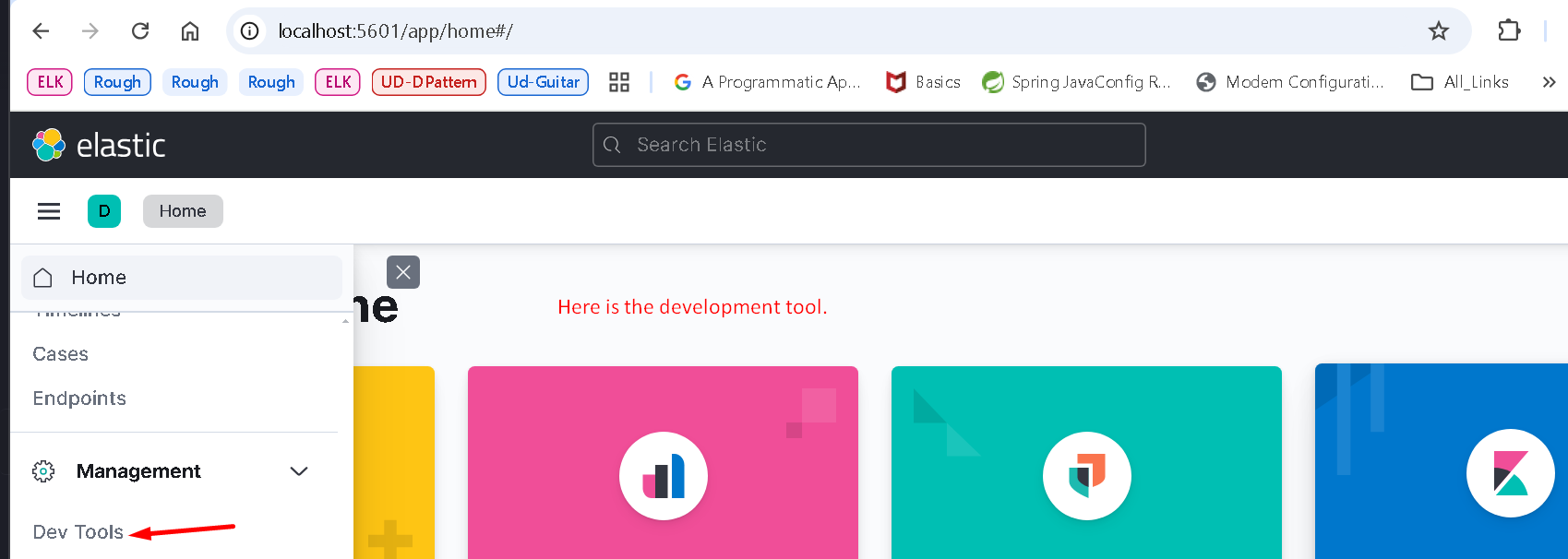
# 

# Detailed Notes: Elasticsearch Console Tool

## Overview

* - \*\*Purpose\*\*: Explore the Console tool, a **development utility** (as this tool is used during development phase) for interacting with Elasticsearch.
* - \*\*Audience\*\*: If already familiar, you can skip this lecture as it may not introduce new information.



## Key Features of Console Tool

* - \*\*Simplified Querying\*\*: Allows sending queries to Elasticsearch with easier syntax compared to cURL.
* - \*\*Helpful Features\*\*:

- JSON formatting.

- Auto-completion.

- Query history and local storage of queries.

## Accessing the Console Tool

* - \*\*Location\*\*: Found within the “Dev Tools” app in Kibana.
* - \*\*Interface\*\*:

- \*\*Left Pane\*\*: Editor for writing queries.

- \*\*Right Pane\*\*: Displays query results.

## Writing and Running Queries

1. 1. \*\*Basic Query\*\*:

- Example: Query cluster health.

- \*\*Steps\*\*:

