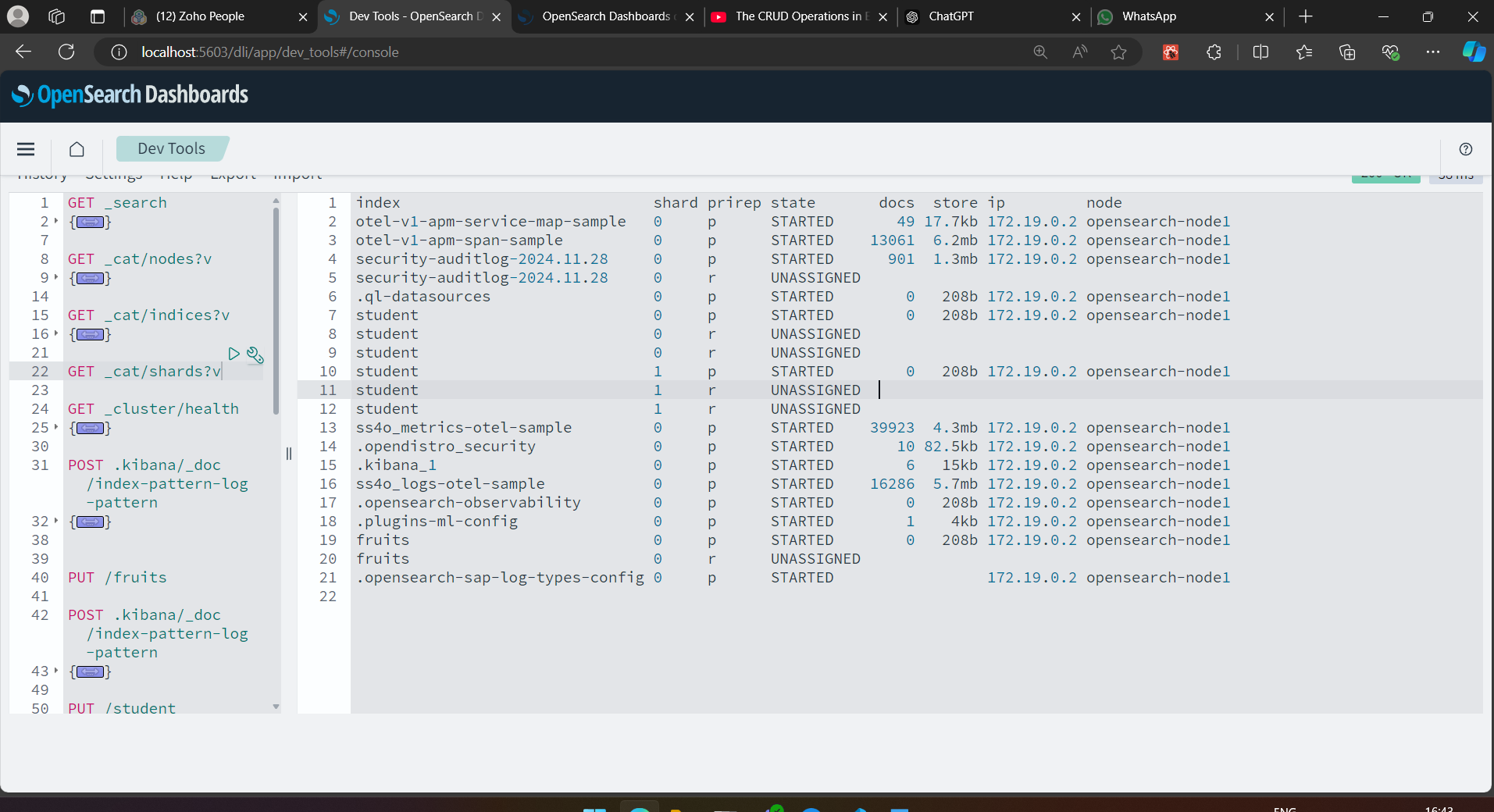
****

GET \_cat/shards?v

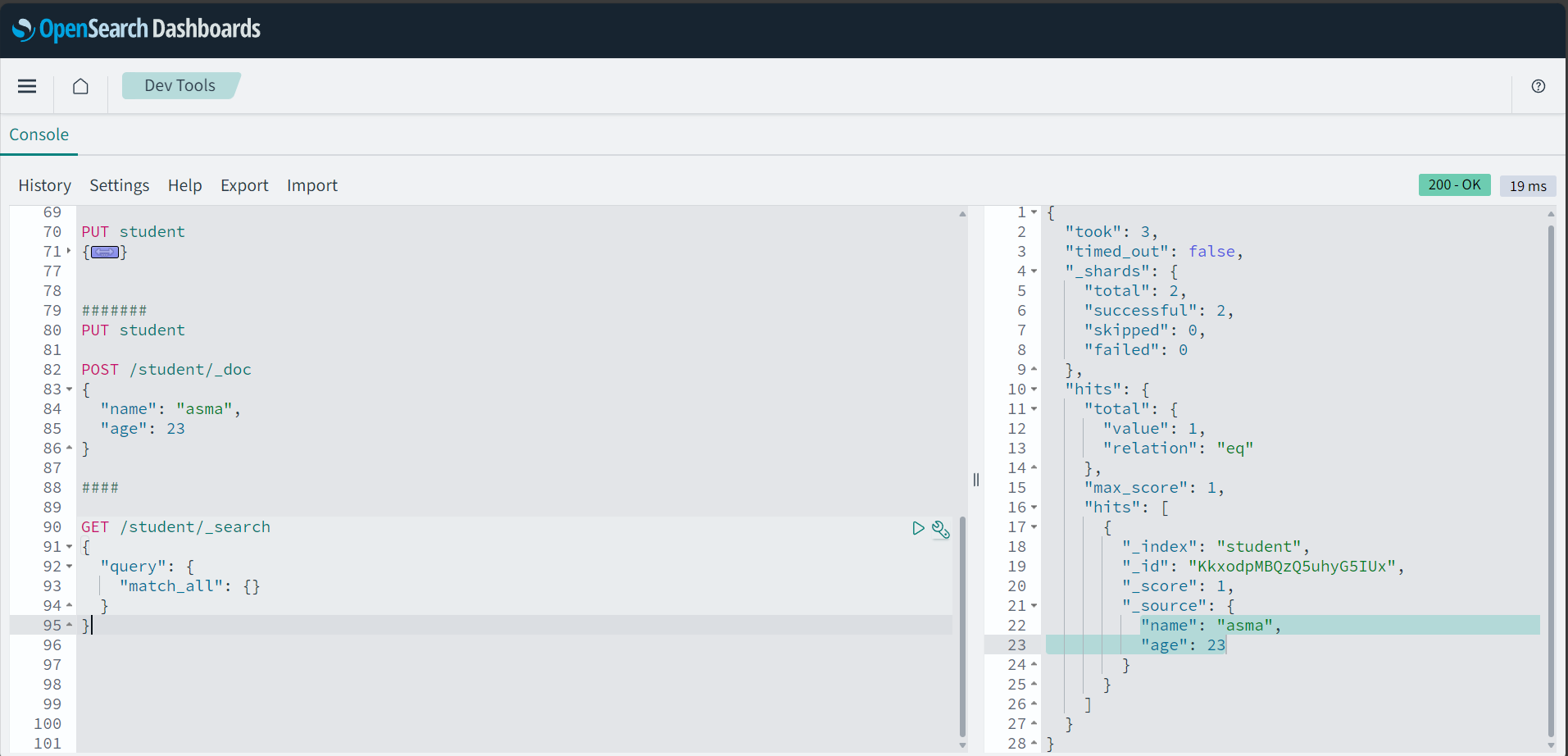
