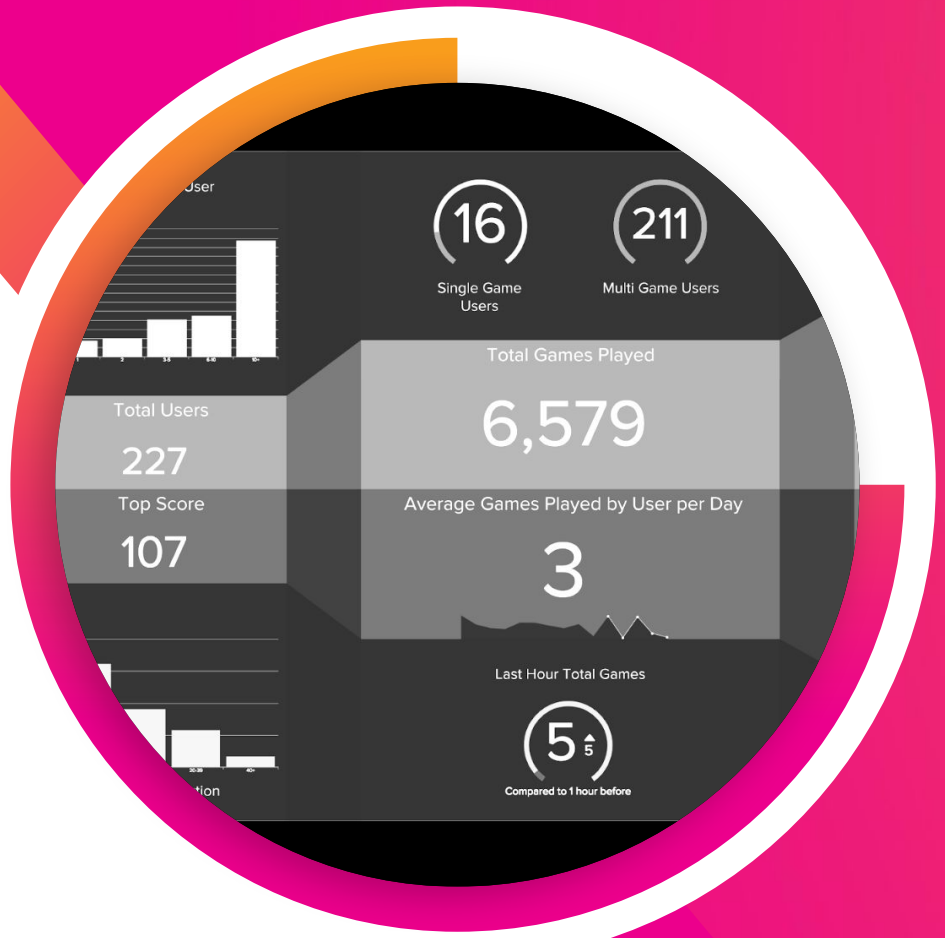


Deep Dive on The New Dashboarding and Content Export Experience

Michael Luo
Principal Software Engineer | Splunk

Yuxiang Kou
Senior Software Engineer | Splunk

splunk> .conf19

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or plans of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results may differ materially. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, it may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements made herein.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only, and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionalities described or to include any such feature or functionality in a future release.

Michael Luo
Principal Software Engineer | Splunk

Yuxiang Kou
Senior Software Engineer | Splunk

splunk> .conf19

Target Audience & Prerequisites

Splunk App Developers

- Familiar with javascript and css.
- Understand how to use node and npm packages.
- Familiar with React and JSX.

If you ever developed SimpleXML dashboard extensions, this session is right for you!

