Lab Report — Understanding Fields in Splunk

Name: Dhruvish
Date: October 20, 2025
Platform: Splunk Cloud Trial

Objective

The goal of this lab was to understand how **Splunk discovers, displays, filters, and transforms fields** during searches.
You practiced distinguishing between default and extracted fields, explored the **Fields sidebar**, shaped results with commands like fields, table, rename, and dedup, and created both **ad-hoc and saved field extractions** using rex and the **Field Extractor**.

Tools Used

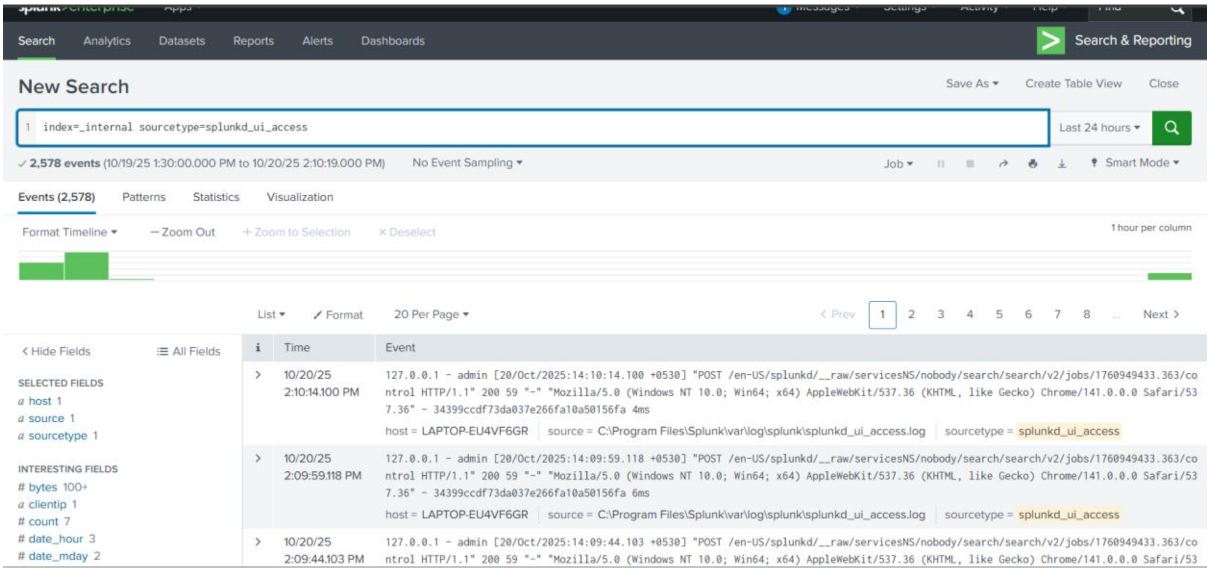
- **Splunk Cloud Trial** (or Splunk Free)
- **Web Browser**

Procedure and Observations

Step 1: Open Search & Pick a Friendly Dataset

- Navigated to **Apps ‣ Search & Reporting**.
- Set the time range to **Last 24 hours**.
- Ran the following search using the internal access logs dataset:

  ```
  index=_internal sourcetype=splunkd_ui_access
  ```
- Observed default metadata fields such as _time, host, source, sourcetype, and index.
- Noted automatically extracted fields like method, status, uri_path, and user.