****

**Recreate shard and create 2 shard and 2 replicas**

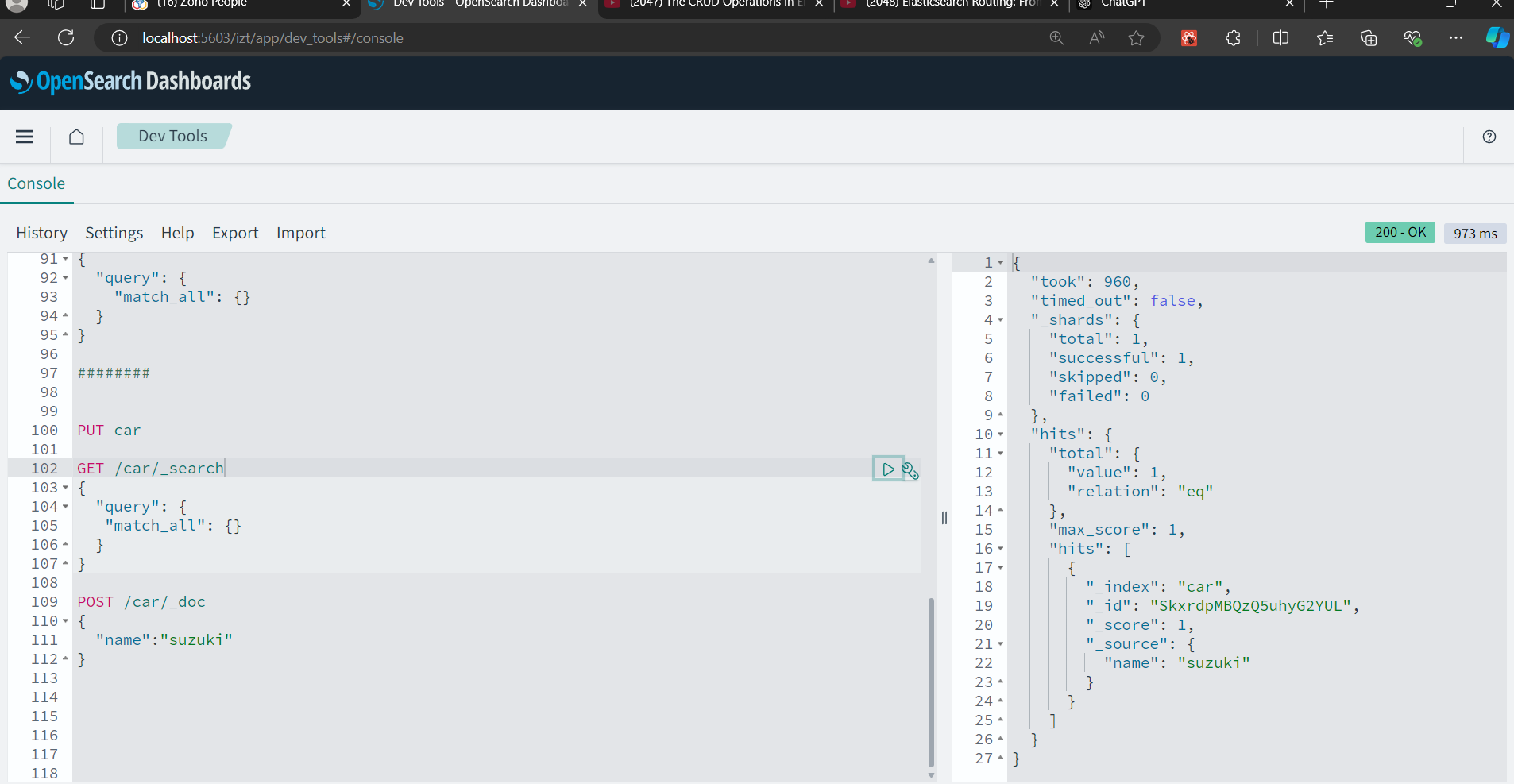
****

****

**Create one documents for student**

****

**Create one car object**

****

### PUT car

This command creates an index named car in Elasticsearch.

