Charts & Custom Visualizations Beyond the Map

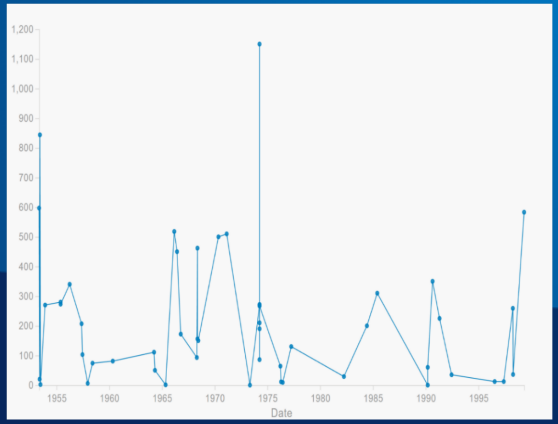
David Martinez & René Rubalcava

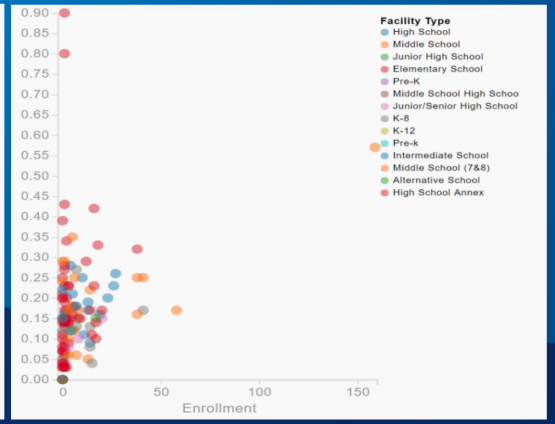
@DavidJmart / @odoenet

Thank You to Our Generous Sponsor

con·terra

Visualization





Visualizations are any medium to present data, visually, to a visual consumer.

- Cartography
- Charts
- Infographics
- Tables

Why Do We Visualize Data?

- Cognitive understanding is faster when visually consumed
- To understand the data
- To frame the data in a different perspective thereby making it easier to reason and consume

"We should never forget that a picture of data is not the goal; it's only the means. Information visualization is all about gaining the understanding so we can make good decisions."

Stephen Fews

Understanding Data

- Compare
- Sort
- Filter
- Highlight
- Aggregate
- Re- express

Cedar

JavaScript Library for Creating Charts

Philosophy

- State of art Visualizations
- Re-usable, sharable
- Integrated with ArcGIS API
- Overridable

Working with Cedar

- A Cedar chart needs four things
  - Chart Type
  - Data
  - Mappings
  - DOM Element

Your First Cedar App

Population

Where's the Map?

Chart with Map

Cedar in Hub & Open Data

- Hub & Open Data requirements drive Cedar development
- Add-on for Ember apps ember-cli-cedar

Recent Improvements

- 2016 focused on flexibility & reliability of data queries:
  - Better support for and examples of SQL expressions
  - Ability to transform data returned from server
  - Query callback for error handling
  - Timeouts for slow loading Data
- 2017 focussing on cedar v1...

Where Are We Headed?

Cedar v1

Released in alpha last week!

- Better support for multi-series charts
- Support for joining multiple datasets
- New chart types (area, radar)
- improved default styling for charts based on Calcite

v1 uses amCharts

- Basing cedar on amCharts will make it easier to:
  - Create and customize new chart types
  - Customize or extend existing built-in chart types
  - Style and theme charts

Examples

1. Bar
2. Area
3. Pie
4. Radar
5. Line
6. Scatterplot
7. Secure Services

More Intuitive JavaScript API

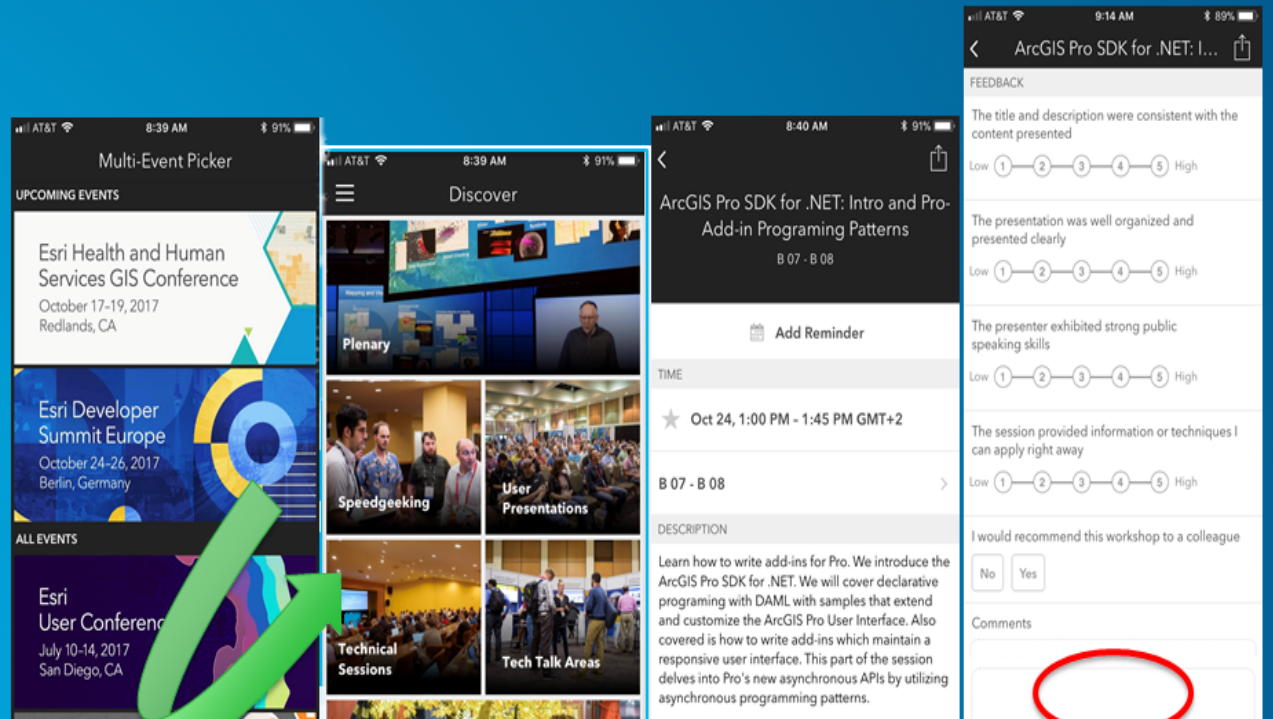
- Async functions use promises instead of callbacks
- fluent, chainable setters and methods
- streamlined, reduced footprint

v1 Roadmap

- open issues
- TLDR: `TODO: documentation!`
- we welcome your ideas, and of course, contributions

Questions?

Help us to improve filling out the survey



esri

THE SCIENCE OF WHERE