

Dynamic Dashboards

Before Taking This Course

- To be successful, students must have a working understanding of these courses:
 - Intro to Splunk
 - Introduction to Dashboards

Course Objectives

- Define token syntax
- Create user inputs
- Create dynamic inputs
- Build cascading inputs
- Create a dynamic drilldown
- Set tokens
- Use dynamic coloring

Course Outline

- Using Tokens
- Adding Inputs
- Using Drilldowns
- Dynamic Visualizations

Topic 1: Using Tokens

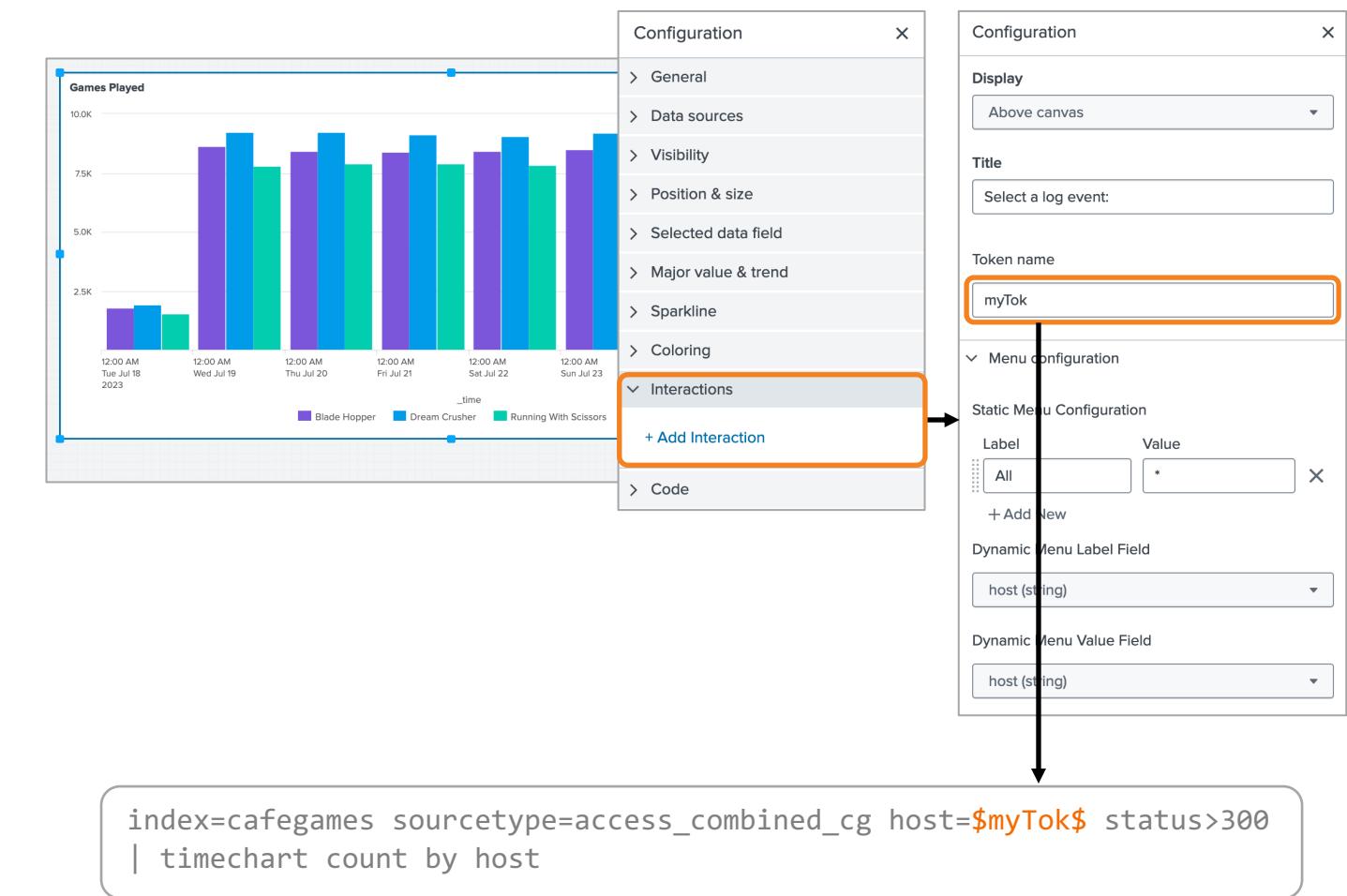
Topic Objectives

- Define dashboard token syntax
- Describe types of predefined tokens
- Use token filters

Dashboard Tokens

- Syntax
 - Use \$...\$ delimiters to access the value of a token
- Types
 - User defined
 - Predefined

Search, environment, inputs, drilldowns, maps, and visualizations
 - Default token



Predefined Tokens – Search

- Set tokens from search job metadata or search results
 - Embed search-related information in other searches or visualizations

Search Results	
\$search name:result.<field_name>\$	Returns the first value for the specified field name from the first result in the search.
Search Job Metadata	
\$search name:job.done\$	Is the job is done? True or false.
\$search name:job.failed\$	Did the job fail? True or false.
\$search name:job.hasResults\$	Did the search return results? True or false.
\$search name:job.inProgress\$	Is the job in progress? True or false.
\$search name:job.isRealTimeSearch\$	Does the job use a real time search? True or false.
\$search name:job.lastUpdated\$	Returns a timestamp of the last update.
\$search name:job.messages\$	List of error and debug messages.
\$search name:job.percentComplete\$	The job's percentage of completeness. Numerical value.
\$search name:job.queued\$	Is the job is queued? True or false.
\$search name:job.resultCount\$	Number of results returned. Returns an integer.
\$search name:job.sid\$	Returns the search job ID.
\$search name:job.startTime\$	Time a search job started. Returns date and time.
\$search name:job.status\$	What is the status? Done, queued, in progress, or failed.
\$search name:result.<field>\$	Returns the first result for the specified field.

Predefined Tokens – Search Example 1

- Search job metadata
 - Search job status

Syntax: \${Search Name:job.status\$}

Returns: queued, inProgress, done, or failed

Edit Data Source

Data source name
Game Sales Search

Access search results or metadata

SPL query
index=cafegames sourcetype=access_combined_cg
product_name=*
| chart count by product_name

Configuration

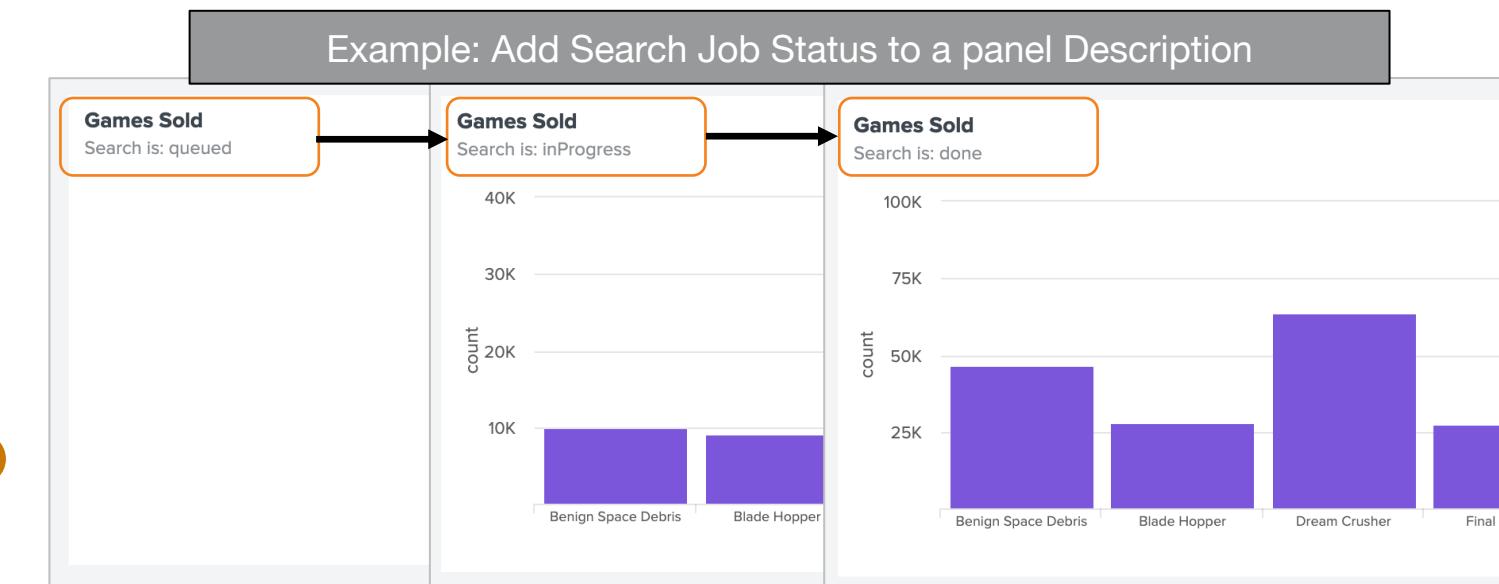
General

Visualization type: Column

Title: Games Sold

Description: Search is: \${Game Sales Search:job.status\$}

Data sources: Game Sales Search



Predefined Tokens – Search Example 2

- Search results
 - Syntax: `$Search Name:result.<field>$`
 - Returns: first value from the first result of the named field

Edit Data Source

Data source name: Table Search

Access search results or metadata

SPL query

```
index=cafegames sourcetype=access_combined_cg
| stats sum(player1score) AS Score by player1name,
  product_name
| rename player1name AS Player, product_name as Game
| sort -Score
```

[Open in Search](#)

Configuration

General

Visualization type: Table

Title: Score Leader: \$Table Search:result.Player\$

Description: Top Score: \$Table Search:result.Score\$

Data sources: Table Search

Example: Add Score Leader and Top Score to Panel

#	Player	Game	Score
1	tcrosioag	Marshmallow Missiles	1980
2	ipassod	Benign Space Debris	1859
3	mpledgerg	Blade Hopper	1825
4	candraultpf	Orvil the Wolverine	1823
5	mnottram7x	Marshmallow Missiles	1820

< Prev 1 2 3 4 5 ... Next >

Score Leader - tcrosioag
Top Score: 1980

Predefined Tokens – Environment Example

- Capture user details and Splunk platform instance and embed on dashboards:
 - Titles, labels, or text
 - Search queries
- Can read environment token data but not write environment token data

1. \$env:user\$	Current user's username: supportuser
2. \$env:user_realname\$	Current user's full name: Administrator
3. \$env:user_email\$	Current user's email address: changeme@example.com
4. \$env:app\$	Current app context: dynamic_dash
5. \$env:locale\$	Current locale: en-US
6. \$env:page\$	Current open page: environment_token_example
7. \$env:product\$	Current instance's product type: enterprise
8. \$env:version\$	Current instance's product version: 9.0.0
9. \$env:is_cloud\$	Current instance is Splunk Cloud: \$env:is_cloud\$
10. \$env:is_enterprise\$	Current instance is Splunk Enterprise: true
11. \$env:is_lite_free\$	Current instance is using a Splunk Light free license: \$env:is_lite_free\$
12. \$env:is_free\$	Current instance is using a Splunk Splunk Enterprise free license: \$env:is_free\$

These tokens are only set only if 'true'

Predefined Tokens – Visualizations

- Set tokens and capture values from a visualization click
- Use the token elsewhere in the dashboard
- Three predefined tokens:
 - **name**
 - **value**
 - **row.<fieldname>.value**
- Captured values vary according to visualization type

Example: Set a token to capture the HTTP Request Method

1

HTTP Request Method

20K
15K
10K
5K
count

DELETE GET HEAD POST method

Response Codes for Method GET

404, 6.879%
401, 0.193%
303, 0.386%
200, 92.514%

2

3

Configuration

- General
- Data sources
- Data configurations
- Data display
- Color and style
- Legend
- X-axis grid and labels
- Y-axis grid and labels
- Y2-axis grid and labels
- Interactions

Set token (method)

Set tokens

Default value

Code

2

Configuration

On click

Set tokens

Tokens are used to configure interactivity in the dashboard.

For example: host = row.**host**.value

name
Field name of the value/location clicked

value
Value of the location clicked

row.<fieldname>.value
Value in the specified series corresponding to the location clicked

Token name

method = Choose an event

Default value

GET

+ Set another token

Cancel Apply

3

Default Token

Used when there is no other token value

This is for the dashboard description.

Global Time Range
Last 24 hours

Server Errors

Chart A (Bar Chart):
Y-axis: count (0.5M to 1.5M). X-axis: log_level (ERROR, INFO, WARN). The ERROR bar is purple and reaches approximately 1.2M, while the INFO bar reaches approximately 0.8M.

Table B (Server Error Detail):
log_level | component | count
INFO Metrics 1099479
INFO PeriodicHealthReporter 109774
INFO LicenseUsage 12348
INFO TailReader 6312
INFO WatchedFile 5833

On Click: Set Tokens

Configuration

On Click
Set Tokens

Tokens are used to configure interactivity in the dashboard.
For example: host = row.host.value

Set Token (errorTok)

Token Name: errorTok
Token Value: row.log_level.v1

A predefined token captures information when a user clicks different visualization elements.

Default Value: *

+ Set Another Token

Cancel Apply

index=_internal sourcetype=splunkd log_level=\$errorTok\$
| stats count by log_level component | sort -count

The default value for **errorTok**, the wildcard asterisk (*), is stored as text in the dashboard source code, in the defaults section

```
...
"defaults": {
  "dataSources": {
    "ds.search": {
      "options": {
        "queryParameters": {
          "latest": "$global_time.latest$",
          "earliest": "$global_time.earliest$"
        }
      }
    }
  }
},
"tokens": {
  "defaults": {
    "errorTok": {
      "value": "*"
    }
  }
}
},
```

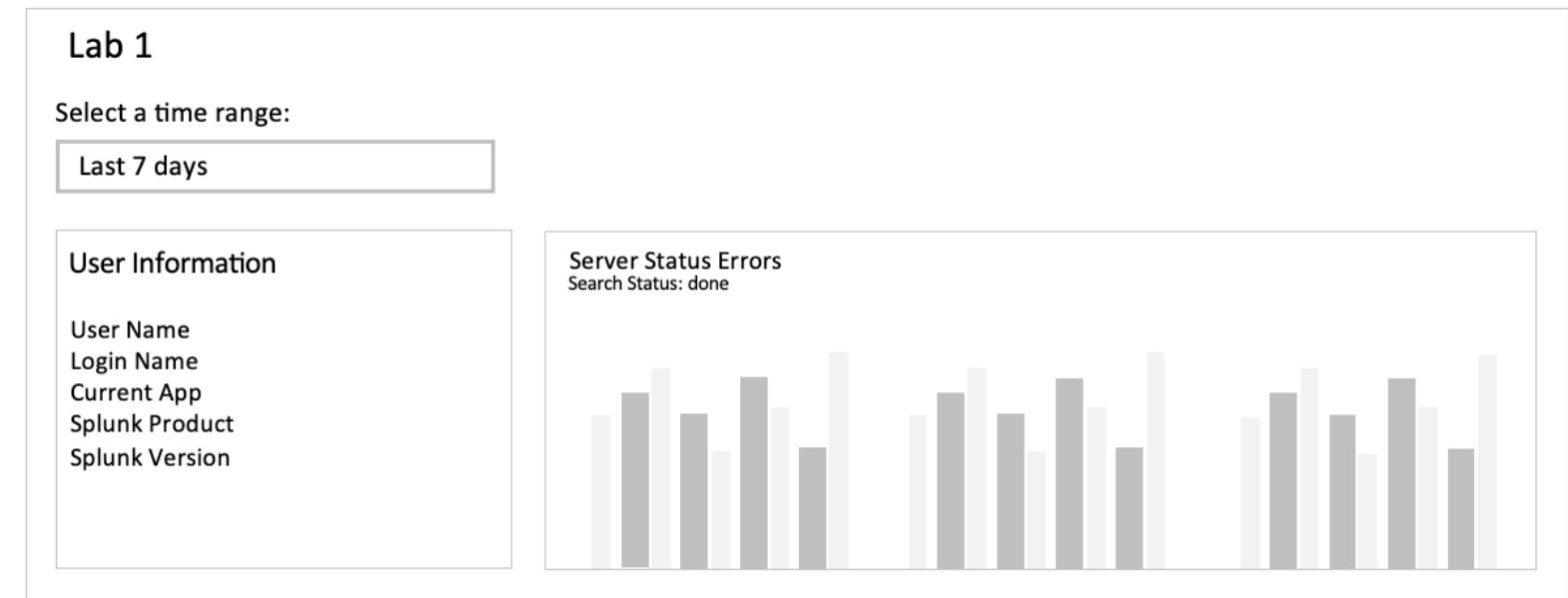
Token Filters

Token filters ensure that you correctly capture a token's value

Token Filter	Description
<code>\$token_name s\$</code>	Wrap value in quotes
<code>\$token_name h\$</code>	Escape any HTML in value
<code>\$token_name u\$</code>	Encode URL values
<code>\$token_name n\$</code>	No encode
<code>\$\$token_name\$\$</code>	Escape the \$ token delimiter character