### GET /car/\_search

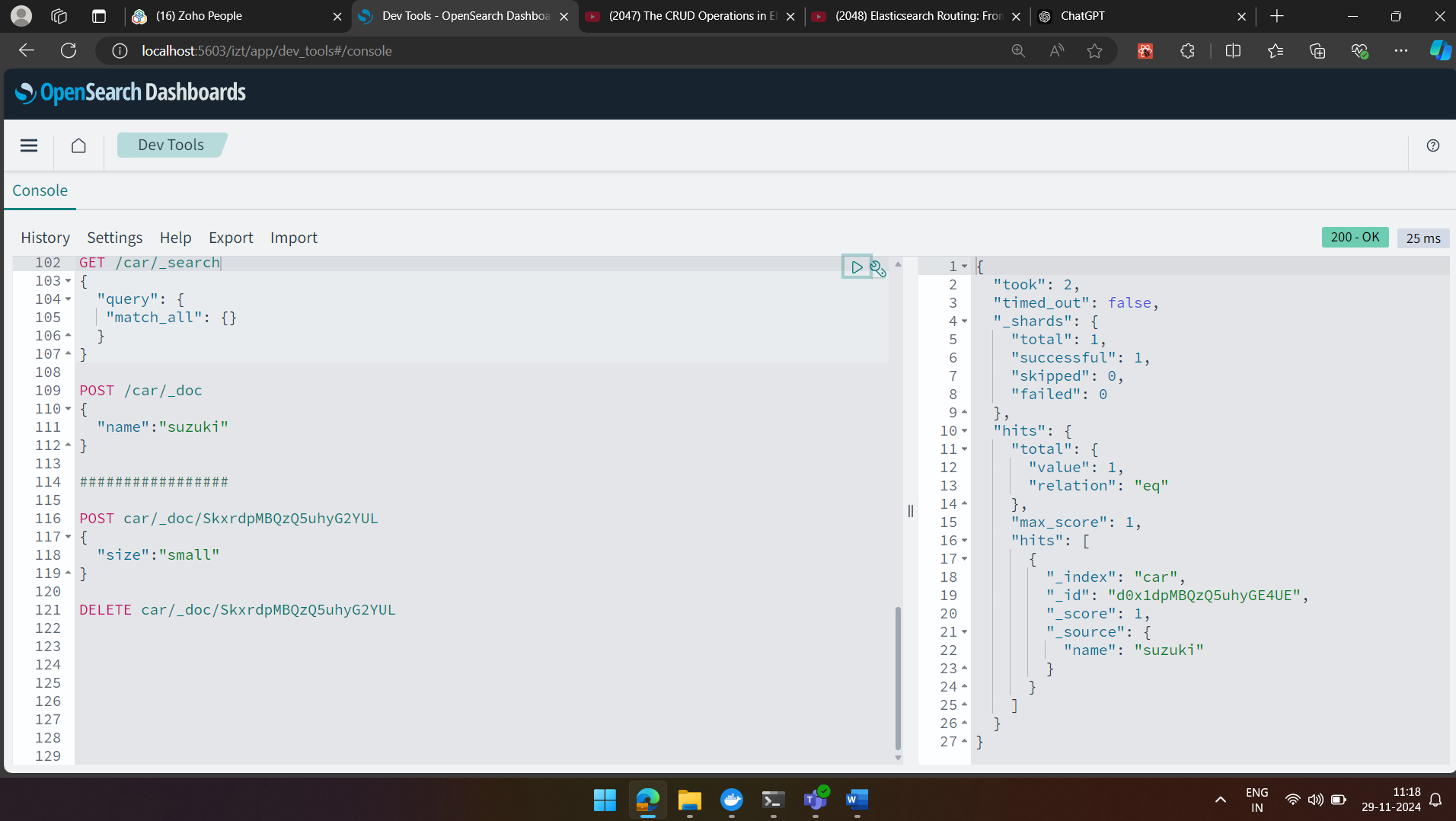
This command searches the car index for documents.

### POST /car/\_doc

This command adds a document to the car index.