1. Enter the HTTP verb (e.g., `GET`).

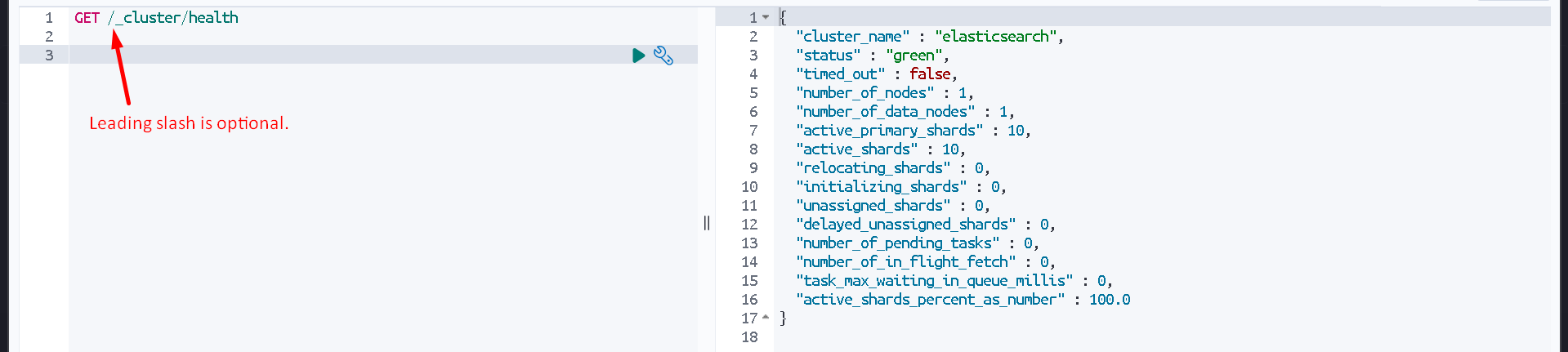
2. Add the request path (e.g., for Cluster health API).

3. Use auto-completion for assistance.

4. Optionally include a leading forward slash (`/`) in the path.

5. Run the query using the triangle button to the right.

- \*\*Results\*\*: Displayed as formatted JSON.



1. 2. \*\*Query with Request Body\*\*:

- Example: `match\_all` query.

- \*\*Structure\*\*:

- HTTP verb and request path.

- Request body containing parameters.

- \*\*Execution\*\*: Same process as above; results appear in the right pane.



## Convenience Features

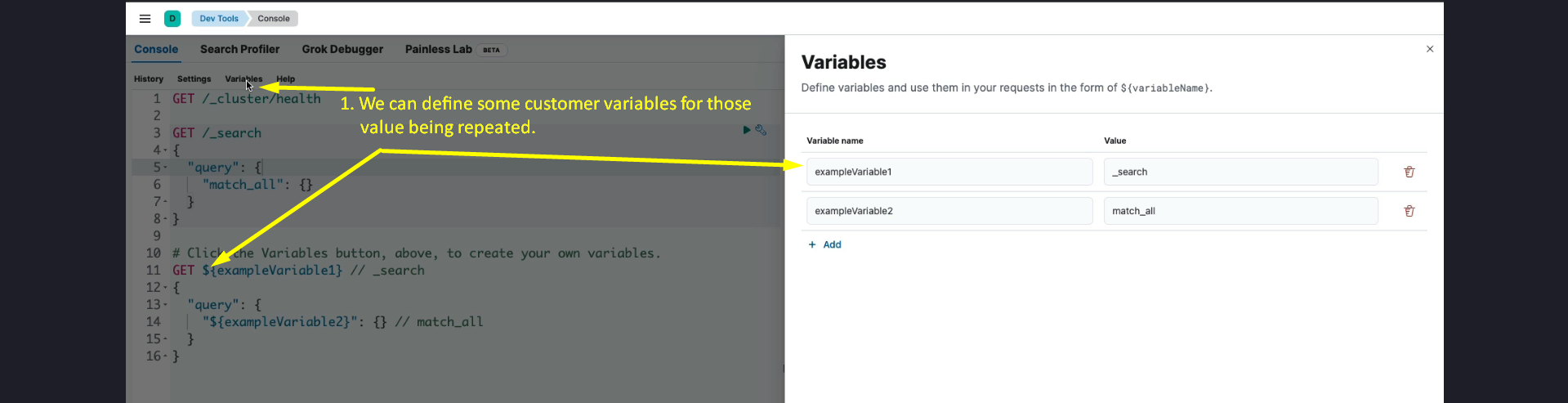
* - \*\*Query Storage\*\*:

- Queries are saved to local storage automatically. So, we can leave console tool (Dev Tool)

- Query history is maintained for quick access to previous queries.

* - \*\*Custom Variables\*\*:

- Variables can be defined and reused in queries.

- Useful for repetitive elements like index names.  


That is all about Console Tool used to quickly run queries against Elasticsearch Cluster.

## Use Cases

1. 1. \*\*Development\*\*:

- Writing and testing new queries.

- Easily porting queries to production.

1. 2. \*\*Debugging\*\*:

- Analyzing production queries.

- Inspecting Elasticsearch cluster issues.

## Comparison to cURL

* - Easier and faster than typing cURL commands.
* - Handles authentication and other complexities automatically.

## Limitations

* - Primarily a development tool.
* - Minimal use in this course, but knowledge of its operation is beneficial for upcoming tasks (e.g., adding index templates in the next lecture).

## Summary

* - \*\*Purpose\*\*: A user-friendly tool for interacting with Elasticsearch.
* - \*\*Advantages\*\*:

- Simplifies query creation.

- Offers time-saving features like auto-completion and query history.

- Useful for both development and debugging.

* - \*\*Next Steps\*\*: Practice with the tool by adding index templates in the following lecture.