Lab Exercise 1 – Using Tokens

- **Description:** Create a dashboard, add a chart, and search token.
- **Duration:** 10 minutes
- **Tasks:**
 - Create a dashboard
 - Add markdown text
 - Add a chart
 - Add predefined tokens



Topic 2: Adding Inputs

Topic Objectives

- Name the types of inputs
- Create a dynamic input
- Create cascading inputs

Inputs

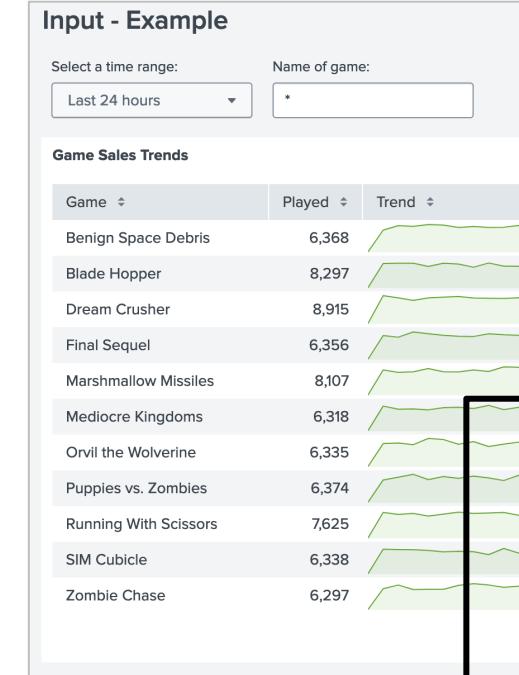
- Types: dropdown, multiselect, text, number, and time range
- Input tokens
 - Automatically generated
 - Customizable
- Visualization editor
 - Customize token name
 - Add static values
 - Specify default values
 - Define the data source
- Source editor
 - Configure when input values populate:
Page load, input change, Submit button clicked

The screenshot shows the Splunk Visualization editor interface. On the left, there's a sidebar with icons for different visualization types: Dropdown, Multiselect, Text, Number, and Time Range. Below these are sections for 'Presets' and 'Time Range'. The 'Presets' section lists various time ranges like 'Last 24 hours', 'Real-time', 'Relative', and 'Other'. The 'Time Range' section shows a dropdown set to 'Last 24 hours' and a detailed view of other time range options such as '30 second window', 'Business week to date', and 'All time (real-time)'.

The screenshot shows the Splunk Source editor interface. It displays four input token configurations: 'Dropdown' (set to 'All'), 'Multiselect' (set to 'All', with 'filter' in the search bar and 'All' selected), 'Text' (set to 'Default Te|'), and 'Number' (set to '11'). Each token has a dropdown menu below it showing additional options or values.

Adding Inputs

- Using the visual editor:
 - Source code automatically created
- Using the source editor:
 - Input ID must begin with `input_`
 - You must add it in two places in the dashboard definition
 - 1 Define the input stanza in the dashboard definition inputs section
 - 2 List the input unique ID in the globalInputs area of the layout section



```

    ...
    "inputs": {
        "input_global_trp": {
            "type": "input.timerange",
            "options": {
                "token": "global_time",
                "defaultValue": "-24h@h,now"
            },
            "title": "Select a time range:"
        },
        "input_MTVk7TGo": {
            "options": {
                "defaultValue": "*",
                "token": "textToken"
            },
            "title": "Name of game:",
            "type": "input.text"
        }
    },
    "layout": {
        "type": "absolute",
        "options": {
            "width": 1440,
            "height": 960,
            "globalInputs": [
                "input_global_trp",
                "input_MTVk7TGo"
            ]
        },
        "description": "",
        "title": "Input - Example"
    }
}
  
```

Time Range Input Example

- Automatically added to every new dashboard
- Can have more than one
- Can customize token name
- Time token is consumed as two tokens
 - `.earliest` and `.latest` are automatically appended
 - Automatically added to the defaults section
- Works with inline and base & chain searches

The screenshot shows the 'Customized Configuration' interface in Splunk. On the right, there is a configuration panel with fields for 'Title' (set to 'Select a time:'), 'Token name' (set to 'timeTok'), and 'Default value' (set to 'Last 24 hours'). Below these is a 'Code' section containing the following JSON:

```

{
  "inputs": {
    "input_gvp3I0QZ": {
      "options": {
        "defaultValue": "-24h@h,now",
        "token": "timeTok"
      },
      "title": "Select a time:",
      "type": "input.timerange"
    }
  },
  "dataSources": {
    "ds_search_1": {
      "type": "ds.search",
      "options": {
        "query": "index=cafegames sourcetype=access_combined_cg",
        "queryParameters": {
          "earliest": "$timeTok.earliest$",
          "latest": "$timeTok.latest$"
        }
      },
      "name": "Search1"
    }
  },
  ...
  "defaults": {
    "dataSources": {
      "ds.search": {
        "options": {
          "queryParameters": {
            "latest": "$global_time.latest$",
            "earliest": "$global_time.earliest$"
          }
        }
      }
    }
  },
  ...
}
  
```

Three callout boxes with arrows point from the configuration panel to specific parts of the JSON code:

- A box labeled 'Token set by the time range input' points to the line `"token": "timeTok"`.
- A box labeled 'Time tokens consumed by this search' points to the line `"earliest": "$timeTok.earliest$"`.
- A box labeled 'Default time tokens available to all dashboard searches' points to the line `"earliest": "$global_time.earliest$"`.

Text Input Example

Enter any number or string to the input

```
{
  "dataSources": {
    "ds_search_1": {
      "type": "ds.search",
      "options": {
        "query": "index=\"cafegames\" product_name=$prodTok|s$ | timechart count by product_name useother=f"
      },
      "name": "Cafe Game Sales"
    }
  },
  "inputs": {
    "input_ta906WDq": {
      "type": "input.text",
      "title": "Enter product name:",
      "options": {
        "token": "prodTok", ←
        "defaultValue": "*"
      }
    }
  },
  ...
  "layout": {
    "globalInputs": [
      "input_global_trp",
      "input_ta906WDq"
    ]
  },
  "title": "Text Input Example",
  ...
}
```

In the search, use \$ as a delimiter and optionally, a token filter

The chart displays the count of products over time, with the y-axis ranging from 0 to 200 and the x-axis showing dates from July 20 to July 27, 2023. The legend includes: Benign Space Debris (purple), Marshmallow Missiles (pink), SIM Cubicle (teal), Blade Hopper (blue), Orville the Wolverine (magenta), Dragon Race (cyan), Puppies vs. Zombies (dark purple), Dream Crusher (yellow), and Running With Scissors (dark blue).

Configuration

Display: Above canvas

Title: Text Input

Token name: prodTok

Default value: *

Code

```

1  {
2    "type": "input.text",
3    "title": "Text Input",
4    "options": {
5      "token": "prodTok",
6      "defaultValue": "*"
7    }
8  }

```

Input ID: input_ta906WDq

Input IDs must start with "input_" and contain only letters, numbers, dash, and underscore

Number Input Example

- Ensures users only enter numbers
- Supports decimals and negative numbers
- Useful when passing a token to a search that requires a numerical argument
- Available options:
 - defaultValue
 - token
 - min: minimum number available
 - max: maximum number available
 - step: interval for the up / down arrow

Configuration

Display: Above canvas

Title: Enter a dollar amount:

Token name: numTok

Min	Max	Step
100	e.g. 100	2

Default value: 100

Code

```

1 {
2   "options": {
3     "defaultValue": 100,
4     "token": "numTok",
5     "min": 100,
6     "step": 2
7   },
8   "title": "Enter a dollar amount:",
9   "type": "input.number"
10 }

```

Input ID: input_V29BTZF3

Input IDs must start with "input_" and contain only letters, numbers, dash, and underscore

Dropdown Menu Input

- Populate dropdown inputs using static values or dynamically with search results
 - Up to 1,000 items in the menu
- Requires defining key/value pairs
 - Label appears in the menu
 - Value is passed as the token

Static Value Example

The screenshot illustrates the configuration of a dropdown menu input. On the left, a list of input types includes 'Dropdown' (selected and highlighted with an orange border), 'Multiselect', 'Text', 'Number', and 'Time Range'. An arrow points from the 'Dropdown' section to a detailed view on the right. This view shows a dropdown menu titled 'Select a sales type' with options: 'All' (selected and checked), 'filter', 'All', 'Online', and 'Retail'. The configuration panel on the right is titled 'Static Value Example' and contains the following fields:

- Configuration**: Display: Above canvas, Title: Select a sales type, Token name: dd_sourcetype_tok
- Menu configuration**: Static Menu Configuration table:

Label	Value
All	*
Online	access_combin
Retail	vendor_sales

 + Add New
- Default selected values**: Choose default: First value, Value: All
- Data sources**, **Visibility**, **Code**: These sections are collapsed.

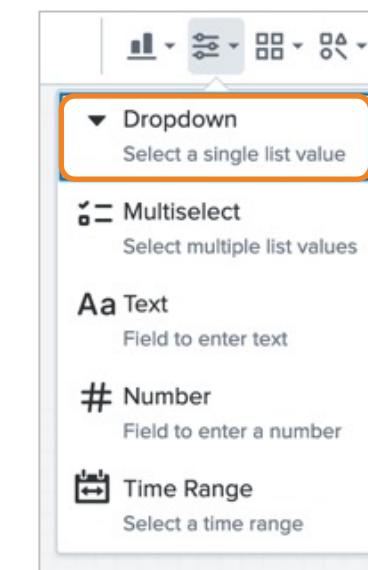
Dropdown Menu Input – Dynamic Example

1 Create a scheduled search that outputs a lookup (optional)

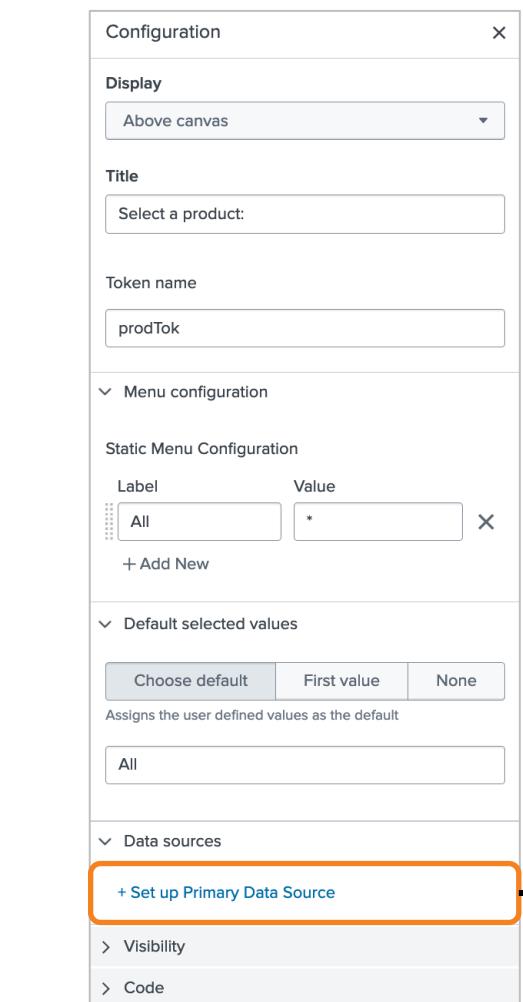
```
index=cafegames product_name="*"
| dedup product_name
| table product_name
| sort product_name
| outputlookup bcg_products
```



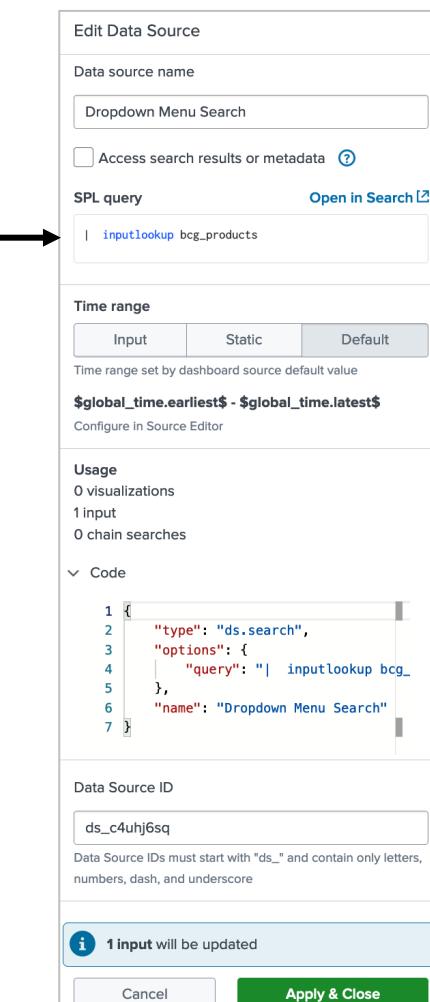
2 Add the input to the dashboard



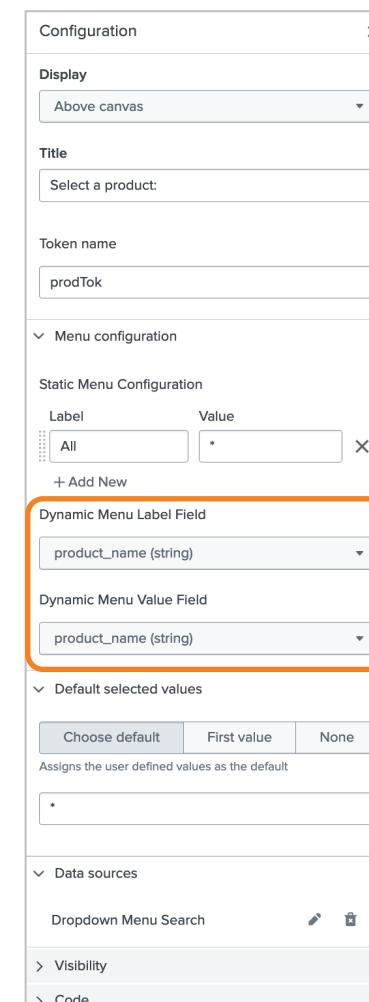
3 Customize title, token, and static options (optional)