Dashboard in Splunk

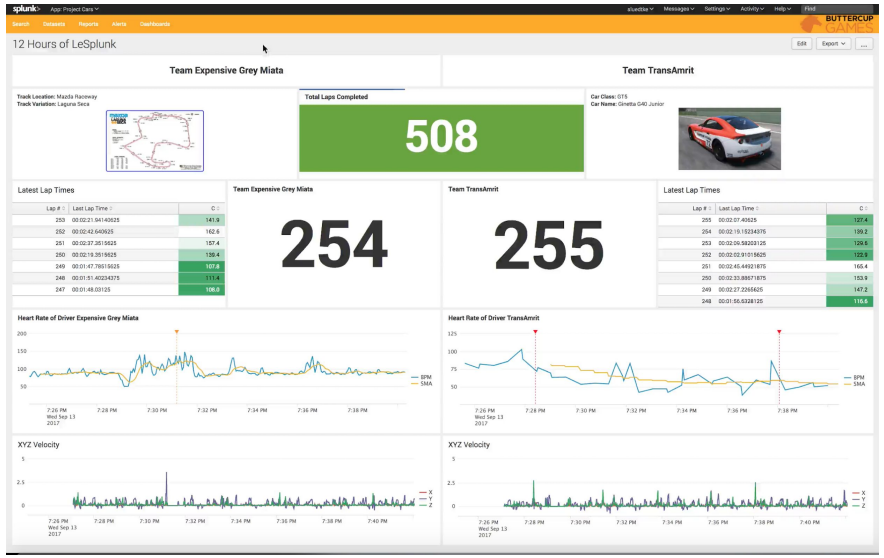
splunk> .conf19

Existing Dashboard Framework

SimpleXML

Shipped as default dashboard in Splunk Enterprise

- Support Javascript/css extension



Glasstable

- Bundled in Splunk premium app such as ITSI
- Support advanced editing experience

splunk> .conf19

Dashboard Challenges

splunk> .conf19

SimpleXML Challenges

Hard to extend and maintain

- Weak APIs - extension are often break when upgrading splunk
- Depends on Splunk web which will change over time
- Developers rely on examples or code snippet to learn

Lacks some popular features

- Absolute positioning, Text, markdown

Uses outdated web technologies

- Backbone, jQuery, RequireJ
- Hard to integrate with modem web technology such as React or Vue

