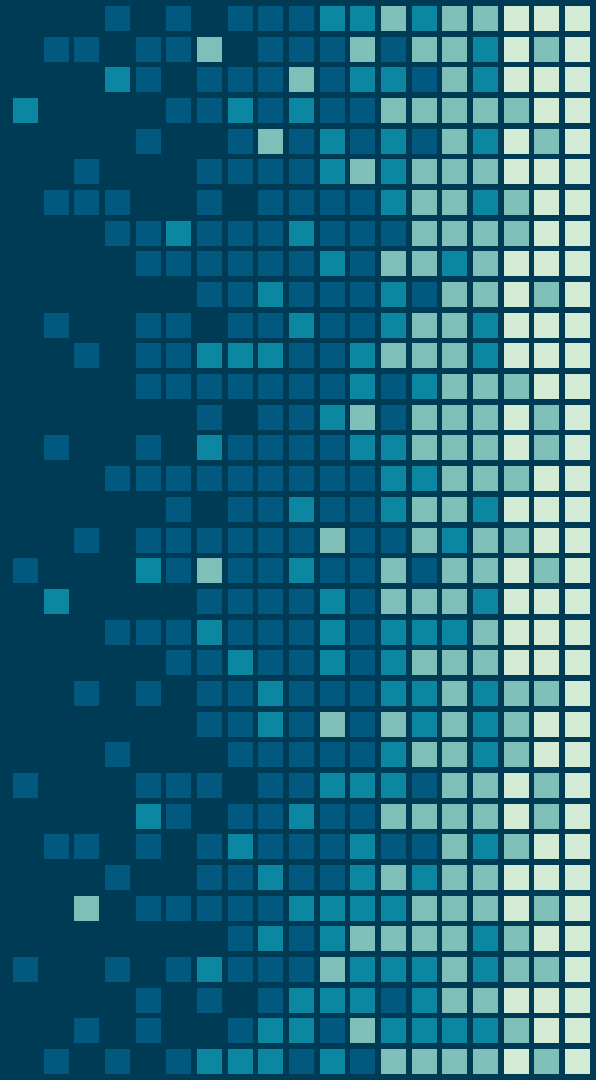


Statistics visualization for the CERN Document Server Invenio-Stats-Js

Ioannis Androulidakis

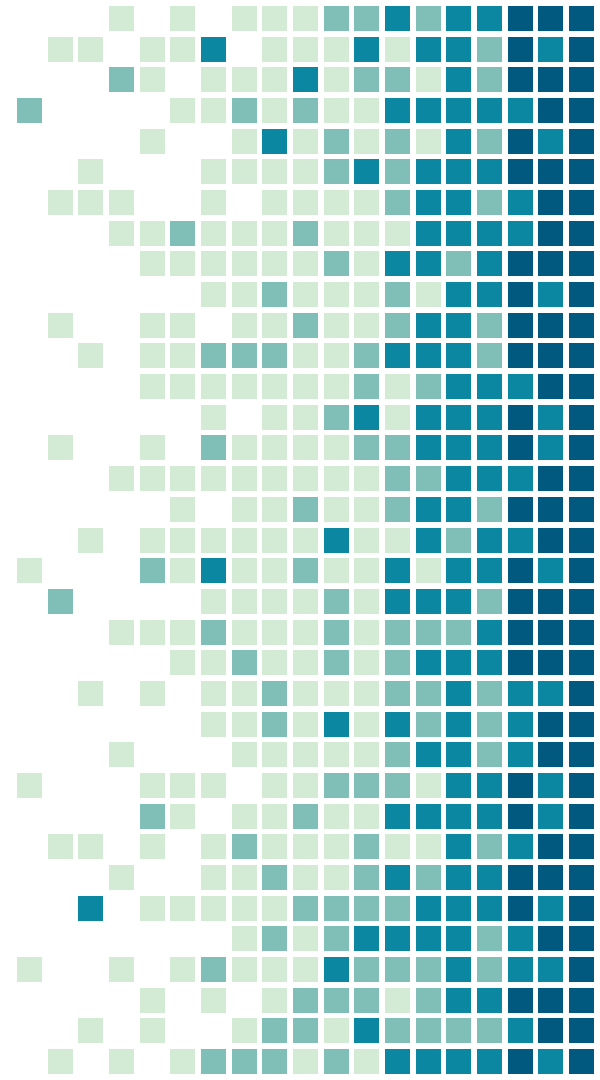
School of Electrical & Computer Engineering,
National Technical University of Athens

August 21, 2017, CERN



DATA FOR HUMANS

Why even visualize ?