4 Use a lookup or ad-hoc search



5 Select the label and value fields



Multiselect Input

- Select multiple options from a menu
- Populate input using static values or dynamically with search results
 - Up to 1,000 items in the menu
- Only a comma delimiter is supported
 - Use the IN operator and format your queries appropriately

```
index=cafegames sourcetype=access_combined_cg product_name IN ($prodTok|$s$)
| timechart count by product_name
```

Static Value Example

The configuration panel shows:

- Configuration**: A dropdown menu for 'Display' set to 'Above canvas'. A title 'Select a product:' and token name 'prodTok' are also defined.
- Display**: Set to 'Above canvas'.
- Title**: 'Select a product:'.
- Token name**: 'prodTok'.
- Menu configuration**: A table of static values:

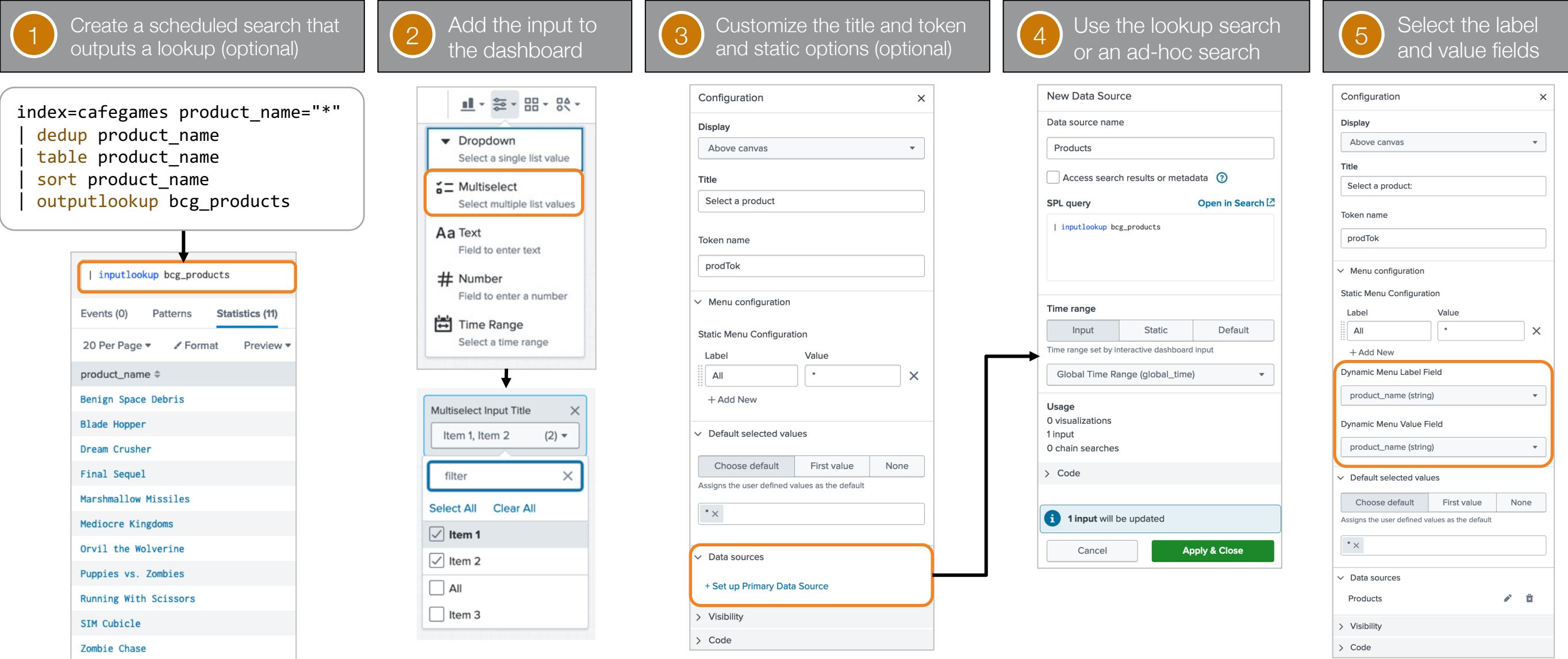
Label	Value
All	*
Benign Space I	Benign Space D
Blade Hopper	Blade Hopper
Dream Crusher	Dream Crusher
Final Sequel	Final Sequel
Marshmallow M	Marshmallow M
- Default selected values**: Options to choose a default value (Choose default, First value, None) and a list of selected values (*).
- Data sources**, **Visibility**, and **Code** sections are also present.

Multiselect Input - Static Values Example

The dashboard displays a timechart with the following data points:

Date	Blade Hopper	Dream Crusher
Thu Jul 20 2023	50	50
Fri Jul 21 2023	100	100
Sat Jul 22 2023	50	50
Mon Jul 24 2023	50	50
Tue Jul 25 2023	100	100
Wed Jul 26 2023	50	50
Thu Jul 27 2023	50	50

Multiselect Input – Dynamic Example



Submit Button

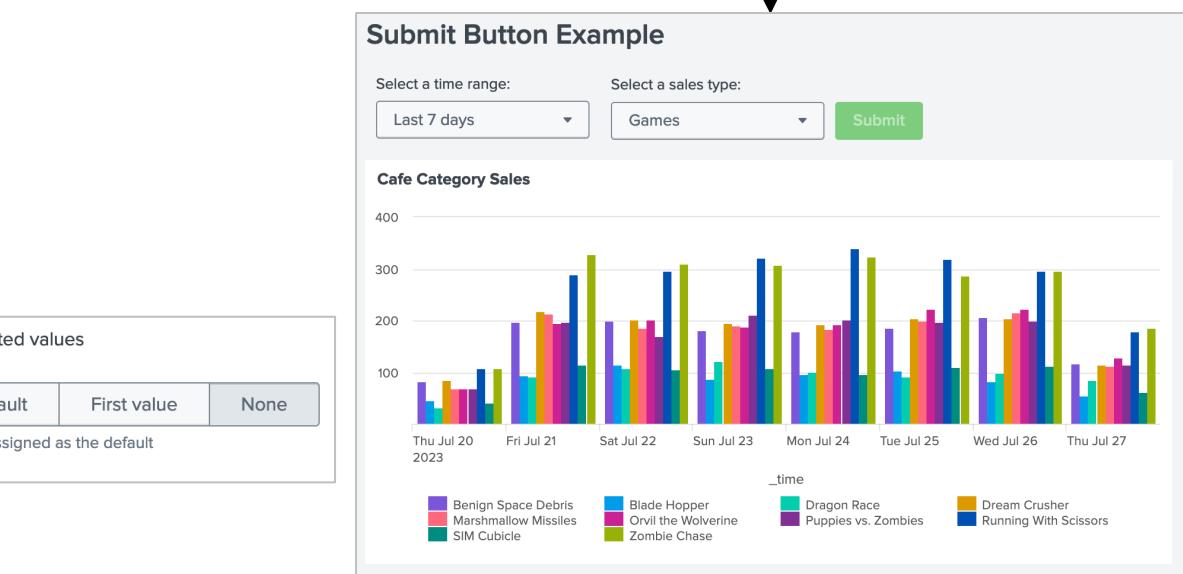
- Click to refresh a dashboard after making input selections
 - Requires adding the submitButton property to options in the layout section
- Boolean settings
 - true: Submit button click required for an input selection to take effect
 - false or not specified: refreshes when input selection is made
- Input default values determine whether visualization displays on initial dashboard load

Submit Button Example

```

Select a time range: Last 7 days
Select a sales type: Games
Submit
Cafe Category Sales
Set token value to render visualization
  
```

...
 "layout": {
 "type": "absolute",
 "options": {
 "submitButton": true,
 "height": 250
 },
 }
 ...



Cascading Inputs

- One input sets values for another input
- To each input search, add the other input tokens

– For example, in a three-input cascade:

Input1 search: add token2 and token3

Input2 search: add token1 and token3

Input3 search: add token1 and token2

Input1	Input2	Input3
All 1	All 2	All 3
Configuration	Configuration	Configuration
Display	Display	Display
Above canvas	Above canvas	Above canvas
Title	Title	Title
Dropdown1	Input2	Input3
Token name	Token name	Token name
token1	token2	token3

Input1 Search

```
| inputlookup myLookup
| search field2=$token2$ field3=$token3$
| dedup field1 | fields field1 | sort field1
```

Input2 Search

```
| inputlookup myLookup
| search field1=$token1$ field3=$token3$
| dedup field2 | fields field2 | sort field2
```

Input3 Search

```
| inputlookup myLookup
| search field1=$token1$ field2=$token2$
| dedup field3 | fields field3 | sort field3
```

Cascading Inputs – Example

Country Input Search

```
| inputlookup bcg_vendors
| search VendorStateProvince=$vStateTok|$ VendorCity=$vCityTok|$|
| dedup VendorCountry | fields VendorCountry | sort VendorCountry
```

State/Province Input Search

```
| inputlookup bcg_vendors
| search VendorCountry=$vCountryTok|$ VendorCity=$vCityTok|$|
| dedup VendorStateProvince | fields VendorStateProvince |
| sort VendorStateProvince
```

City Input Search

```
| inputlookup bcg_vendors
| search VendorCountry=$vCountryTok|$ VendorStateProvince=$vStateTok|$|
| dedup VendorCity | fields VendorCity | sort Vendor
```

The table search consumes all three tokens

```
index=sales sourcetype=vendor_sales VendorCountry=$vCountryTok$ VendorStateProvince=$vStateTok$ VendorCity=$vCityTok$
| stats sparkline(count) as Trend values(Vendor) as Vendor values(product_name) as GamesSold by VendorCountry
```

Lab Exercise 2 – Create Cascading Inputs

Description: Create a dashboard, add dropdown inputs, and make the inputs cascade.

Duration: 25 minutes

Tasks:

- Create a dashboard
- Add dropdown menu inputs
- Make inputs cascade
- Add a chart

The screenshot shows a dashboard titled "Lab2". At the top, there are three input fields: "Select a time range:" with "Last 7 days" selected, "Select a category:" with "All" selected, and "Select a game:" with "All" selected. Below these inputs are two cards. The left card is titled "Games Played" and contains a donut chart with four segments. The right card is titled "Top Scores" and contains a list of four items, each represented by a horizontal bar.

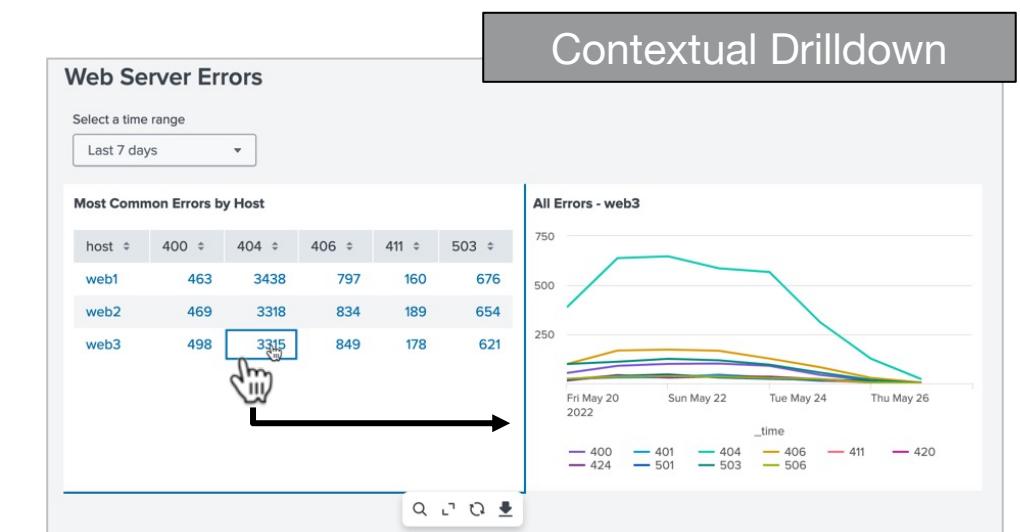
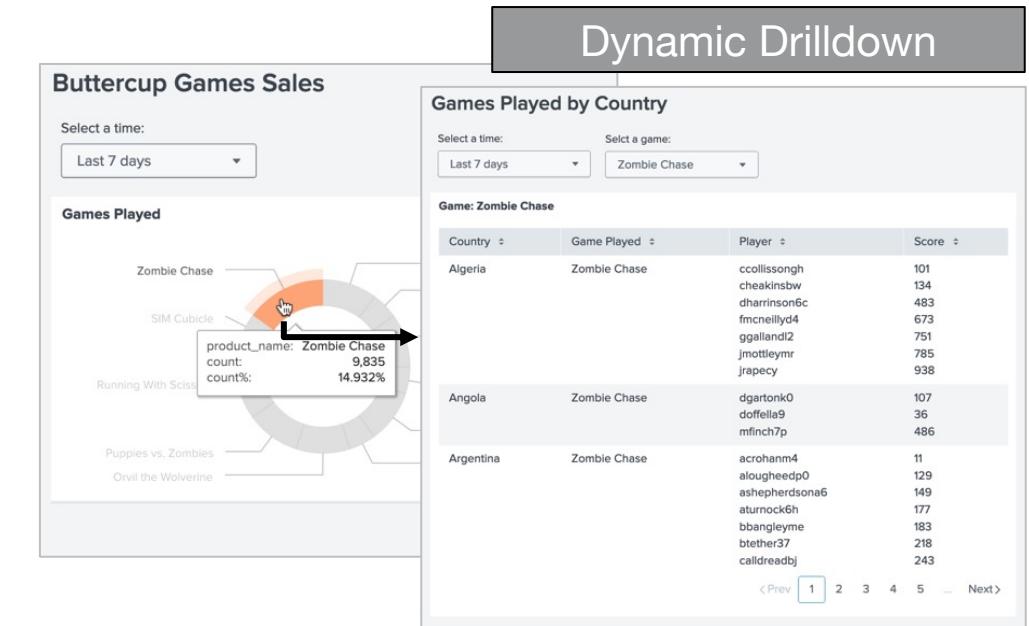
Topic 3: Adding Drilldowns

Topic Objectives

- Identify types of drilldowns
- Create a dynamic drilldown
- Create a contextual drilldown

Custom Drilldowns

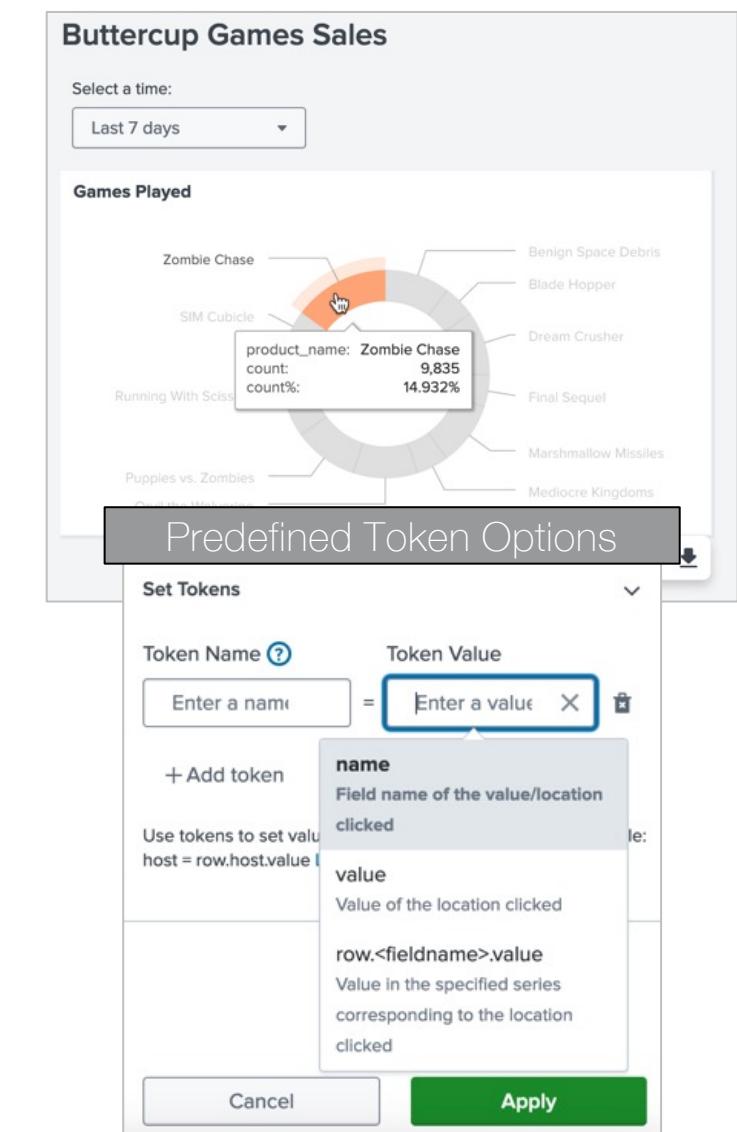
- Dynamic Drilldown
 - Link to dashboard
 - Set tokens
 - Pass values as tokens from a user click to another dashboard
 - Specify a path to a URL
 - Relative path to a dashboard in another app
 - Absolute path to an external website
- Contextual Drilldown
 - Set tokens
 - Pass values as tokens from a user click to visualizations on the same dashboard



Custom Drilldowns – Setting Tokens

- Set tokens with a visualization click
- Use predefined tokens or a static token value
 - **name** captures the field name of the value or location clicked
 - **value** captures the value of the location clicked
 - **row.<field_name>.value** captures the value in the specified series of the location clicked
- Use **\$...\$** delimiters in the token value to pass the value of an existing token
 - For example, pass the time range using:

Token Name	Token Value
<timeRangeToken>.earliest	\$<timeRangeToken>.earliest\$
<timeRangeToken>.latest	\$<timeRangeToken>.latest\$



Setting Tokens – Example

Set a token with one visualization

Use that token in the search used by others

The screenshot illustrates the process of setting a token and using it across different visualizations.