Glasstable Challenges

Not available for developers

- Cannot be used in 3rd-party apps
- Cannot be customized through JavaScript/CSS

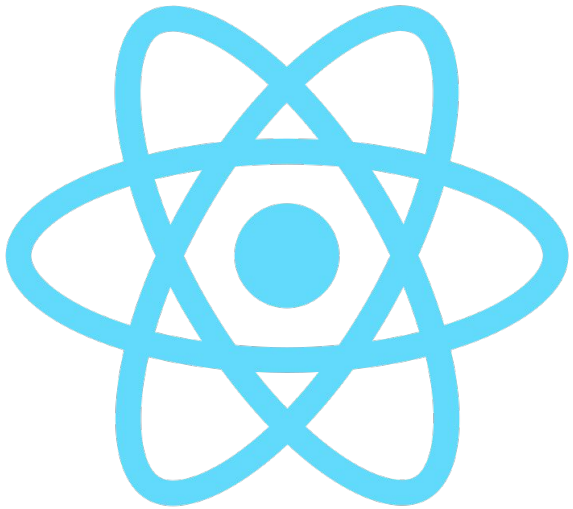
Uses outdated web technologies

- Backbone, jQuery, Bootstrap, RequireJS

Introducing New Dashboard Framework

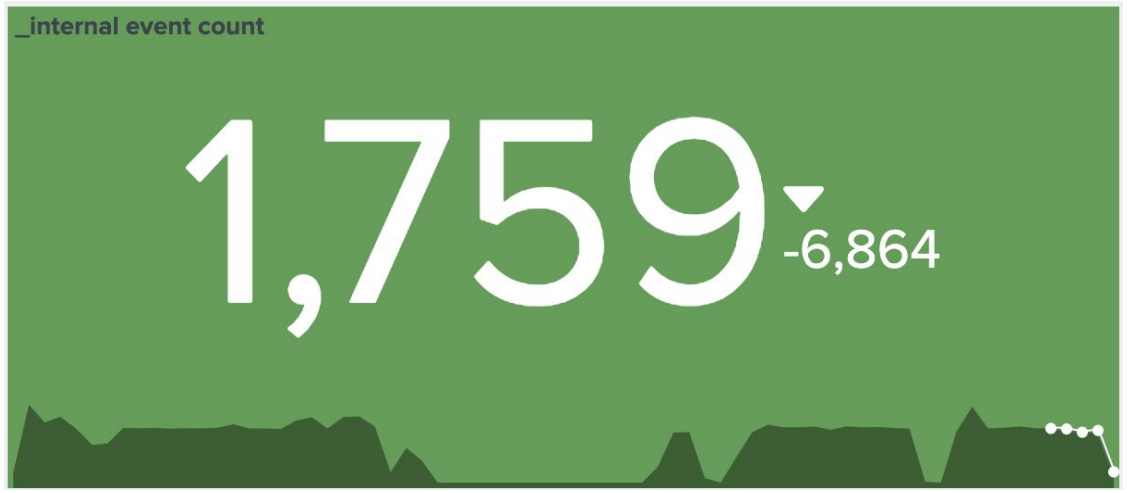
One framework solves all problems

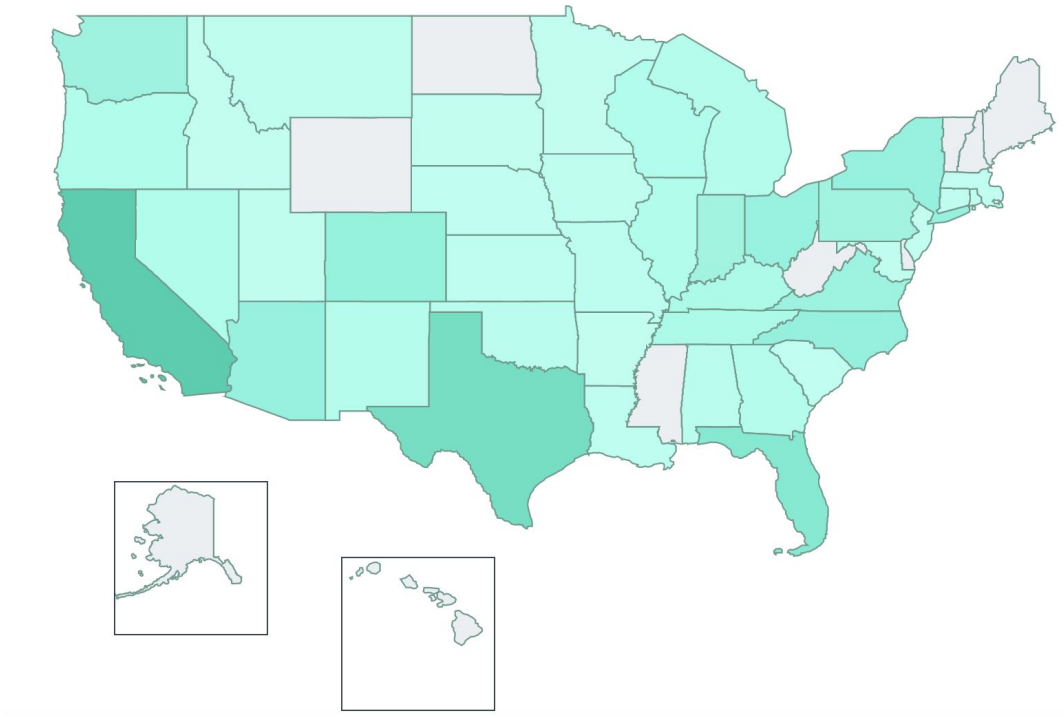
splunk> .conf19