Github insights

CERNDocumentServer / cds

Unwatch 14 Star 2 Fork 19

<> Code Issues 81 Pull requests 2 Projects 0 Wiki Insights

Contributors Traffic Commits Code frequency Punch card Network Members Dependents

May 4, 2014 – Aug 8, 2017

Contributions to cdslabs_qa, excluding merge commits

Contributions: Commits



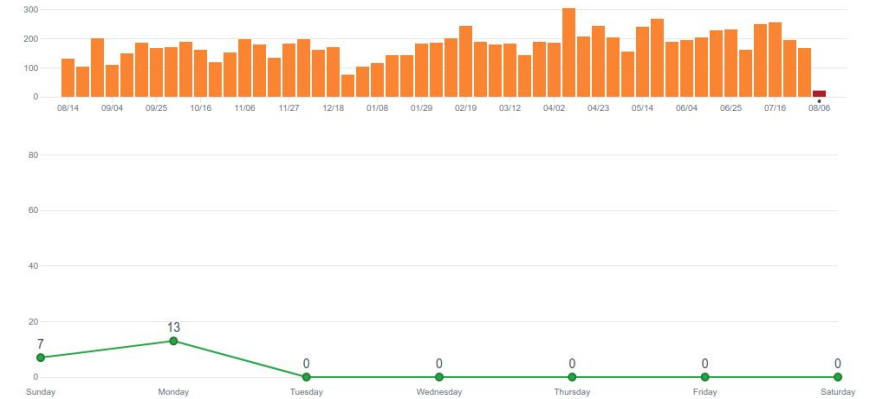
ceph / ceph

Watch 520 Star 3,481 Fork 2,030

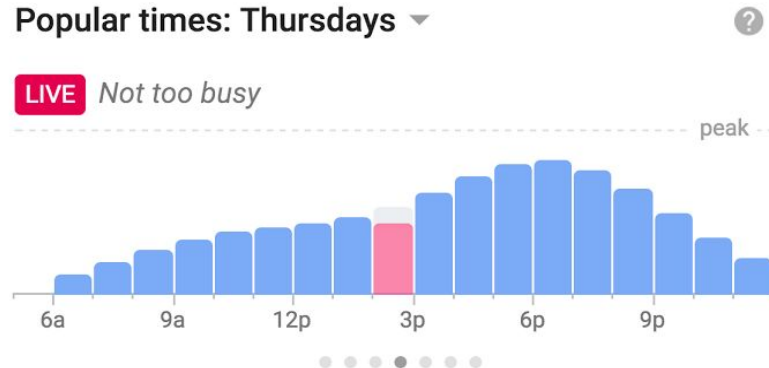
<> Code Pull requests 500 Projects 1 Insights

Contributors Commits Code frequency Punch card Network Members Dependents

Use ← and → to navigate



Google Maps popular hours



 People typically spend **20 min** here

Youtube statistics

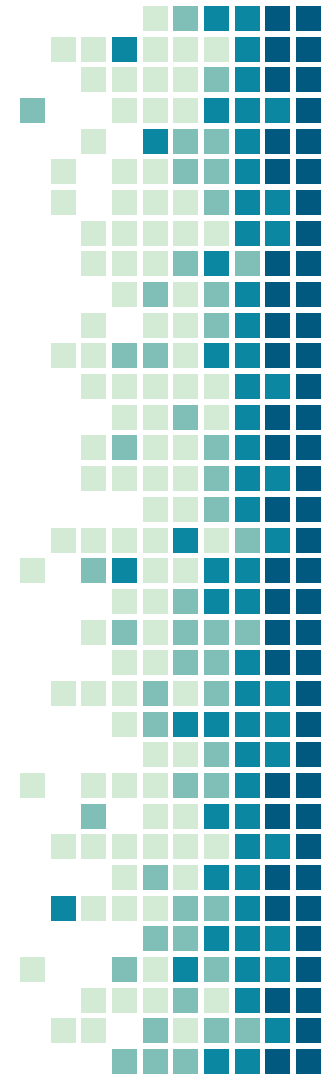


Storytelling

Visualization a simple, intuitive and powerful representation of data

Extract all core information of a dataset

- Discover new insights
- Observation and understanding of behavior
- Trend analysis / pattern recognition
- Data reasoning and correlation
- Decision making and planning