Configuration Panel:

- On click:** Set tokens
- Tokens are used to configure interactivity in the dashboard.**
- For example:** host = row.host.value
- Set token (hostTok):** Use predefined token
 - Token name:** hostTok
 - Token value:** row.host.value
 - Default value:** *
- + Set another token**

Web Server Errors Visualization:

Select a time range: Last 7 days

host	200	400	404	406	503
web1	3696	198	1307	325	228
web2	3679	201	1346	321	242
web3	3737	203	1349	329	254

A mouse cursor is hovering over the value 1349 for host web3 under the 404 column.

All Errors - web3 Timechart:

The chart shows the count of errors over time (Wed Jul 26 to Tue Aug 1, 2023) for various error codes. The Y-axis ranges from 0 to 300. The X-axis shows dates: Wed Jul 26, Fri Jul 28, Sun Jul 30, Tue Aug 1.

Time	400	404	406	409	415	416	503	504	508	510
Wed Jul 26	~10	~250	~10	~10	~10	~10	~10	~10	~10	~10
Fri Jul 28	~10	~180	~10	~10	~10	~10	~10	~10	~10	~10
Sun Jul 30	~10	~190	~10	~10	~10	~10	~10	~10	~10	~10
Tue Aug 1	~10	~170	~10	~10	~10	~10	~10	~10	~10	~10

Edit Data Source Panel:

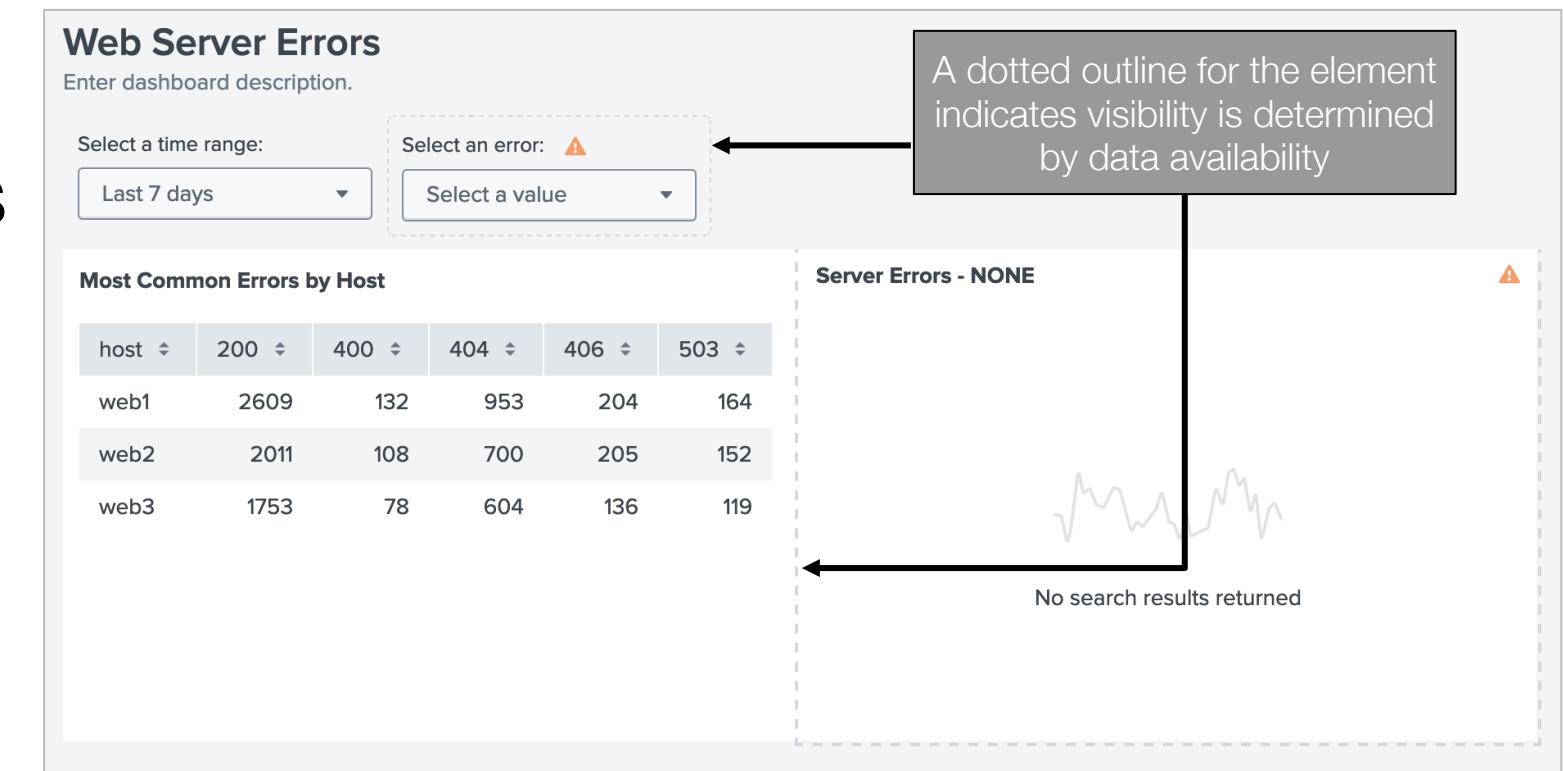
- Data source name:** All Errors
- Access search results or metadata:**
- SPL query:**

```
index=cafegames sourcetype=access_combined_cg host=$hostTok$ status>399 | timechart count by status useother=f
```

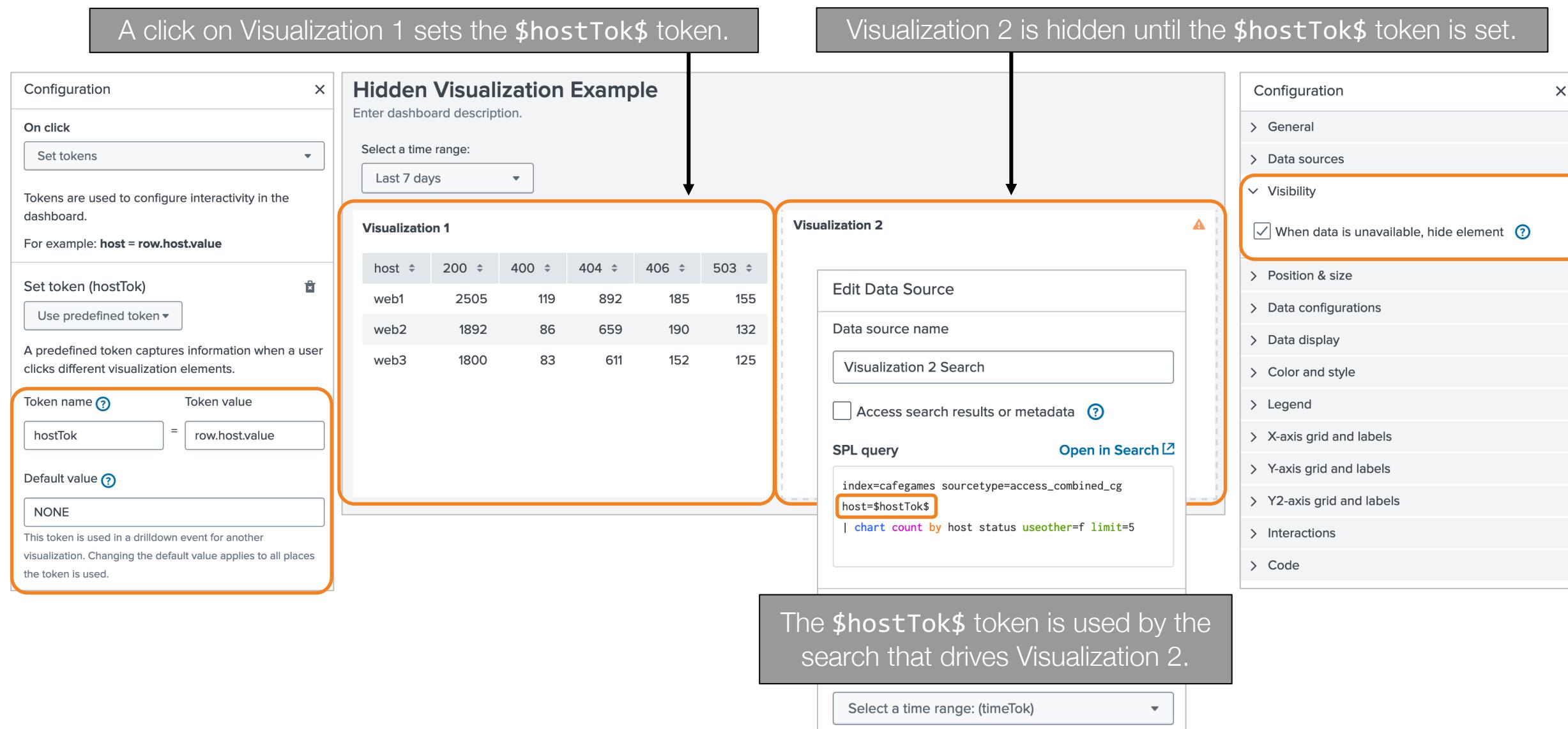
[Open in Search](#)
- Time range:**
 - Input:** Time range set by interactive dashboard input
 - Static:**
 - Default:**
- Select a time range:** (timeTok)

Hiding Elements

- Hide visualizations and inputs based on the existence of search results
- Requires absolute layout
- Available for the following:
 - Charts
 - Icons
 - Shapes
 - Dropdown input
 - Multiselect input



Hiding Elements – Visualization Example



Hiding Elements – Visualization Example (cont.)

Hide visualization 2 again by having a user click reset the token to the default value

Hidden Visualization Example

Enter dashboard description.

Select a time range:

Last 7 days

Visualization 1

host	200	400	404	406	503
web1	2505	119	892	185	155
web2	1892	86	659	190	132
web3	1800	83	611	152	125

Visualization 2

No search results returned

Configuration

On click

Set tokens

Tokens are used to configure interactivity in the dashboard.

For example: host = row.host.value

Set token (hostTok)

Enter static value ▾

A static token value is set to a specific string that does not change.

Token name ? Token value

hostTok = NONE

Default value ?

NONE

This token is used in a drilldown event for another visualization. Changing the default value applies to all places the token is used.

+ Set another token

Cancel Apply

Hiding Elements – Input Example

My Dropdown Input is hidden until the `$hostTok$` token is set by a click on Visualization 1

Visualization 2 is hidden until a selection from My Dropdown Input sets the `$statusTok$` token consumed by Visualization 2.

Hidden Input Example

Visualization 1

Visualization 2

Visualizations

Host Status

Host	200	400	404	406	503
web1	2480	119	883	183	152
web2	Configuration	132			
web3	On click	123			

No search results returned

Set token (hostTok)

Use predefined token ▾

A predefined token captures information when a user clicks different visualization elements.

Token name [?](#) Token value

hostTok = row.host.value

Default value [?](#)

NONE

This token is used in a dropdown event for another visualization. Changing the default value applies to all places the token is used.

Dynamic Drilldown – Origin Dashboard

The screenshot illustrates the configuration of a dynamic drilldown between two dashboards:

- Source Dashboard (Left): Buttercup Games Sales**
 - Shows a donut chart titled "Product Sales" with categories: Running with Scissors, Benign Space Debris, Orvil the Wolverine, Dream Crusher, Final Sequel, Sim Cubicle, Puppies vs. Zombies, Marshmallow Missles, Blade Hopper, Zombie Chase, and Running with Scissors.
 - Configuration panel shows "Interactions" expanded, with "Link to dashboard" selected under "On Click".
- Destination Dashboard (Right): Product Sales Trends by Country**
 - Configuration panel shows "Set Tokens" expanded, with "Link to dashboard" selected under "On Click".
 - Configuration panel shows "Select an App" set to "Buttercup Games Sales Team" and "Select a Dashboard" set to "Product Sales Trends by Country".
 - Configuration panel shows "Set Tokens" expanded, with tokens defined as follows:
 - prodTok = row.product
 - global_time. = \$global_time
 - global_time. = \$global_time
 - Buttons: "Cancel" and "Apply" are at the bottom right.