New Layouts

Updated Visualizations

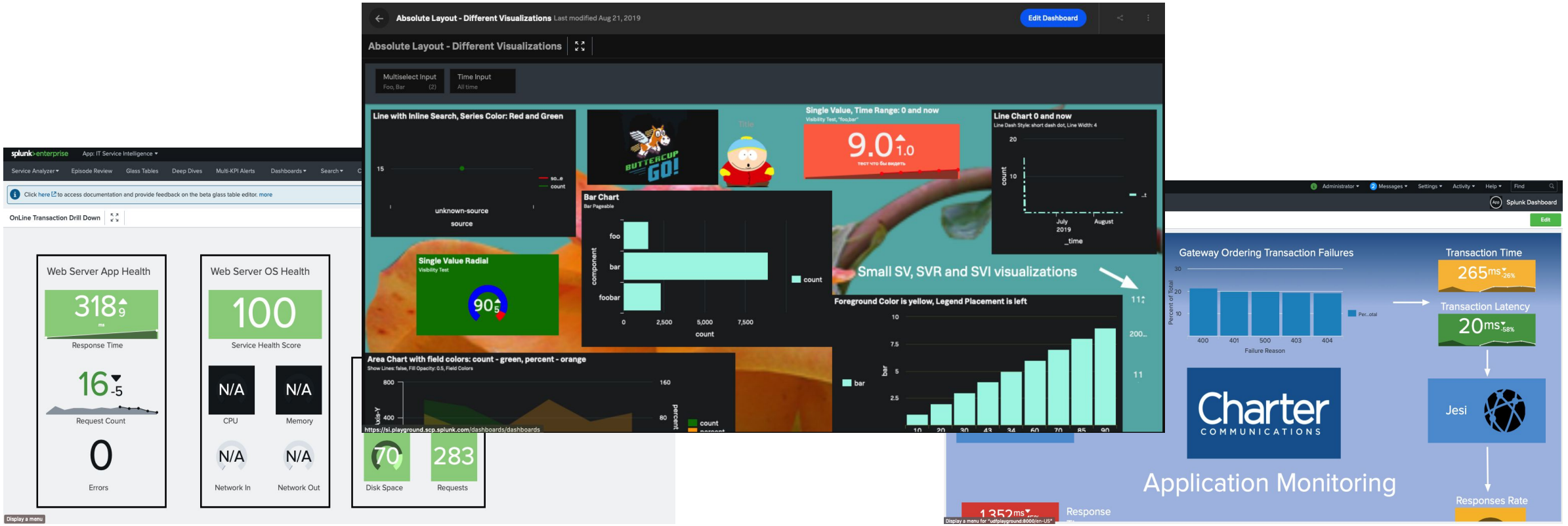




Many Other New Widgets

Splunk Built Applications

Splunk Investigate



Splunk IT Service Intelligence

Splunk Dashboard Beta App

splunk> .conf19

Technical Overview

splunk> .conf19

Key Components

Dashboard Definition.

- A JSON object that describe a dashboard.

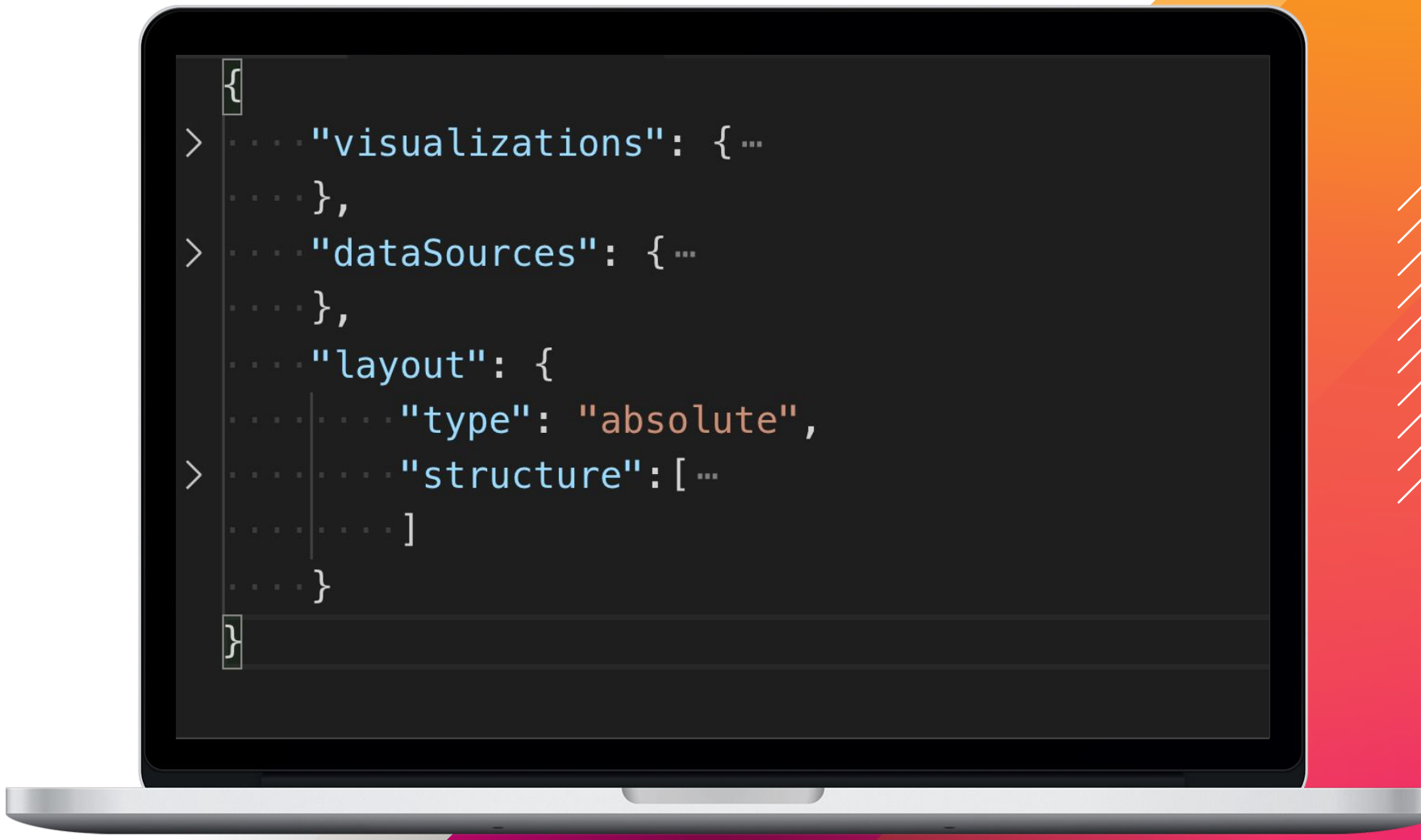
Dashboard Preset.