Motivation

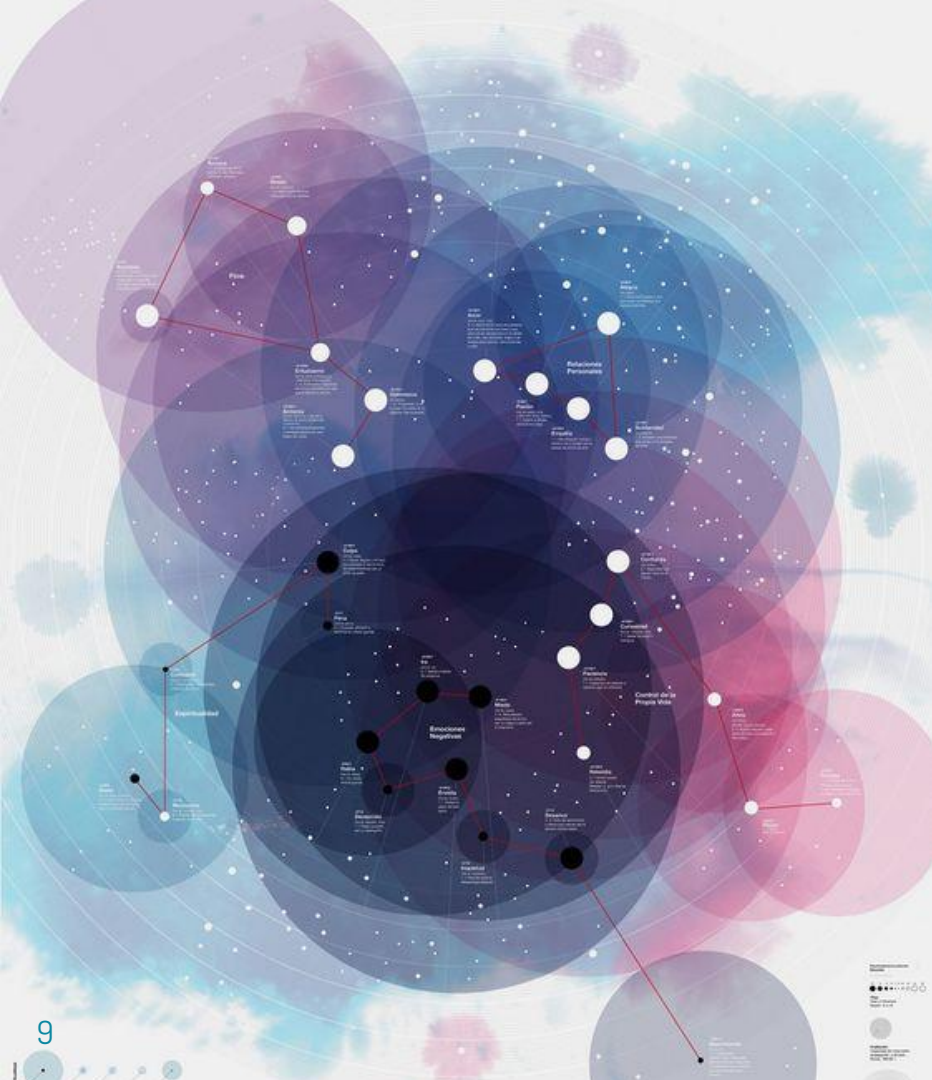


Kibana Dashboard

“ *Design a flexible, standalone visualization module for Invenio, that delivers highly-configurable, interactive and responsive charts* ”