Annotations explain the process:

- A callout points to the "Link to dashboard" action in both destination configurations, with the text: "Match these tokens on the destination dashboard".
- A callout points to the "Link to dashboard" token definition in the destination configuration, with the text: "To these tokens on the origin dashboard".
- A callout points to the "Apply" button in the destination configuration, with the text: "6".
- A callout points to the "name" section of the token options table, with the text: "Origin dashboard token options: name".

name	Field name of the value/location clicked
value	Value of the location clicked
row.<fieldname>.value	Value in the specified series corresponding to the location clicked
=	Enter a value

Dynamic Drilldown – Destination Dashboard

If passing a time input token match their token names on both dashboards

Configuration

Display

Title

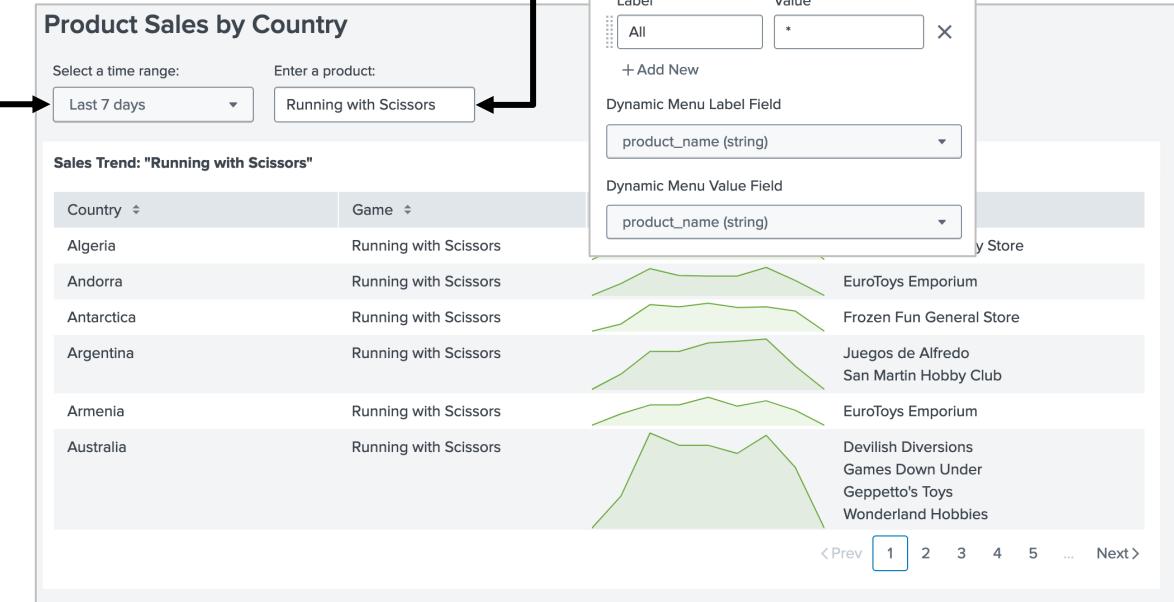
Select a time range:

Token name

timeTok

Default value

Last 7 days



Can use the source token on destination inputs

Configuration

Display

Title

Select a product:

Token name

prodTok

Menu configuration

Static Menu Configuration

Label	Value
All	*

+ Add New

Dynamic Menu Label Field

product_name (string)

Dynamic Menu Value Field

product_name (string)

Add the source token to the search used by the visualization on the destination dashboard

Edit Data Source

Data source name

Product Sales

Access search results or metadata

SPL query Open in Search

```
index=sales sourcetype=vendor_sales
product_name=$prodTok|$s$ 3
| stats sparkline(count) as Trend values(Vendor) as
  Vendors by VendorCountry product_name
| rename VendorCountry as Country product_name as
  Game | sort Country
```

Time range

Input Static Default

Time range set by interactive dashboard input

Select a time range: (timeTok)

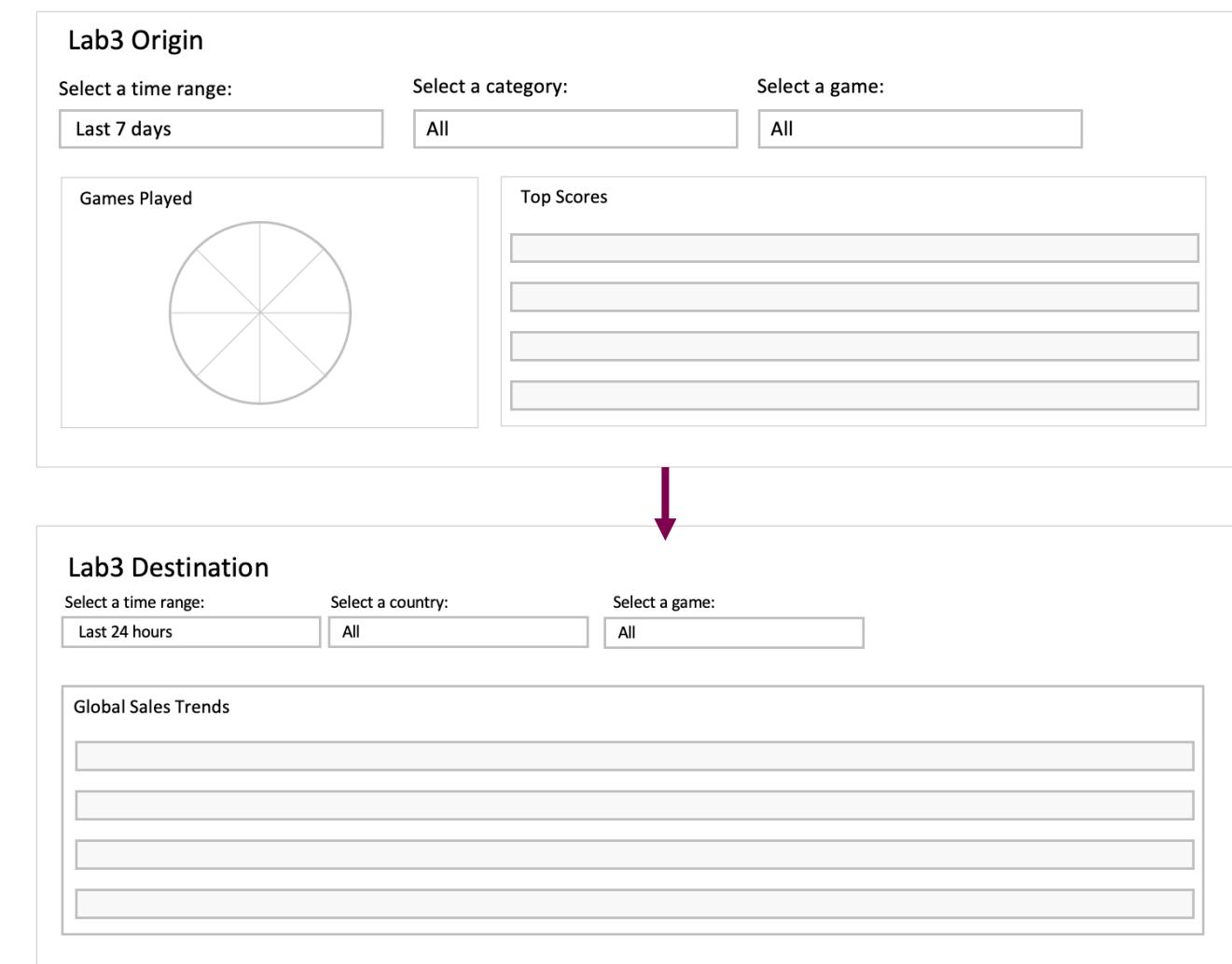
Lab Exercise 3 – Create a Dynamic Drilldown

Description: Create a dynamic drilldown from the Games Played pie chart to a new form

Duration: 25 minutes

Tasks:

- Create a destination form
- Configure the data source
- Add dropdown inputs
- Clone a dashboard
- Add a drilldown



Topic 4: Dynamic Visualizations

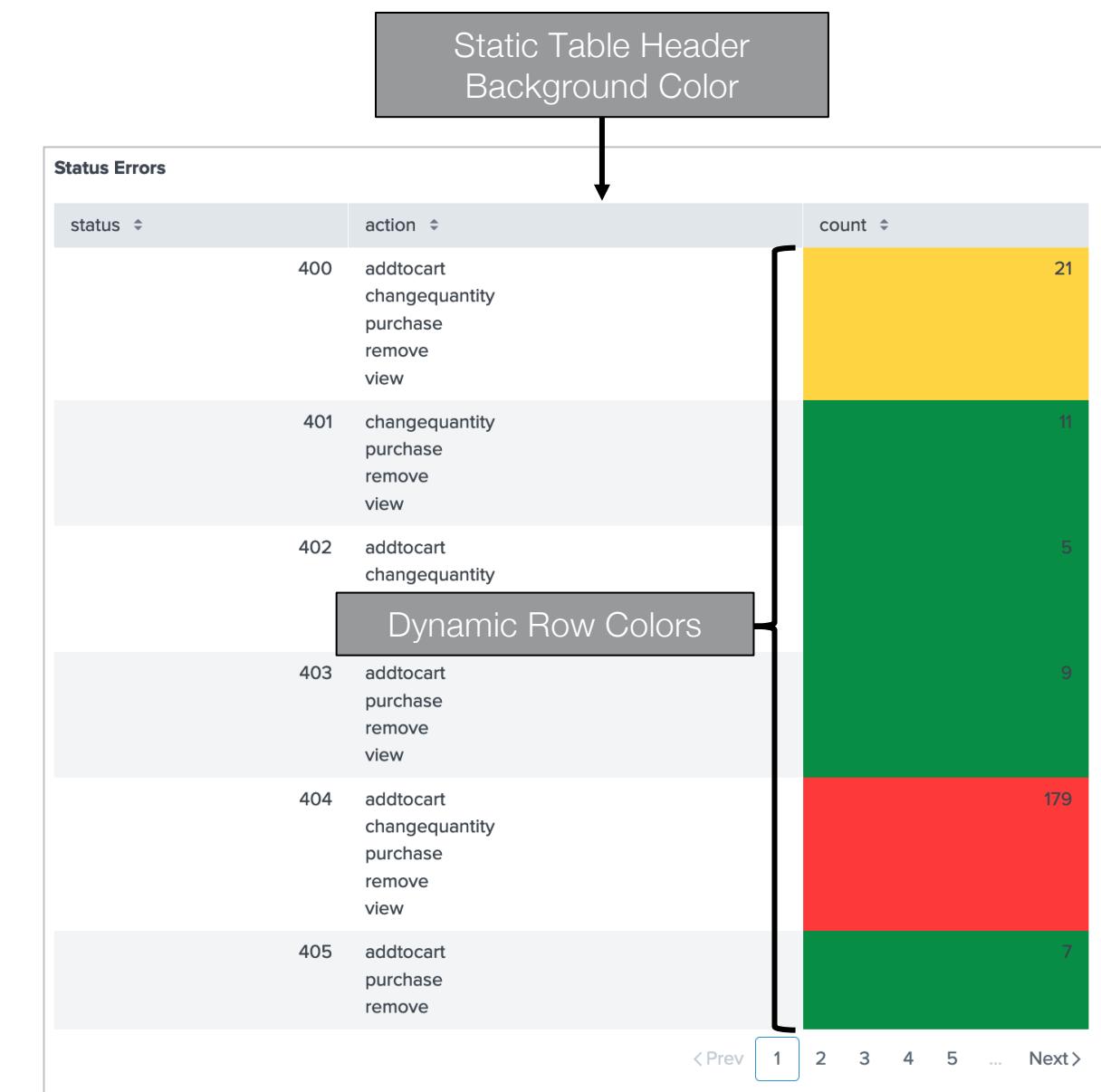
Topic Objectives

- Describe static and dynamic coloring
- Add dynamic coloring to a visualization

Visualization Coloring

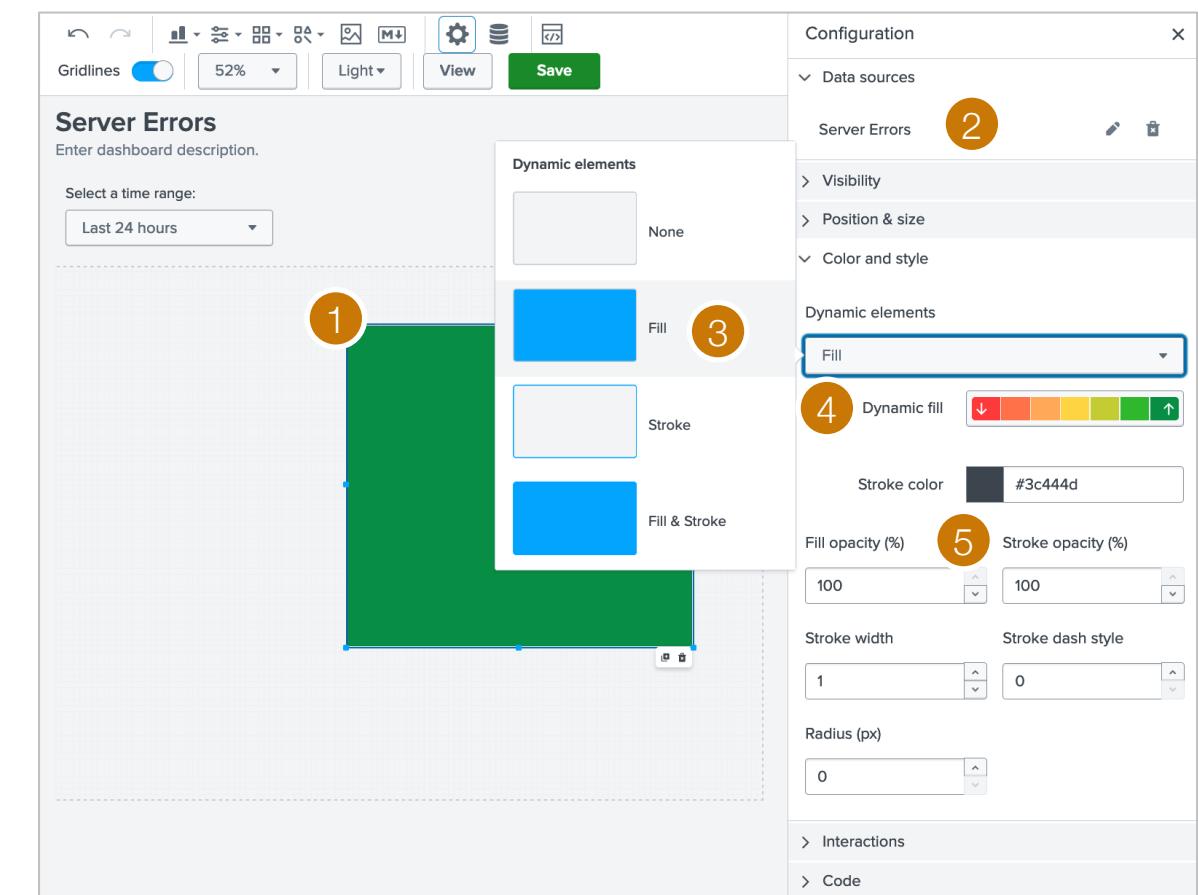
- Static: color part of the visualization
 - Fill, lines, rows, and more
- Dynamic: use search results to dynamically color visualizations
 - Color values by trend, value, and more
 - Fill, lines, columns, rows, values, text, and more
 - Available for these visualizations:

• Icon	• Single Value
• Punchcard	• Single Value Radial
• Marker Gauge	• Single Value Icon
• Table	• Sankey
• Shape	• Choropleth SVG



Dynamic Coloring – Shape Example

- To enable dynamic coloring:
 - 1 Select the shape
 - 2 Set up the Primary Data Source
 - 3 Select an option in Dynamic Elements
 - 4 Set threshold values and colors
 - 5 Set fill and stroke opacity