- A JavaScript module that contains a collection of dashboard elements.

DashboardCore.

- A React component that renders a dashboard from dashboard definition and preset.

Dashboard Definition



@splunk/dashboard-presets

A collection of dashboard components

Layouts

- Absolute Layout
- Grid Layout
- Row Column Layout

Visualizations

- Bar/column
- Line/area
- Pie
- Scatter
- Bubble
- Single value
- Table
- Text
- Image
- Markdown
- Icon

Inputs

- Time range picker
- Dropdown Input
- Text Input
- MultiSelect Input

DataSource

- Enterprise Search
- SDC Search

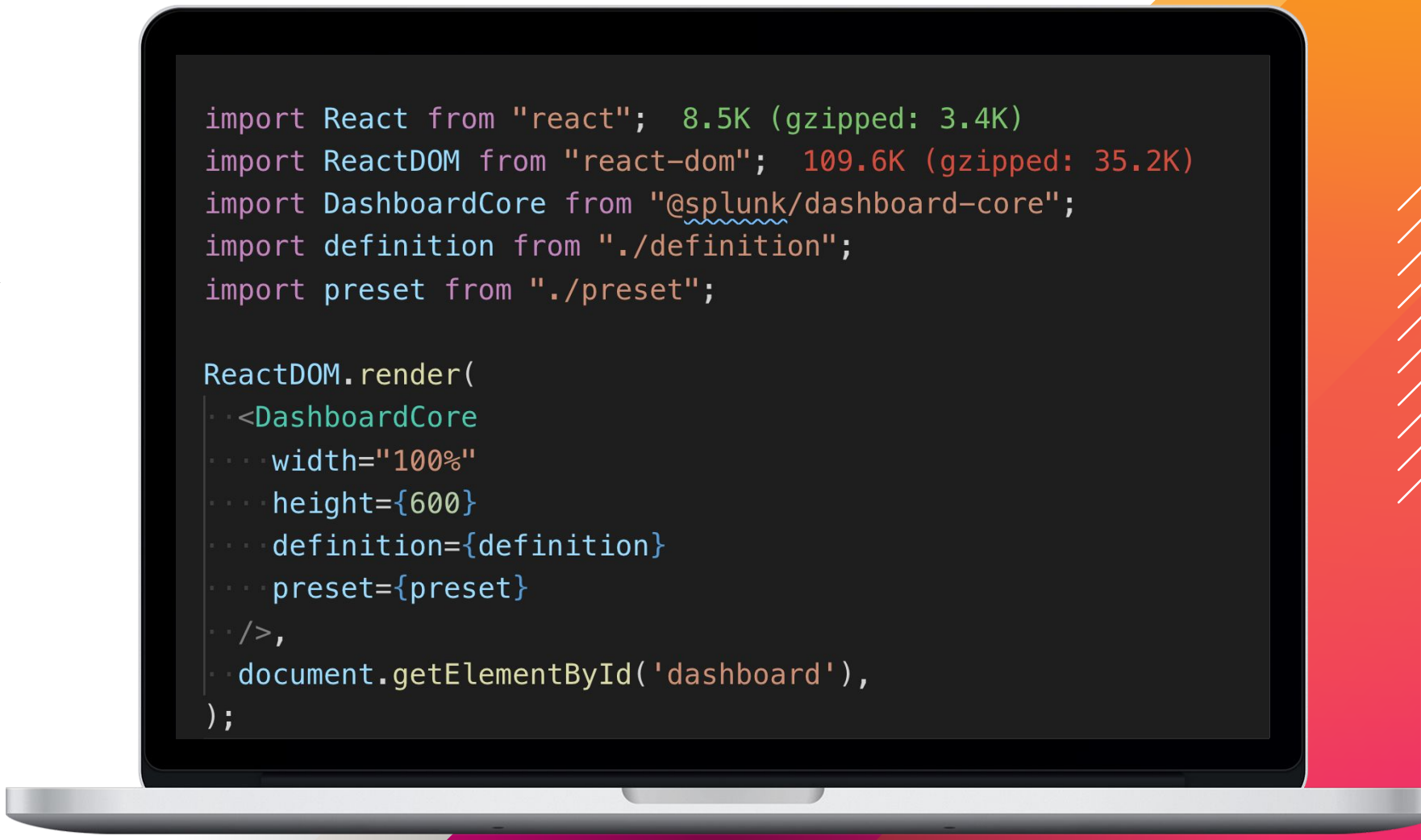
Default Presets

Enterprise Preset	SCS Preset
@splunk/dashboard-presets/EnterprisePreset	@splunk/dashboard-presets/SDCPreset
All Built-in visualizations	All Built-in visualizations
All Built-in Inputs	All Built-in Inputs
Grid, Absolute and Row Column Layout	Grid, Absolute and Row Column Layout
SPL DataSource	SPL2 DataSource