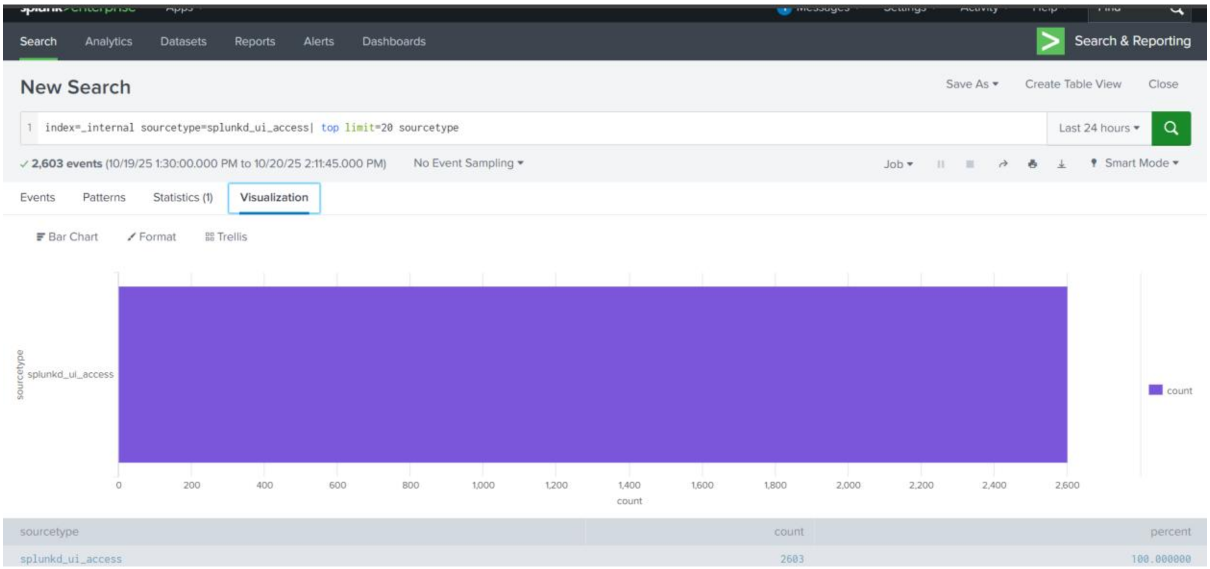
```
index=_internal sourcetype=splunkd_ui_access
```

✓ 2,578 events (10/19/25 1:30:00.000 PM to 10/20/25 2:10:19.000 PM)

i	Time	Event
>	10/20/25 2:10:14.100 PM	127.0.0.1 - admin [20/Oct/2025:14:10:14.100 +0530] "POST /en-US/splunkd/__raw/servicesNS/nobody/search/search/v2/jobs/1760949433.363/control HTTP/1.1" 200 59 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 Safari/537.36" - 34399ccdf73da037e266fa10a50156fa 4ms host = LAPTOP-EU4VF6GR source = C:\Program Files\Splunk\var\log\splunk\splunkd_ui_access.log sourcetype = splunkd_ui_access
>	10/20/25 2:09:59.118 PM	127.0.0.1 - admin [20/Oct/2025:14:09:59.118 +0530] "POST /en-US/splunkd/__raw/servicesNS/nobody/search/search/v2/jobs/1760949433.363/control HTTP/1.1" 200 59 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 Safari/537.36" - 34399ccdf73da037e266fa10a50156fa 6ms host = LAPTOP-EU4VF6GR source = C:\Program Files\Splunk\var\log\splunk\splunkd_ui_access.log sourcetype = splunkd_ui_access
>	10/20/25 2:09:44.103 PM	127.0.0.1 - admin [20/Oct/2025:14:09:44.103 +0530] "POST /en-US/splunkd/__raw/servicesNS/nobody/search/search/v2/jobs/1760949433.363/control HTTP/1.1" 200 59 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 Safari/53

Step 2: Meet the Fields Sidebar

- Opened the **Fields sidebar** (on the left).
- Expanded **Selected Fields** and **Interesting Fields**.
- Hovered over each field to preview top values and clicked some to apply quick filters.
- Pinned the user and status fields for quick visibility in all searches.



```
index=_internal sourcetype=splunkd_ui_access| top limit=20 sourcetype
```

✓ 2,603 events (10/19/25 1:30:00.000 PM to 10/20/25 2:11:45.000 PM)

sourcetype	count	percent
splunkd_ui_access	2603	100.000000

Step 3: Include, Exclude, and Clean Up Fields

Practiced shaping the results by adding/removing fields and renaming columns.