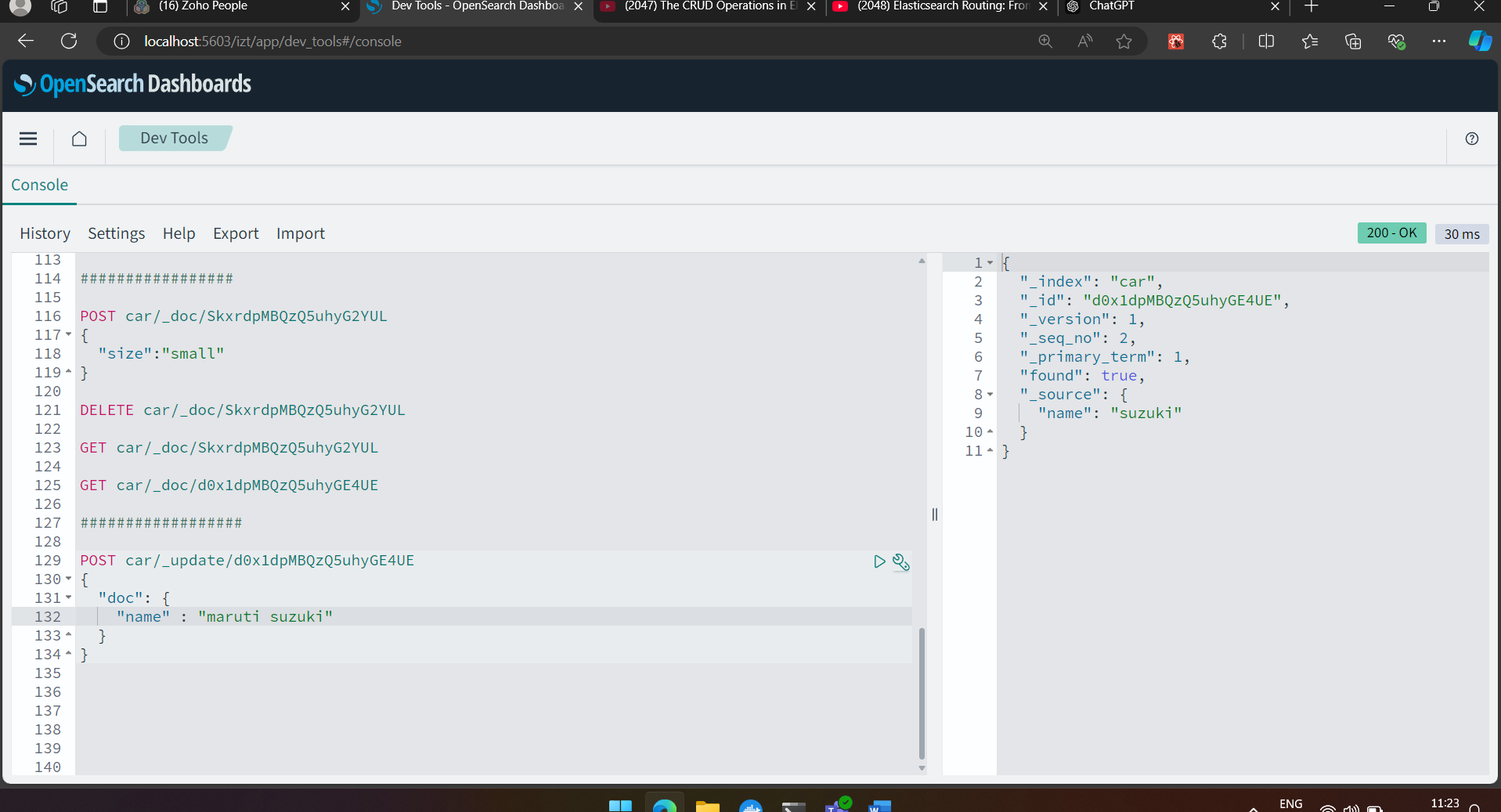
Using below commad we can delete particular index