@splunk/dashboard-core

The engine that renders a dashboard

@splunk/dashboard-core is a React component that renders dashboard



Demo - Simple Dashboard on Splunk Enterprise

https://github.com/splunk/dashboard-conf19-examples

splunk> .conf19

Recap

- Bootstrap a Splunk Enterprise application
- Install dashboard packages and their dependencies
- Craft dashboard definition
  - Choose layout.
  - Define visualization and their datasource, place them in the layout.
- Use DashboardCore renders the defined dashboard

Step 1: Install Packages

1) Install basic packages
   - $ npm install react react-dom styled-components@^4 @splunk/react-ui

2) Install dashboard packages
   - $ npm install @splunk/dashboard-context
   - $ npm install @splunk/dashboard-core
   - $ npm install @splunk/dashboard-presets

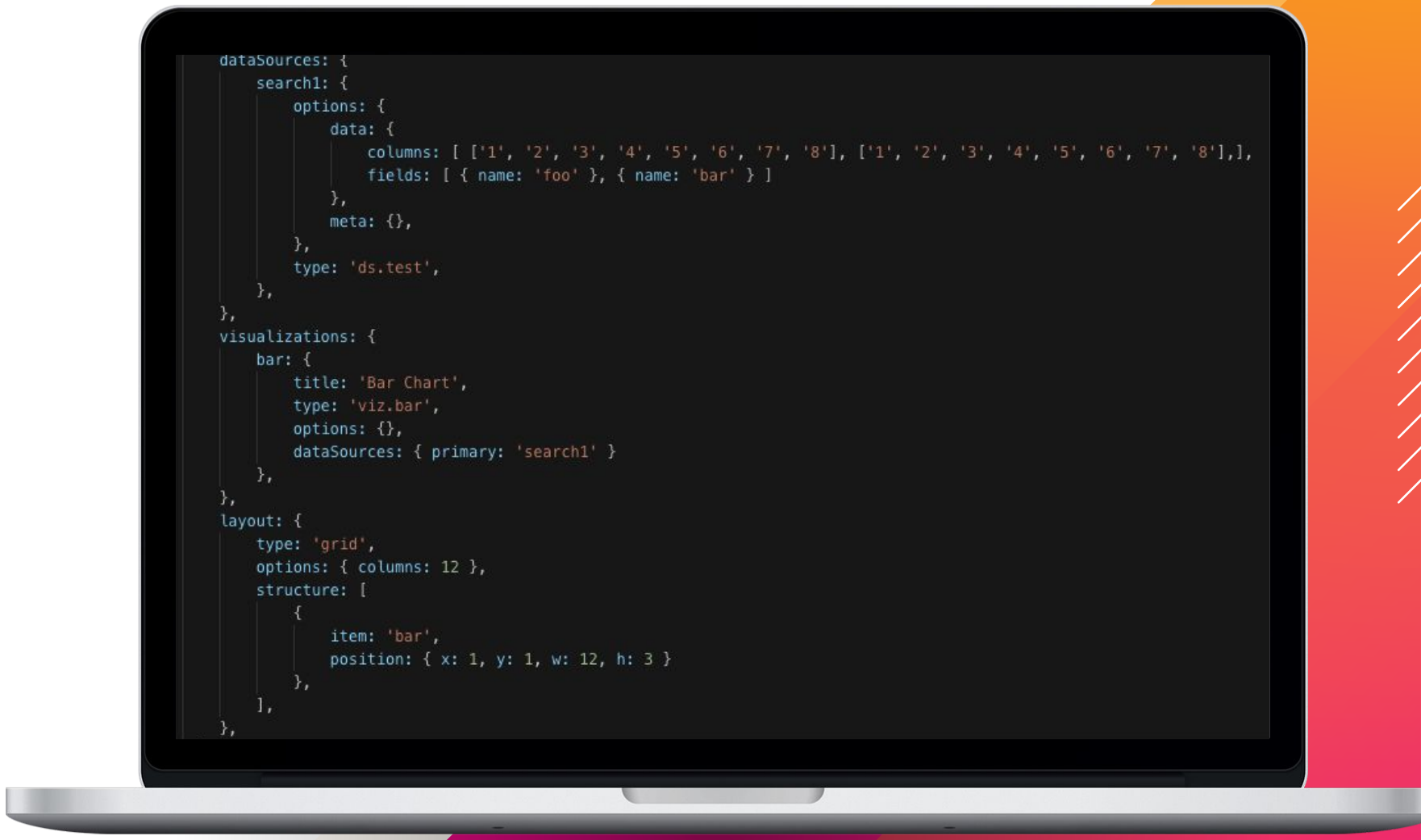
3) Make sure the versions are compatible, check official docs