- Display only selected columns:

  index=_internal sourcetype=splunkd_ui_access | fields _time user method uri_path status

- Create a clean table output:

index=_internal sourcetype=splunkd_ui_access | table _time user method uri_path status

- Hide noisy fields:

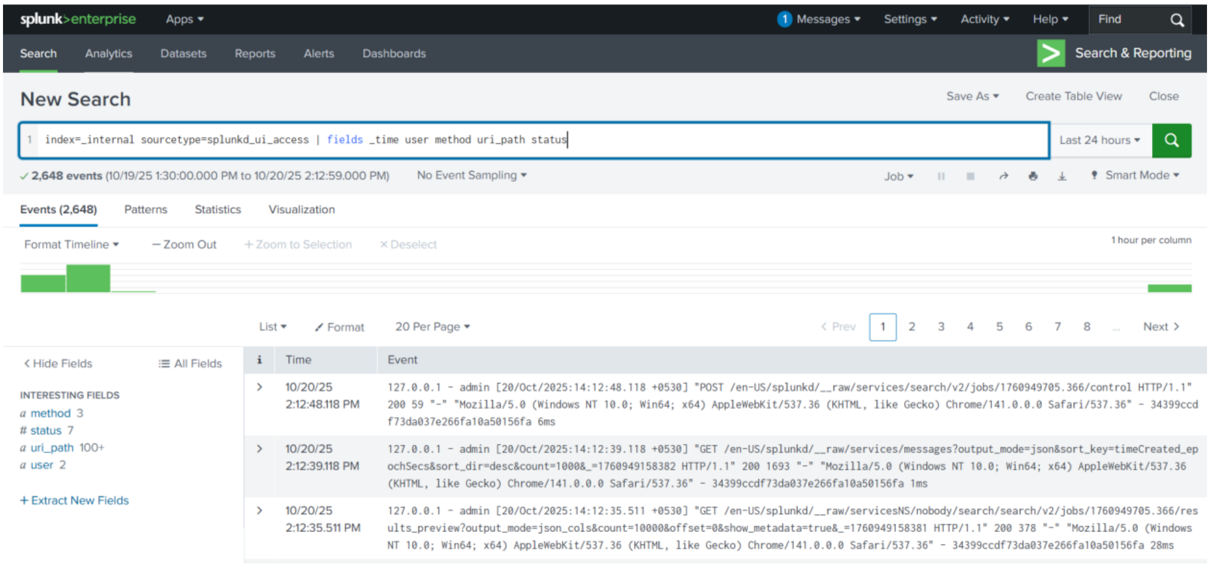
  index=_internal sourcetype=splunkd_ui_access | fields - punct, linecount

- Rename for clarity:

  index=_internal sourcetype=splunkd_ui_access | table _time status uri_path | rename uri_path AS url_path, status AS http_status

- List each user once:

  index=_internal sourcetype=splunkd_ui_access | dedup user | table user



```
index=_internal sourcetype=splunkd_ui_access | fields _time user method uri_path status
```

✓ 2,648 events (10/19/25 1:30:00.000 PM to 10/20/25 2:12:59.000 PM)

INTERESTING FIELDS

a method	3
# status	7
a uri_path	100+
a user	2

i	Time	Event
>	10/20/25 2:12:48.118 PM	127.0.0.1 - admin [20/Oct/2025:14:12:48.118 +0530] "POST /en-US/splunkd/__raw/services/search/v2/jobs/1760949705.366/control HTTP/1.1" 200 59 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 Safari/537.36" - 34399ccdf73da037e266fa10a50156fa 6ms
>	10/20/25 2:12:39.118 PM	127.0.0.1 - admin [20/Oct/2025:14:12:39.118 +0530] "GET /en-US/splunkd/__raw/services/messages?output_mode=json&sort_key=timeCreated_epochSecs&sort_dir=desc&count=1000&_=1760949158382 HTTP/1.1" 200 1693 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 Safari/537.36" - 34399ccdf73da037e266fa10a50156fa 1ms
>	10/20/25 2:12:35.511 PM	127.0.0.1 - admin [20/Oct/2025:14:12:35.511 +0530] "GET /en-US/splunkd/__raw/servicesNS/nobody/search/search/v2/jobs/1760949705.366/results_preview?output_mode=json_cols&count=10000&offset=0&show_metadata=true&_=1760949158381 HTTP/1.1" 200 378 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/141.0.0.0 Safari/537.36" - 34399ccdf73da037e266fa10a50156fa 28ms

Step 4: Summaries by Field

Used stats and top to aggregate field data.