DELETE car/\_doc/SkxrdpMBQzQ5uhyG2YUL

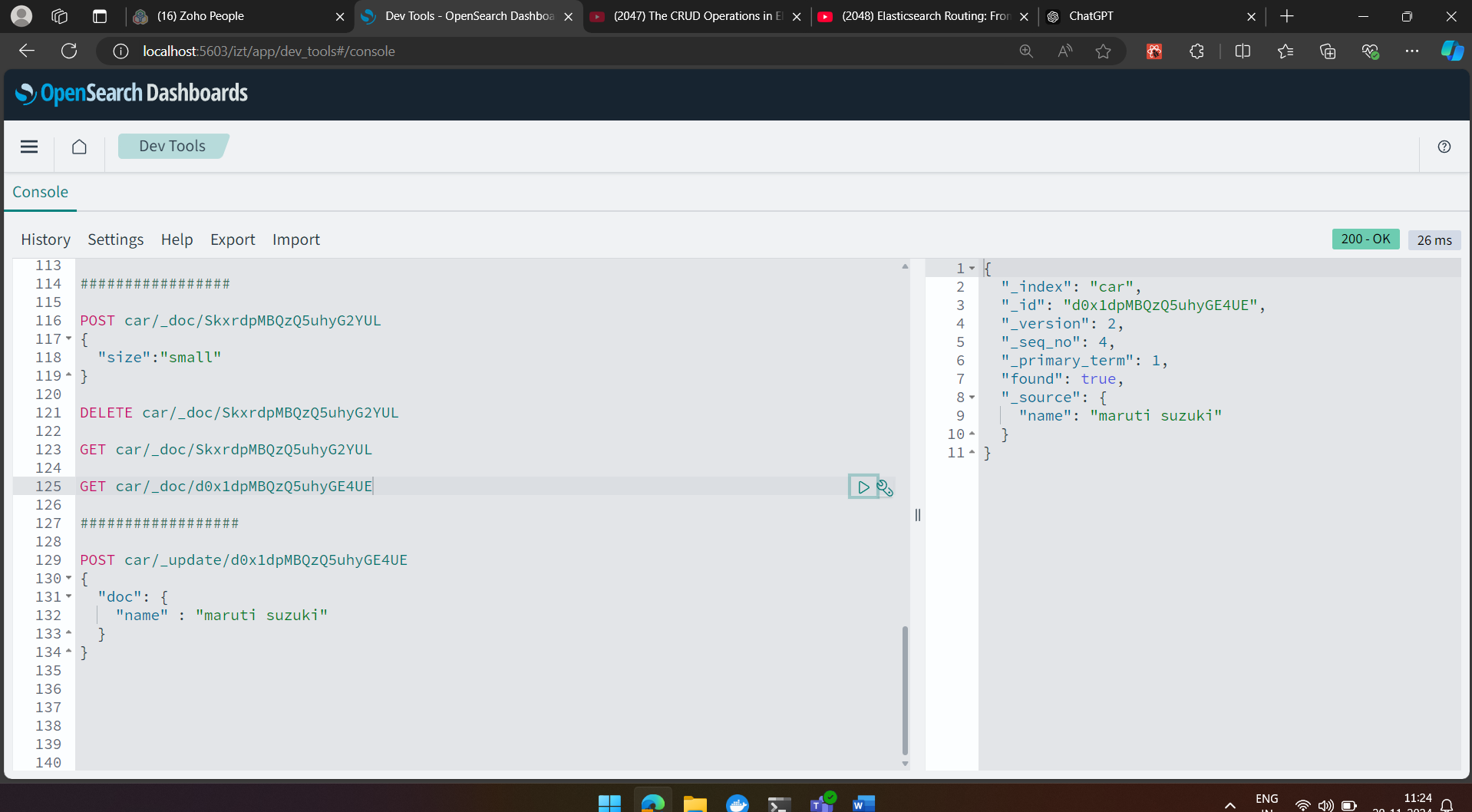
****

**Update the index value**

Before update



After update



Use script when we want to make a value dynamic