Step 2: Import and Use

```
// in your main JavaScript file
import React from 'react’;
import { render } from 'react-dom';
import DashboardCore from '@splunk/dashboard-core';
import Presets from '@splunk/dashboard-presets/ViewOnlyPresets’;

render(
    <DashboardCore width="100%" height={400} preset={Presets} />,
 document.querySelector(‘body’)
);
```

Step 3: Create a Definition



```
dataSources: {
    search1: {
        options: {
            data: {
                columns: [ ['1', '2', '3', '4', '5', '6', '7', '8'], ['1', '2', '3', '4', '5', '6', '7', '8'],],
                fields: [ { name: 'foo' }, { name: 'bar' } ]
            },
            meta: {},
        },
        type: 'ds.test',
    },
},
visualizations: {
    bar: {
        title: 'Bar Chart',
        type: 'viz.bar',
        options: {},
        dataSources: { primary: 'search1' }
    },
},
layout: {
    type: 'grid',
    options: { columns: 12 },
    structure: [
        {
            item: 'bar',
            position: { x: 1, y: 1, w: 12, h: 3 }
        },
    ],
},
```

Step 4: Use the Definition

```
// in your main JavaScript file
import React from 'react’;
import { render } from ‘react-dom’;
import DashboardCore from '@splunk/dashboard-core';
import Presets from '@splunk/dashboard-presets/ViewOnlyPresets’;

const definition = { /* created previously */ };

render(
    <DashboardCore width="100%" height={500} preset={Presets}
      definition={definition} />,
 document.querySelector(‘body’),
);
```

Step 1: Define a Visualization

Code snippet

```
{
    // ...
    visualizations: {
        my_bar_viz: {
            title: 'Bar Chart',
            type: 'viz.bar',
            options: { ... },
            dataSources: {
                primary:
'search1',
            },
        },
    },
    // ...
}
```

id of the visualization

type of the visualization

options such as axis label, axis title, font color

defines how the visualization get data

Step 2: Define Layout and Put Visualization In

Code snippet

```
{
    // ...
    layout: {
        type: 'grid',
        options: { columns: 12 },
        structure: [
            {
                item: 'my_bar_viz',
                position: { x: 1, y: 1, w: 12, h: 3 }
            },
        ],
    },
    // ...
}
```