- Top 5 users:

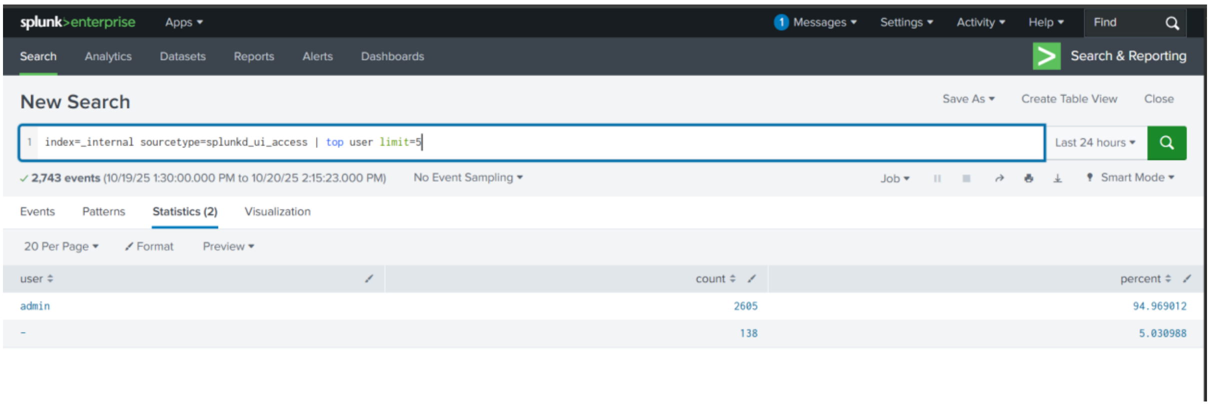
  index=_internal sourcetype=splunkd_ui_access | top user limit=5

- Count by status code:

  index=_internal sourcetype=splunkd_ui_access | stats count by status

- Method vs. status matrix:

  index=_internal sourcetype=splunkd_ui_access | stats count by method status

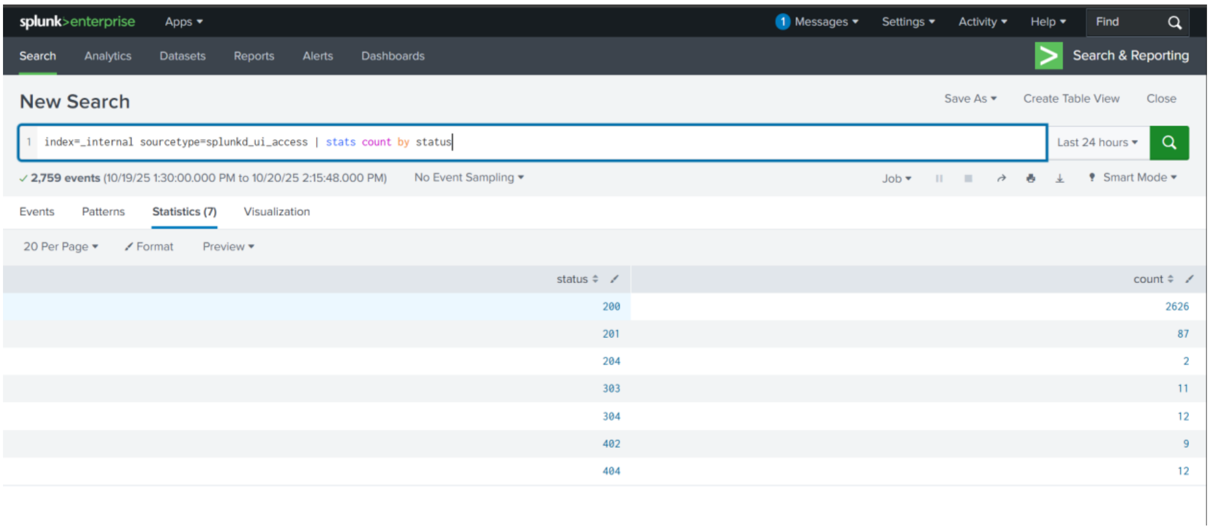


New Search

```
index=_internal sourcetype=splunkd_ui_access | top user limit=5
```