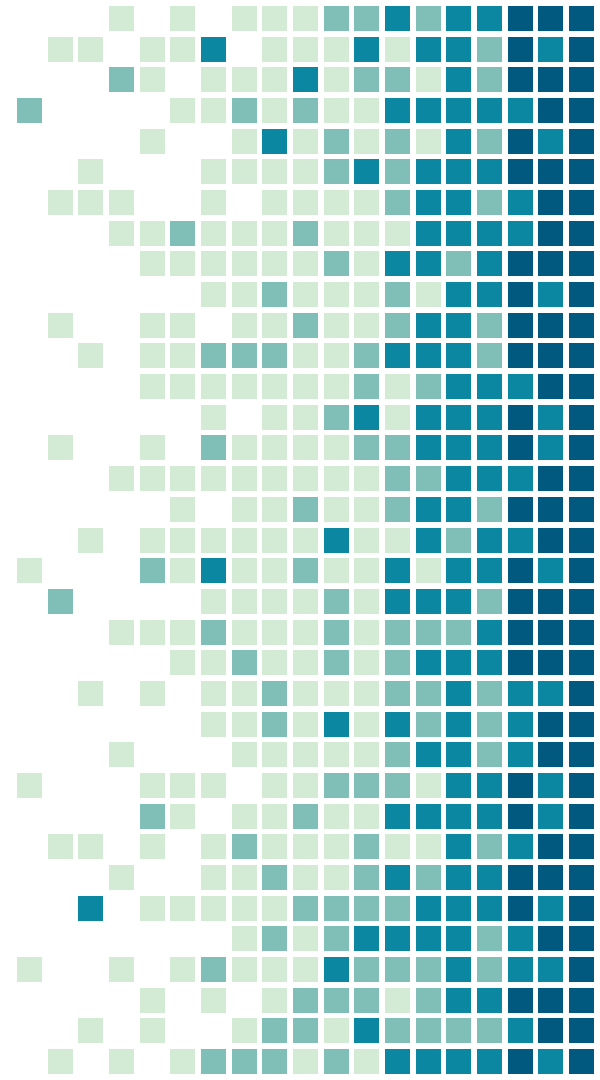
Best practices

- Strip away the noise
- Keep it clean and concise
- Identify needs and customize
- Convey a substantive message
- Keep it open-source 😊



CURRENT PICTURE

Invenio + CDS



CERN projects involved

Invenio 3.0*

A digital library framework that enables building and running large scale digital repositories

Collection of modules that provide functionality

<http://inveniosoftware.org/>

<https://github.com/inveniosoftware/>

CERN Document Server

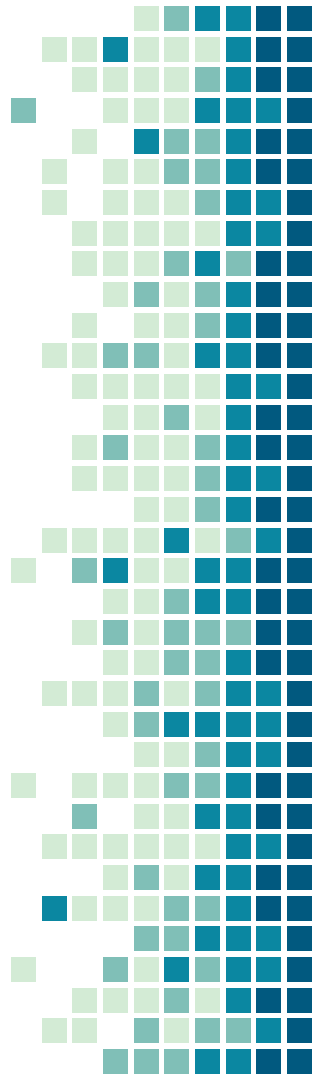
Enables access to articles, reports and multimedia content in High Energy Physics