an array of visualization items

The id of visualization defined earlier

row column number

width and height in terms of rows and columns

Step 3: Define Data Source

Code snippet

```
{
    // ...
    dataSources: {
        search1: {
            options: {
                data: {
                    columns:[['1','2','3','4','5','6','7','8'],['1','2','3','4','5','6','7','8'],],
                    fields: [ { name: 'foo' }, { name: 'bar' } ]
                },
            },
            type: 'ds.test',
        },
    },
    // ...
}
```

id of data source, will be used by visualization

hard-coded data in configurations

type of data source

Step 3.1: Define Splunk Enterprise Search Source

```
{
    // ...
    dataSources: {
        search1: {
            "type": "ds.search",
            "options": {
          "queryParameters": { "earliest": "-15m", "latest": "now” },
          "query": "index=_internal |  timechart count()"
        },
        },
    },
    // ...
}
```

a new type, requires preset change, more details later

SimpleXML vs. New Dashboard Framework

SimpleXML	New Dashboard Framework
A Splunk Enterprise feature for end users	A set of dashboard components for developers
Bundled with Splunk Enterprise	Shipped as NPM packages
XML definition (rows, panels)	JSON definition (visualizations, data sources, layouts)
Includes built-in visualization custom visualization	Includes built-in visualization custom visualization
Extend via custom javascript + custom css	React, JSX, ES6 and whatever you want to use.

Advanced Dashboards

splunk> .conf19

Custom Preset

Code snippet

```
1   import Line from '@splunk/dashboard-visualizations';
2   import SplunkSearch from '@splunk/datasources/SplunkSearch';
3   import GridLayoutViewer from '@splunk/dashboard-layouts/GridLayoutViewer';
4
5   export const preset = {
6       visualizations: { 'viz.line': Line },
7       dataSources: { 'ds.search': SplunkSearch },
8       layouts: { grid: GridLayoutViewer },
9   };
10
```

Customize Visualization

Include your own components in the preset

DEV2178 - Build your own custom data visualization on dashboard

- Thursday, October 24, 10:30 AM - 11:15 AM

Demo - Custom Data Source

splunk> .conf19

Recap

1) Identify the data source and its format.

2) Create new datasource that extend from @splunk/datasources/DataSource

3) Implement datasource lifecycle methods
   - setup()
   - request()
   - teardown()

4) Extend preset to include your new datasource.

Other Advanced Use Cases

Token, namespaces, filters

- Similar to how they work in SimpleXML

Event handlers

- Create and register event handlers triggered by keyboard, mouse or other events
- Example: drilldown to url

Theming

- Dark theme

Guidelines for SCP Dashboard

splunk> .conf19

Integrate Dashboard in SCP App

SCP is a whole different world

- Everything is fresh new (no SimpelXML, no SplunkJS)
- Build app around REST APIs, libraries and SDKs
- Deploy to the cloud

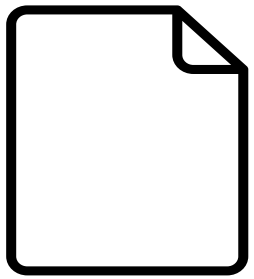
Suggestions

- Get familiar with the new search REST APIs
- Use @splunk/datasources/CloudSearch component
- Get familiar with SCP authentication/authorization concepts

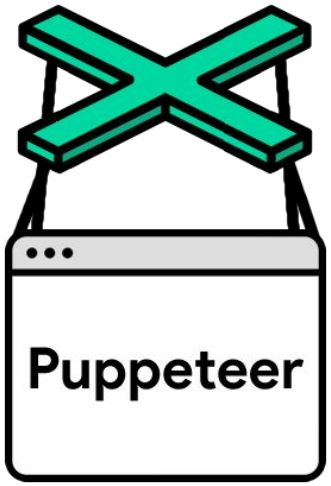
Export Dashboard as PNG/PDF Files

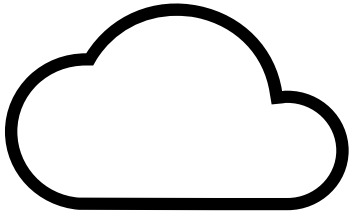
splunk> .conf19

High-level Architecture



Snapshot Definition





SCS Search Service

Resources:

Michael Luo | Principal Software Engineer
Yuxiang Kou | Senior Software Engineer

Don’t miss other dashboard sessions
e.g. ***FN1815, FN1933, FN1735***

splunk> .conf19

.conf19

splunk>

Thank You !

Go to the .conf19 mobile app to

RATE THIS SESSION