Dynamic Coloring – Table Example

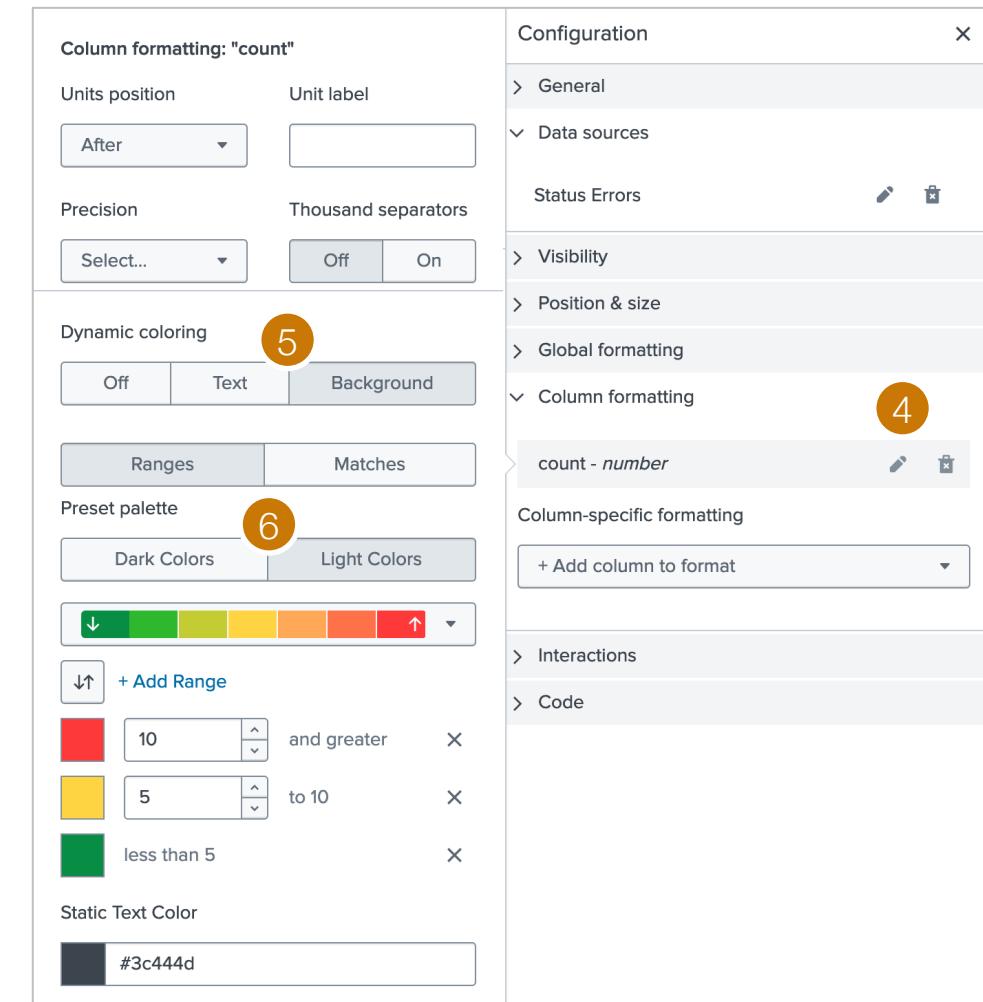
- To enable dynamic coloring:
 - 1 Select the table
 - 2 Set up the Primary Data Source
 - 3 Select the column to be dynamically colored

The screenshot shows the Splunk Dashboard Editor interface. On the left is a dashboard preview titled "Server Status" with a table titled "Status Errors". The table has columns: host, status, action, and count. The data rows show various errors from host "web1" with status codes 400 through 405 and their corresponding actions and counts. The "Configuration" pane on the right is open, showing the "Column-specific formatting" section. A dropdown menu "host - string" is expanded, and the "status - number" field is selected. Three numbered callouts point to specific elements: 1 points to the table header, 2 points to the "Status Errors" table in the dashboard preview, and 3 points to the "status - number" field in the configuration pane.

host	status	action	count
web1	400	addtocart purchase remove view	5
web1	401	purchase view	3
web1	402	changequantity remove view	3
web1	403	purchase	3
web1	404	addtocart changequantity purchase remove view	55
web1	405	purchase	2

Dynamic Coloring – Table Example (cont.)

- 4 Click the pencil icon beside the column selected
- 5 Select text or background of the column to be colored
- 6 Choose between dark and light colors



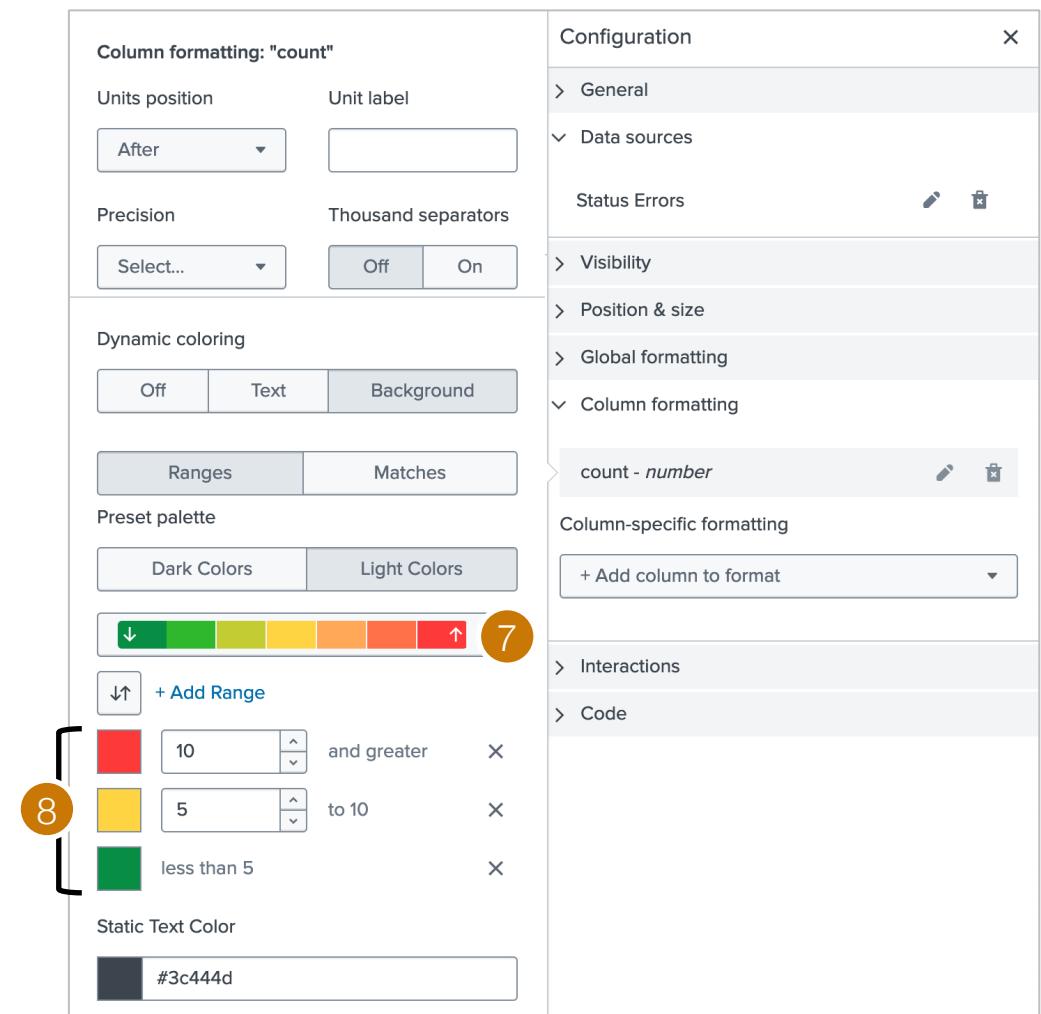
Dynamic Coloring – Table Example (cont.)

7 Select a preset palette

- If needed, switch the range order of the colors

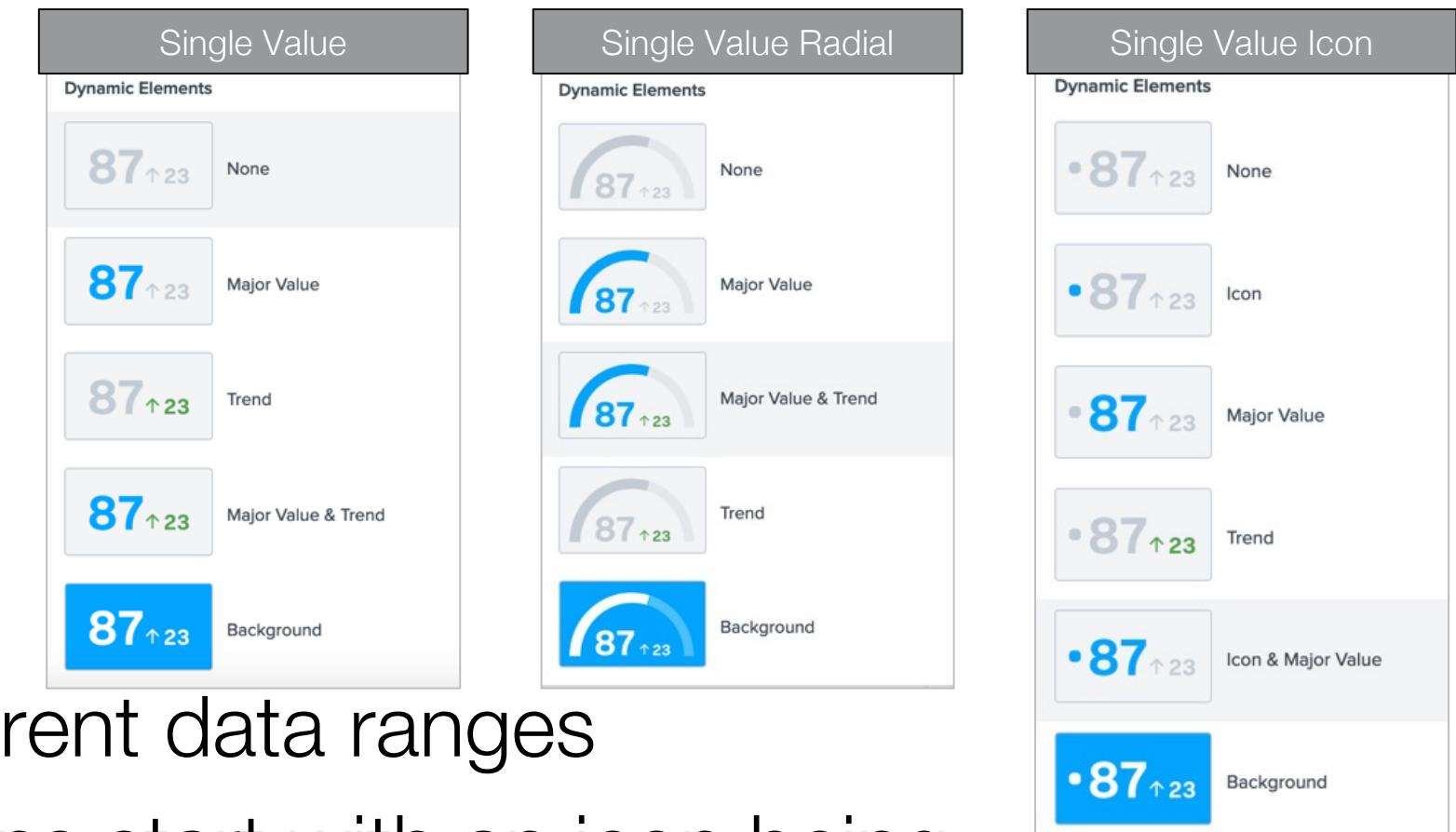
8 Specify the range values

- More than seven can be added using the source editor
- If needed, change the range colors



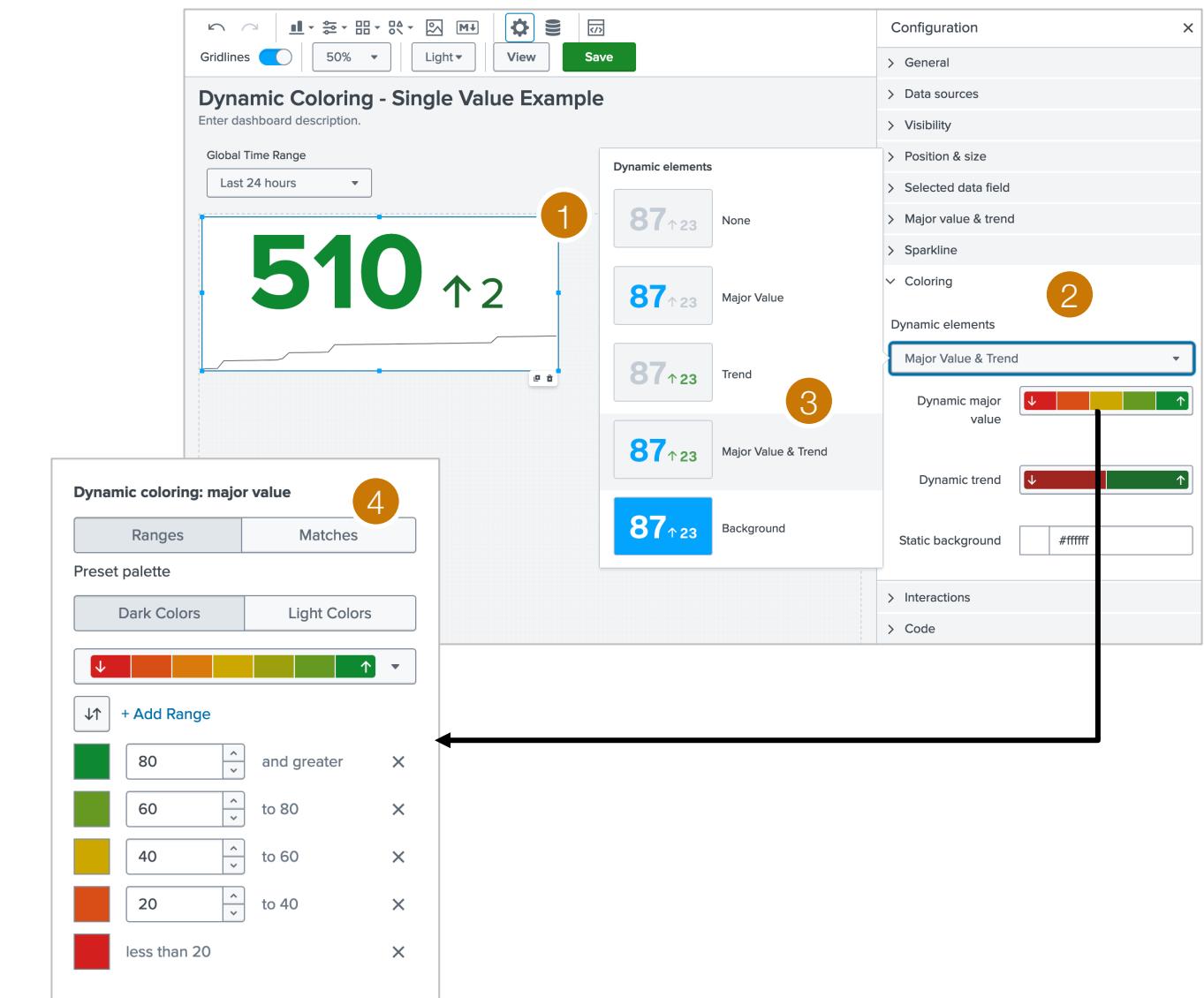
Dynamic Coloring – Single Value

- Single Value and Single Value Radial
 - Major value, trend, and background
- Single Value Icon
 - Icon, major value, trend, and background
- Set color thresholds for different data ranges
- Single value icon visualizations start with an icon being added to a layout and then linked to a data source