✓ **2,743 events** (10/19/25 1:30:00.000 PM to 10/20/25 2:15:23.000 PM)

user	count	percent
admin	2605	94.969012
-	138	5.030988



New Search

```
index=_internal sourcetype=splunkd_ui_access | stats count by status
```

✓ **2,759 events** (10/19/25 1:30:00.000 PM to 10/20/25 2:15:48.000 PM)

status	count
200	2626
201	87
204	2
303	11
304	12
402	9
404	12

Step 5: Ad-Hoc Field Extraction with rex

Created a new field dynamically using **regular expressions**.

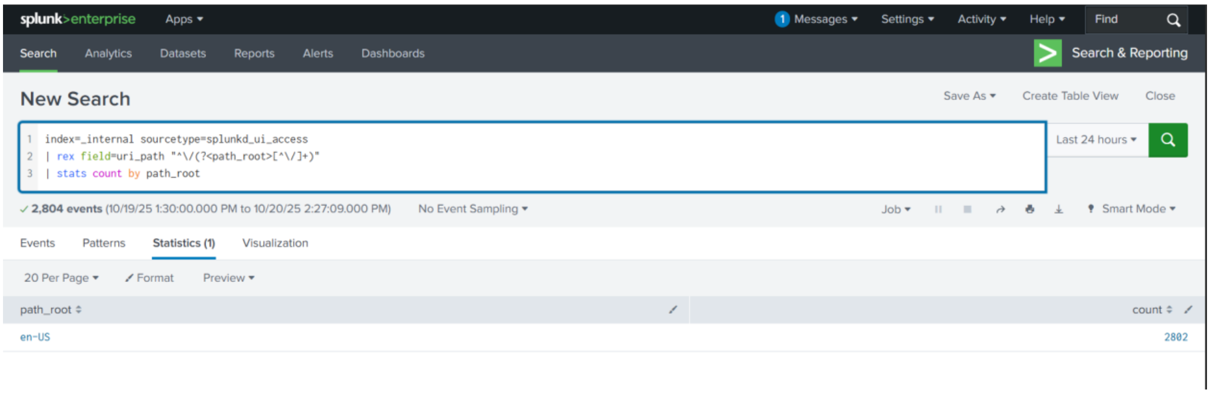
- Extracted the first directory segment from uri_path:

  index=_internal sourcetype=splunkd_ui_access

  | rex field=uri_path "^\/(?<path_root>[^\/]+)"

  | stats count by path_root

- Verified the new field path_root appeared in results.



```
index=_internal sourcetype=splunkd_ui_access
| rex field=uri_path "^\/(?<path_root>[^\/]+)"
| stats count by path_root
```

✓ 2,804 events (10/19/25 1:30:00.000 PM to 10/20/25 2:27:09.000 PM)   No Event Sampling

path_root	count
en-US	2802

Step 6: Saved Field Extraction (Field Extractor)

- Ran a focused search:

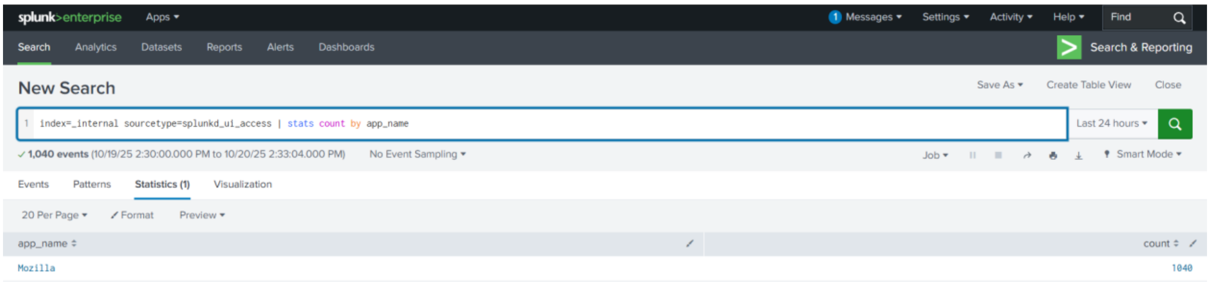
  index=_internal sourcetype=splunkd_ui_access uri_path="*app*"

- From **Event Actions** ▸ **Extract Fields (FX)**, created a regex-based extraction for **app_name**.