An Invenio instance

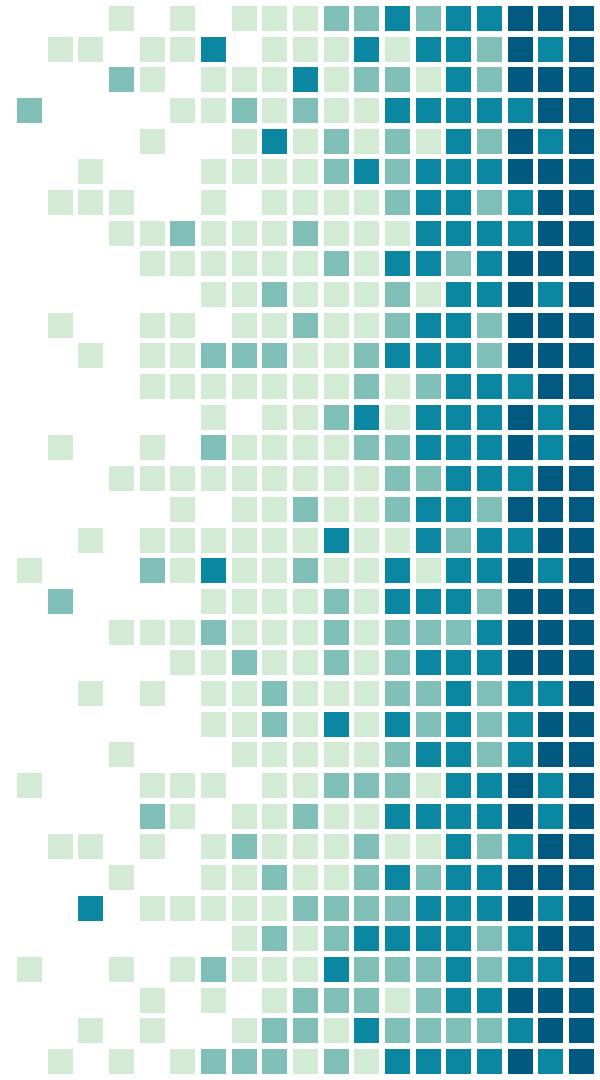
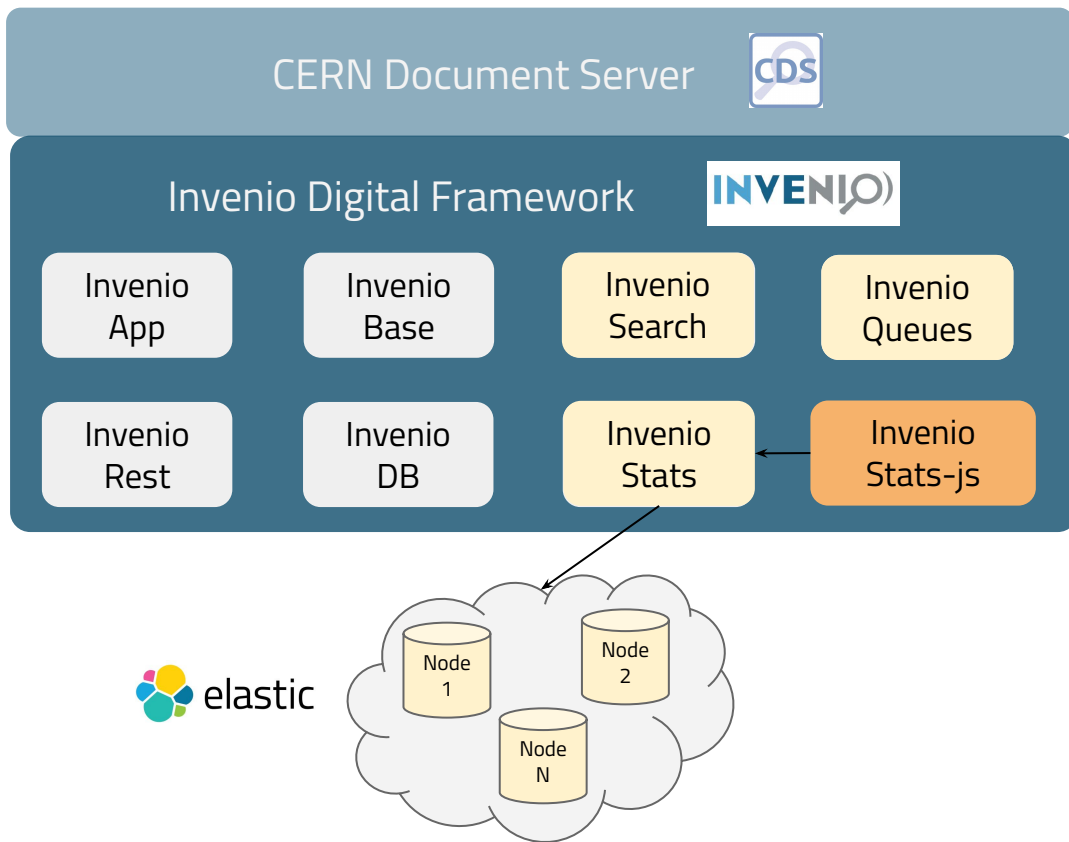
<https://cds.cern.ch/>