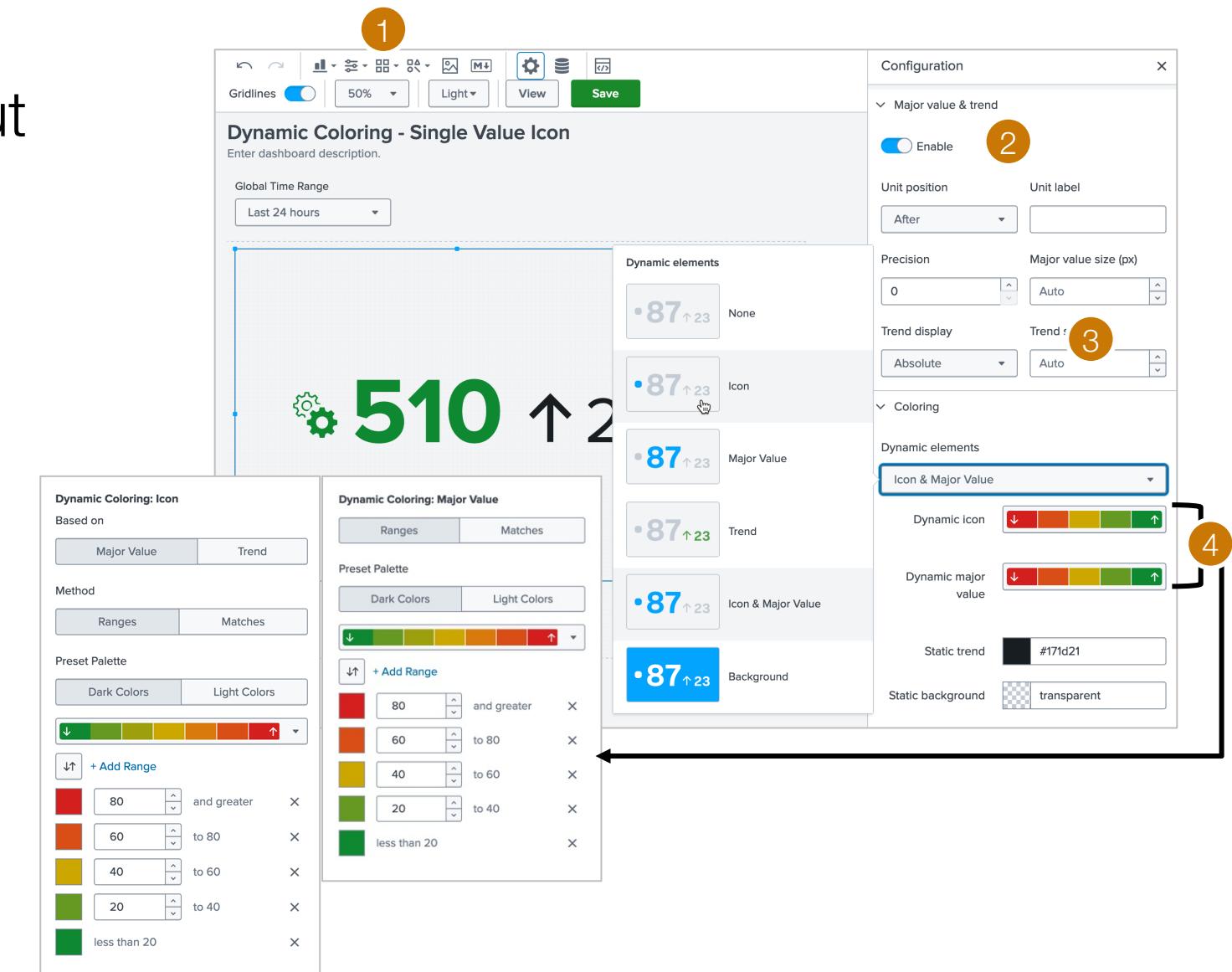
Dynamic Coloring – Single Value Example

- To enable dynamic coloring:
 - 1 Add a single value visualization to an absolute layout
 - 2 Scroll to the Coloring section
 - 3 In the Dynamic elements menu, select which elements should be dynamically colored
 - 4 Adjust the color palettes and ranges or accept the defaults



Dynamic Coloring – Single Value Icon Example

- To enable dynamic coloring:
 - 1 Add an icon to an absolute layout
 - 2 Enable Major value & trend
 - 3 In the Dynamic Elements menu, select which elements should be dynamically colored
 - 4 Adjust the color palettes and ranges or accept the defaults



Dynamic Options Example – Source Editor

Visualization Formatting

```

1  "visualizations": {
2    "viz_C8fYK7RH": {
3      "type": "splunk.table",
4      "options": {
5        "columnFormat": {
6          "count": {
7            "data": "> table | seriesByName(\"count\") | formatByType(countColumnFormatEditorConfig)",
8            "rowColors": "> table | seriesByName('count') | pick(countRowColorsEditorConfig)",
9            "rowBackgroundColors": "> table | seriesByName(\"count\") | rangeValue(countRowBackgroundColorsEditorConfig)"
10           }
11         }
12       },
13     "context": {
14       "countColumnFormatEditorConfig": {
15         "number": {
16           "thousandSeparated": false,
17           "unitPosition": "after"
18         }
19       },
20       "countRowColorsEditorConfig": [
21         "#3c444d"
22       ],
23       "countRowBackgroundColorsEditorConfig": [
24         {
25           "value": "#669922",
26           "to": 10
27         },
28         {
29           "value": "#CBA700",
30           "from": 10,
31           "to": 30
32         },
33         {
34           "value": "#D41F1F",
35           "from": 30
36         }
37       ]
38     }
39   }
40

```

Object Properties

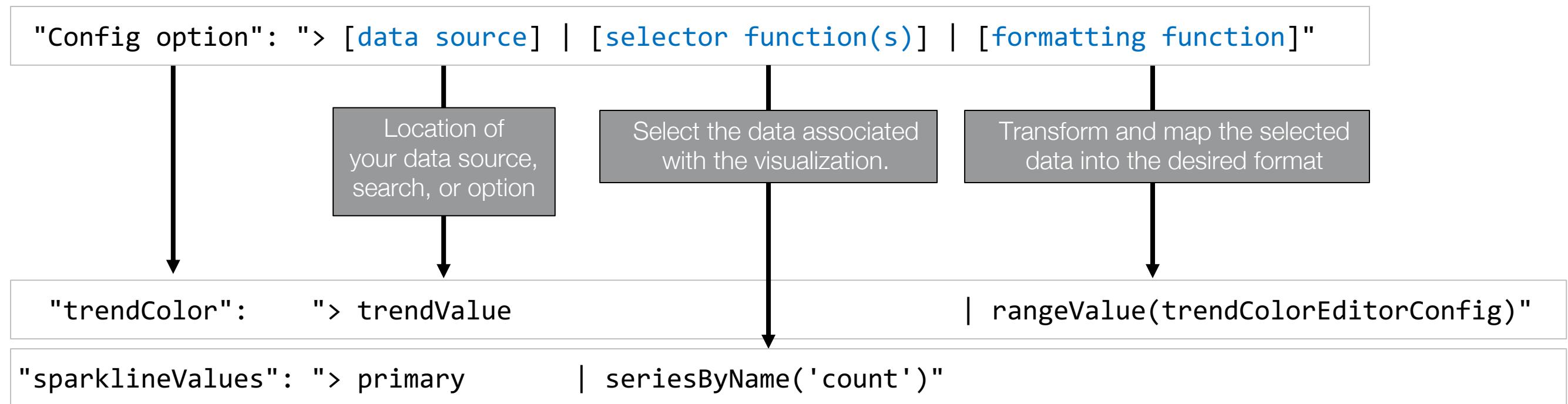
Column of query results to use

Colors and ranges for each color

status	action	count
400	addtocart purchase remove view	28
401	changequantity purchase remove	6
402	addtocart purchase view	6
403	purchase view	6
404	addtocart changequantity purchase remove view	163
405	changequantity purchase view	7

Dynamic Options – Source Editor Only

- Some options are only available by editing the source code
- Add these in the options section of your visualization
 - Items listed configure the illustrative specifics
 - Use dynamic options syntax (DOS) structure



Dynamic Options – Source Editor Only (cont.)

The diagram illustrates the configuration of a single-value visualization in the Splunk Source Editor. It shows the JSON configuration on the left and its corresponding visual representation on the right.

JSON Configuration:

```

... "visualizations": {
    "viz_ZCtqeNe7": {
        "type": "splunk.singlevalue",
        "options": {
            "majorColor": "> majorValue | rangeValue(majorColorEditorConfig)",
            "trendColor": "> trendValue | rangeValue(trendColorEditorConfig)",
            "showSparklineAreaGraph": true,
            "sparklineStrokeColor": "> majorColor"
        },
        "dataSources": {
            "primary": "ds_Hn6PPItg"
        },
        "context": {
            "majorColorEditorConfig": [
                {
                    "value": "#D41F1F",
                    "to": 20
                },
                {
                    "value": "#D94E17",
                    "from": 20,
                    "to": 80
                },
                {
                    "from": 80,
                    "value": "#118832"
                }
            ],
            "trendColorEditorConfig": [
                {
                    "to": 0,
                    "value": "#9E2520"
                },
                {
                    "from": 0,
                    "value": "#1C6B2D"
                }
            ]
        }
    }
},
...
  
```

Visual Representation:

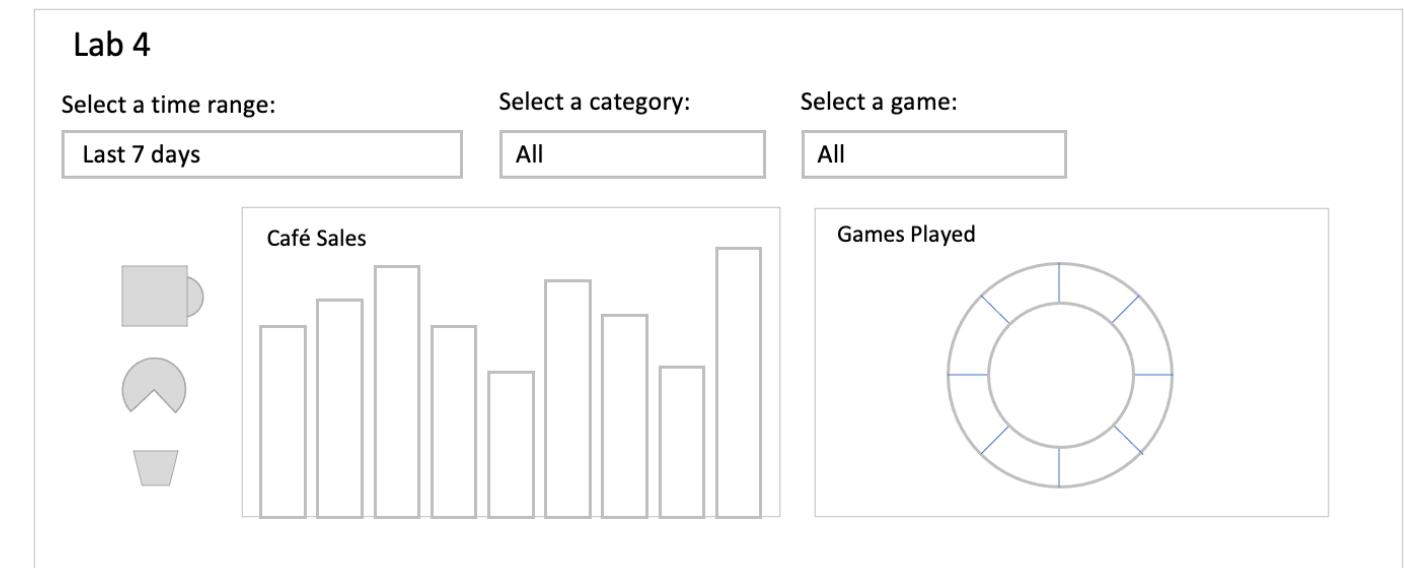
A large green number **558** is displayed above a green sparkline area graph. A red arrow points down from the number to the value **-6,279** in the bottom right corner.

Annotations:

- "Config option": "> [data source] | [formatting function]"
- "majorColor": "> majorValue | rangeValue(majorColorEditorConfig)" **Example**
- "trendColor": "> trendValue | rangeValue(trendColorEditorConfig)" **Example**
- "Config option": [setting]
- "showSparklineAreaGraph": **true** **Example**
- "Config option": "> [data source]"
- "sparklineStrokeColor": "> majorColor" **Example**

Lab 4 – Add Dynamic Coloring

- **Description:** Clone a dashboard, delete a visualization, customize a visualization, add icons and dynamic coloring.
- **Duration:** 25 minutes
- **Tasks:**
 - Clone a dashboard
 - Revise the data sources
 - Revise the pie chart
 - Add a single value visualization
 - Add a column chart
 - Add icons
 - Add dynamic coloring



Summary

Wrap Up



- You should now be able to:
 - Create user inputs
 - Define token syntax
 - Create dynamic inputs
 - Build cascading inputs
 - Create a dynamic drilldown
 - Set tokens
 - Use dynamic coloring



Documentation

Search Docs

Search

Topic 1: Using Tokens

- [Use inputs and tokens to make dashboards dynamic](#)

Topic 2: Adding Inputs

- [Use Drilldown for Dashboard Interactivity](#)
- [Link to a dashboard](#)
- [Predefined Drilldown Tokens](#)

Topic 3: Adding Drilldowns

- [Object options and defaults reference](#)
- [Advanced dynamic options syntax](#)
- [The source code stanza of a visualization](#)

Topic 4: Dynamic Visualizations

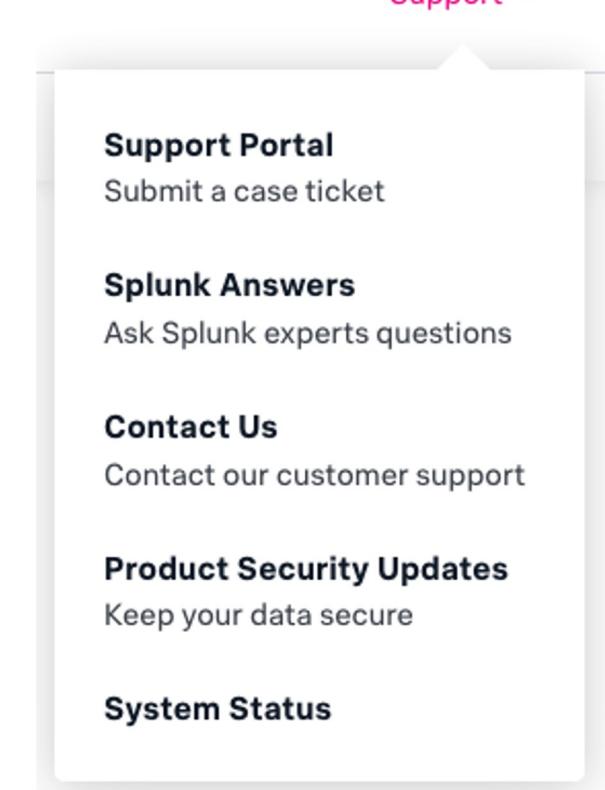
- [How the dashboard definition is structured in the source editor](#)

References

- Splunk Community Portal – community.splunk.com
 - Answers
 - Discussions
 - Splunk Trust
 - User Groups
 - Ideas
- Splunk Blogs – splunk.com/blog/
- Splunk Apps – splunkbase.com
 - Apps
 - Curated Collections
- Splunk Docs on Twitter – twitter.com/splunkdocs
- Splunk Dev on Twitter – twitter.com/splunkdev
- Splunk on Slack – splk.it/slack
- .conf – conf.splunk.com

Support Programs

- Web
 - Documentation: dev.splunk.com and docs.splunk.com
 - Wiki: wiki.splunk.com
- Splunk Lantern: Guidance from Splunk experts
 - lantern.splunk.com
- Global Support: Support for critical issues, a dedicated resource to manage your account – 24 x 7 x 365
 - Web: splunk.com/index.php/submit_issue
- Enterprise, Cloud, ITSI, Security Support
 - Web: splunk.com/en_us/about-splunk/contact-us.html#tabs/customersupport
 - Phone: (855) SPLUNK-S or (855) 775-8657



Learning Paths

Search Expert - Recommended Courses

Free eLearning courses are in blue and courses with an * are present in both learning paths.