- Saved it to the app context with appropriate permissions.

- Validated using:

  index=_internal sourcetype=splunkd_ui_access | stats count by app_name



```
index=_internal sourcetype=splunkd_ui_access | stats count by app_name
```

✓ 1,040 events (10/19/25 2:30:00.000 PM to 10/20/25 2:33:04.000 PM)   No Event Sampling

app_name	count
Mozilla	1040

Step 7: Calculated Fields with eval

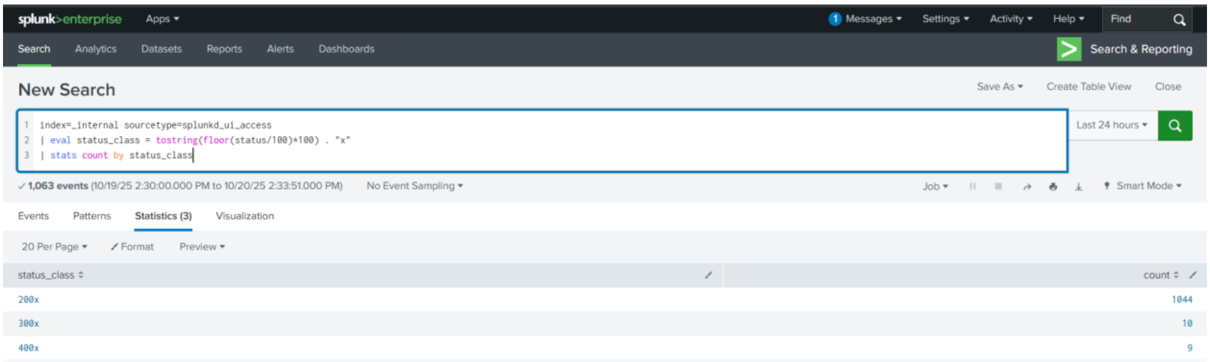
Derived a new field from the existing HTTP status field.

- Grouped response codes into classes (e.g., 2xx, 3xx, 4xx):

  index=_internal sourcetype=splunkd_ui_access

  | eval status_class = tostring(floor(status/100)*100) . "x"

  | stats count by status_class



```
index=_internal sourcetype=splunkd_ui_access
| eval status_class = tostring(floor(status/100)*100) . "x"
| stats count by status_class
```

✓ 1,063 events (10/19/25 2:30:00.000 PM to 10/20/25 2:33:51.000 PM) No Event Sampling

status_class	count
200x	1044
300x	10
400x	9

Step 8: (Optional) Field Aliases for Friendly Names

- Created a **Field Alias** mapping:
  - Source field: status
  - Alias: http_status
- Verified with:

  index=_internal sourcetype=splunkd_ui_access | stats count by http_status

Reflection

- **Default vs. Extracted Fields:**
  Default fields (e.g., _time, host, source) come from event metadata, while extracted fields (e.g., method, status, user) are parsed dynamically from event data.
  In daily analysis, extracted fields add context and enable meaningful filters.
- **Ad-hoc vs. Saved Extractions:**
  Use rex for temporary, one-off field extractions during analysis; use **Field Extractor** for persistent fields accessible across searches.
- **Calculated Fields & Aliases:**
  These features enhance consistency and readability across teams and apps, ensuring that analysts use uniform field names and groupings.

Summary

This lab demonstrated how to explore and manipulate fields within Splunk. You learned to:

- Identify default and extracted fields
- Use the **Fields sidebar** effectively
- Shape and clean output with fields, table, rename, and dedup
- Perform **ad-hoc** and **saved** extractions
- Create **calculated fields** and **aliases** for clarity and consistency

These skills are essential for building **efficient searches**, **standardized dashboards**, and **collaborative analytics** workflows in real-world Splunk environments.