- Introduction to Splunk *
- Using Fields *
- Scheduling Reports and Alerts
- Visualizations
- Statistical Processing
- Working with Time
- Comparing Values
- Result Modification
- Leveraging Lookups and Subsearches
- Correlation Analysis
- Search Under the Hood
- Multivalue Fields
- Search Optimization *

Learning Paths (cont.)

Knowledge Manager - Recommended Courses

Free eLearning courses are in blue and courses with an * are present in both learning paths.

- Introduction to Splunk *
- Using Fields *
- Introduction to Knowledge Objects
- Creating Knowledge Objects
- Creating Field Extractions
- Enriching Data with Lookups
- Data Models
- Introduction to Dashboards
- Dynamic Dashboards
- Creating Maps
- Search Optimization *

Splunk Mobile

- Free app available to all Splunk Cloud and Splunk Enterprise customers
- Analyze data and receive actionable alerts on-the-go with mobile-friendly dashboards
- iOS and Android
- See the [Product Brief](#)

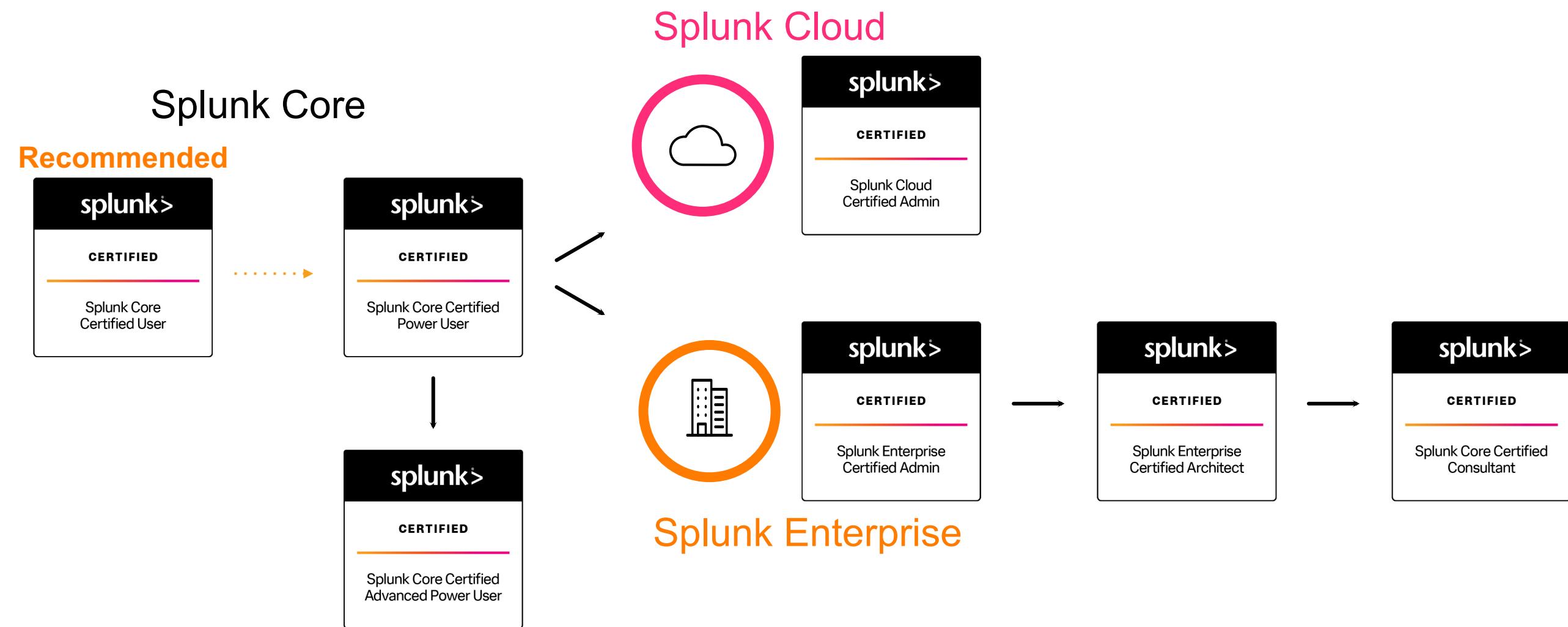


Splunk Certification

Offerings and Requirements

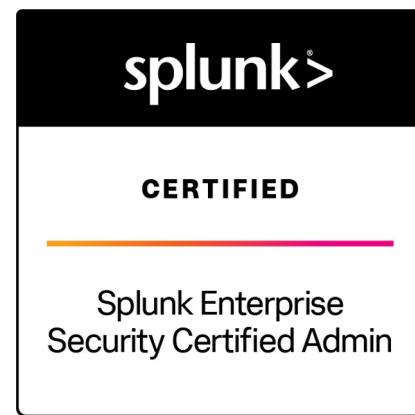
Splunk Core and Beyond

Regardless of which Splunk product you use, it all starts with Splunk Core



App-Specific Offerings

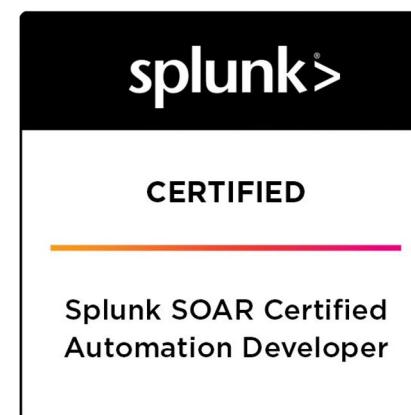
For Splunk Add-Ons



ES
Administration



ITSI
Administration



SOAR
Automation
Developer

Splunk Core Certified User

This entry-level certification demonstrates an individual's basic ability to use Splunk software



Prerequisite Certification(s):

- None

Prerequisite Course(s):

- None

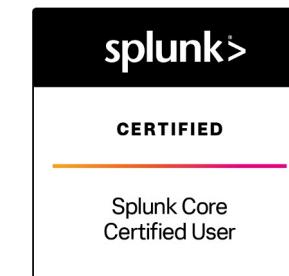
Splunk Core Certified User Exam

Time to study! We suggest candidates looking to prepare for this exam complete Fundamentals 1 **or** the following courses:

- Intro to Splunk
- Using Fields
- Scheduling Reports and Alerts
- Visualizations
- Statistical Processing
- Working with Time
- Leveraging Lookups and Subsearches
- Search Optimization
- Enriching Data with Lookups
- Data Models

See [here](#) for registration assistance.

Congratulations! You are a...



Recommended Next Step

- Splunk Core Certified Power User

Splunk Core Certified Power User

This entry-level certification demonstrates an individual's foundational competence of Splunk's core software



Prerequisite Certification(s):

- None

Prerequisite Course(s):

- None

Splunk Core Certified Power User Exam

Time to [study](#)! We suggest candidates looking to prepare for this exam complete Fundamentals 2 **or** the following courses:

- Visualizations
- Statistical Processing
- Working with Time
- Comparing Values
- Result Modification
- Correlation Analysis
- Search Under the Hood
- Introduction to Knowledge Objects
- Creating Knowledge Objects
- Creating Field Extractions
- Data Models
- Creating Maps

See [here](#) for registration assistance.

Congratulations! You are a...



Recommended Next Steps

- [Splunk Core Certified Advanced Power User](#)
- [Splunk Enterprise Certified Admin](#)
- [Splunk Cloud Certified Admin](#)

Splunk Core Certified Advanced Power User

This certification demonstrates an individual's ability to generate complex searches, reports, and dashboards with Splunk's core software to get the most out of their data



Prerequisite Certification(s):

- [Splunk Core Certified Power User](#)

Prerequisite Course(s):

- None

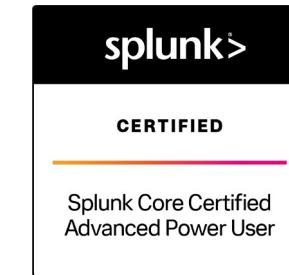
Splunk Core Certified Advanced Power User Exam

Time to [study](#)! We suggest candidates looking to prepare for this exam complete Fundamentals 3, Creating Dashboards, and Advanced Searching & Reporting **or** the following courses:

- Using Fields
- Working with Time
- Comparing Values
- Result Modification
- Leveraging Lookups and Subsearches
- Correlation Analysis
- Search Under the Hood
- Multivalue Fields
- Search Optimization
- Creating Field Extractions
- Enriching Data with Lookups
- Data Models
- Creating Maps
- Introduction to Dashboards
- Dynamic Dashboards

See [here](#) for registration assistance.

Congratulations! You are a...



Recommended Next Steps

- [Splunk Enterprise Certified Admin](#)
- [Splunk Cloud Certified Admin](#)

Splunk Cloud Certified Admin

This certification demonstrates an individual's ability to support the day-to-day administration and health of a Splunk Cloud environment



Prerequisite Certification(s):

- [Splunk Core Certified Power User](#)

Prerequisite Course(s):

- None

Splunk Cloud Certified Admin Exam

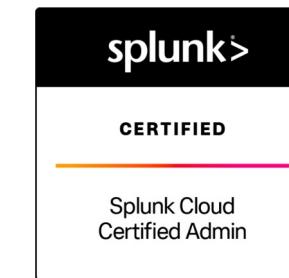
Time to [study](#)! We suggest candidates looking to prepare for this exam complete **either** the Splunk Cloud Administration **or** the Transitioning to Splunk Cloud course.

Both courses will equally prepare candidates for the exam, but are tailored to meet the needs of the individual based on prior Splunk experience.

Splunk Cloud Administration is designed for net-new administrators working in a Splunk Cloud environment. **Transitioning to Splunk Cloud** is for experienced Enterprise administrators looking to maximize their success in migrating to a Cloud environment.

See [here](#) for registration assistance.

Congratulations! You are a...



Splunk Enterprise Certified Admin

This certification demonstrates an individual's ability to support the day-to-day administration and health of a Splunk Enterprise environment



Prerequisite Certification(s):

- [Splunk Core Certified Power User](#)

Prerequisite Course(s):

- None

Splunk Enterprise Certified Admin Exam

Time to [study](#)! We suggest candidates looking to prepare for this exam complete the following courses:

- Splunk System Administration
- Splunk Data Administration

See [here](#) for registration assistance.

Congratulations! You are a...



Recommended Next Steps

- [Splunk Enterprise Certified Architect](#)

Splunk Certified Architect

This certification demonstrates an individual's ability to deploy, manage, and troubleshoot complex Splunk Enterprise environments



Prerequisite Certification(s):

- [Splunk Core Certified Power User](#)
- [Splunk Enterprise Certified Admin](#)

Prerequisite Course(s):

- Architecting Splunk Enterprise Deployments
- Troubleshooting Splunk Enterprise
- Splunk Cluster Administration
- Splunk Deployment Practical Lab

Splunk Enterprise Certified Architect Exam

Time to [study](#)! We **require** candidates looking to register for this exam to complete the following prerequisite courses:

- Architecting Splunk Enterprise Deployments
- Troubleshooting Splunk Enterprise
- Splunk Cluster Administration
- Splunk Deployment Practical Lab

Candidates who are **Splunk Enterprise Certified Admin** and have completed all of the above courses will automatically receive an exam authorization for the Splunk Enterprise Certified Architect exam within 5-7 business days of receiving their passing lab results.

See [here](#) for registration assistance.

Congratulations! You are a...



Recommended Next Steps

- [Splunk Core Certified Consultant](#)

Splunk Core Certified Consultant

This certification demonstrates an individual's ability to properly size, install, and implement Splunk environments and to advise others on how to utilize the product and maximize its value for their needs



Prerequisite Certification(s):

- [Splunk Core Certified Power User](#)
- [Splunk Enterprise Certified Admin](#)
- [Splunk Enterprise Certified Architect](#)

Prerequisite Course(s):

- Advanced Power User courses **or** digital badge*
- Core Consultant Labs
 - Indexer Cluster Implementation
 - Distributed Search Migration
 - Implementation Fundamentals
 - Architect Implementation 1-3
- Services Core Implementation

Splunk Core Certified Consultant Exam

Time to [study](#)! We **require** candidates looking to register for this exam to complete the following prerequisite courses:

- *Fundamentals 3, Creating Classic Dashboards, Advanced Searching & Reporting**
- Core Consultant Labs
- Services Core Implementation

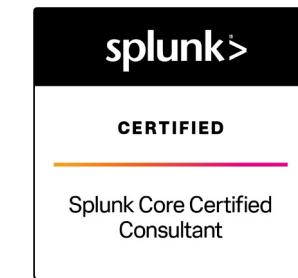
Candidates who are **Splunk Enterprise Certified Architects** and have completed all of the above courses must contact certification@splunk.com to request their Core Consultant exam authorization.

See [here](#) for registration assistance.

*These Advanced Power User courses can be replaced with a Splunk Certified Advanced Power User badge **or** completion of the following courses:

- | | |
|--|--|
| <ul style="list-style-type: none"> • Using Fields • Creating Field Extractions • Enriching Data with Lookups • Data Models • Search Optimization • Working with Time • Leveraging Lookups and Subsearches | <ul style="list-style-type: none"> • Comparing Values • Correlation Analysis • Result Modification • Multivalue Fields • Search Under the Hood • Creating Maps • Introduction to Dashboards • Dynamic Dashboards |
|--|--|

Congratulations! You are a...



Recommended Next Steps

- None

Splunk Enterprise Security Certified Admin

This certification demonstrates an individual's ability to install, configure, and manage a Splunk Enterprise Security deployment



Prerequisite Certification(s):

- None

Prerequisite Course(s):

- None

Splunk Enterprise Security Certified Admin Exam

Time to [study](#)! We suggest candidates looking to prepare for this exam complete the following course:

- Administering Splunk Enterprise Security

Please note: all candidates are expected to have working knowledge and experience as either Splunk Cloud or Splunk Enterprise Administrators.

See [here](#) for registration assistance.

Congratulations! You are a...



Recommended Next Steps

- Splunk Phantom Certified Admin

Splunk IT Service Intelligence Certified Admin

This certification demonstrates an individual's ability to deploy, manage, and use Splunk ITSI to monitor mission-critical services



Prerequisite Certification(s):

- None

Prerequisite Course(s):

- None

Splunk IT Service Intelligence Certified Admin Exam

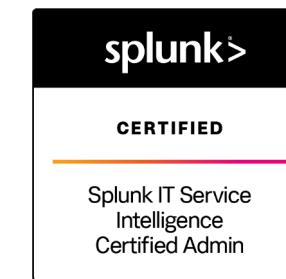
Time to [study](#)! We suggest candidates looking to prepare for this exam complete the following course:

- Implementing Splunk IT Service Intelligence

Please note: all candidates are expected to have working knowledge and experience as either Splunk Cloud or Splunk Enterprise Administrators.

See [here](#) for registration assistance.

Congratulations! You are a...



Recommended Next Steps

- [Courses on Observability](#)

Splunk SOAR Certified Automation Developer

This certification demonstrates an individual's ability to install and configure a SOAR server, integrate it with Splunk, and plan, design, create, and debug playbooks



Prerequisite Certification(s):

- None

Prerequisite Course(s):

- None

Splunk SOAR Certified Automation Developer Exam

Time to [study](#)! We suggest candidates looking to prepare for this exam complete the following courses:

- Administering SOAR (Phantom)
- Developing SOAR (Phantom) Playbooks
- Advanced SOAR (Phantom) Implementation

Please note: all candidates are expected to have working knowledge and experience as either Splunk Cloud or Splunk Enterprise Administrators.

See [here](#) for registration assistance.

Congratulations! You are a...



Recommended Next Steps

- None

Thank You



splunk® turn data into doing™