<https://github.com/CERNDocumentServer/>

**Currently under development*



Architecture



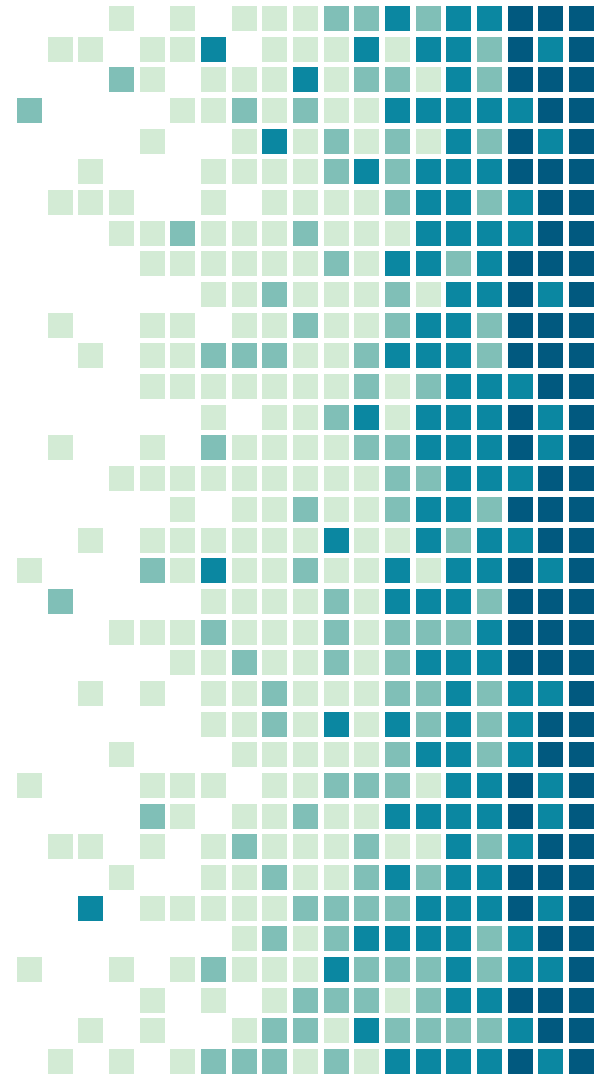
Upcoming CDS videos

The screenshot displays the CDS videos interface. At the top, there is a navigation bar with 'CDS videos', a search bar, and 'Upload' and 'Log In' links. The main content area features a large 3D architectural rendering of a modern facility with a parking lot, surrounded by greenery and mountains in the background. Below the video player, the title 'Esplanade des Particules' is displayed, along with metadata: '© 2017 CERN', 'Conditions of Use', '2017-03-30', '00:03:39', and '3000000'. A 'Show more' link is present. Below the title, there are tabs for 'Route de Meyrin', 'Globe', 'Works', 'Travaux', and 'Reception'. A 'Share' section includes social media icons for Facebook, Twitter, LinkedIn, YouTube, and Email. An 'Embed' section provides an HTML code snippet: `<iframe src="https://cdslabs-qa.cern.ch/video/OPEN-MOVIE-2017-012-001" width="560" height="315" allowfullscreen="true" allow="autoplay; fullscreen; picture-in-picture" style="border: none; width: 100%; height: 100%; margin: 0; padding: 0;"></iframe>`. On the right side, there is a 'Download' section with a 'Download Master file (387.5 Mb)' button and a list of 'Other formats', 'Thumbnails', and 'Additional files'. Below that, a 'Technical Metadata' section lists:

- avg_frame_rate: 24/1
- bit_rate: 14776148
- codec_name: h264

SOFTWARE STACK

Our toolset



Core Components

D3.js v4.10

Data-driven documents

De facto visualization framework

Powerful **functional-style programming**

Selections, bindings, maps, filters, transitions, interactions

Low level concepts in **SVG, HTML & DOM**

NodeJS v4.8

Runtime environment based on the V8 Javascript engine

Event-driven architecture

Automation of **package installation** and code compilation

Ecmascript v6



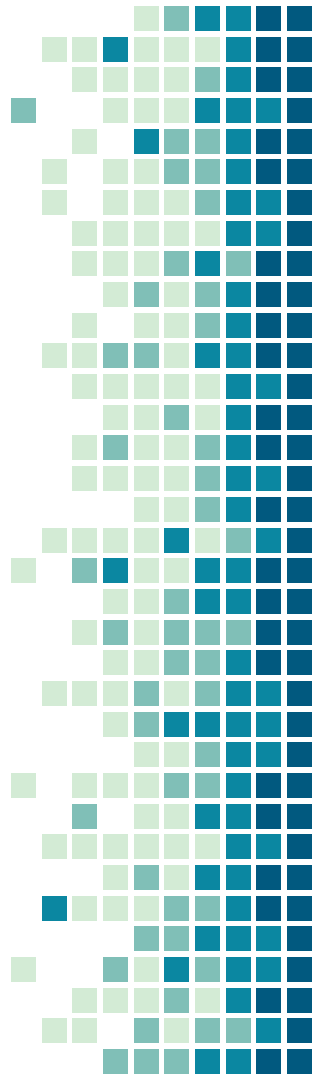
The latest version of **Javascript**, designed to support async/await, arrow functions, **classes**, modules, template literals

Webpack v3.5




A module **bundler** for modern Javascript applications

Build assets, scripts, styles in a **dependency graph**



Code

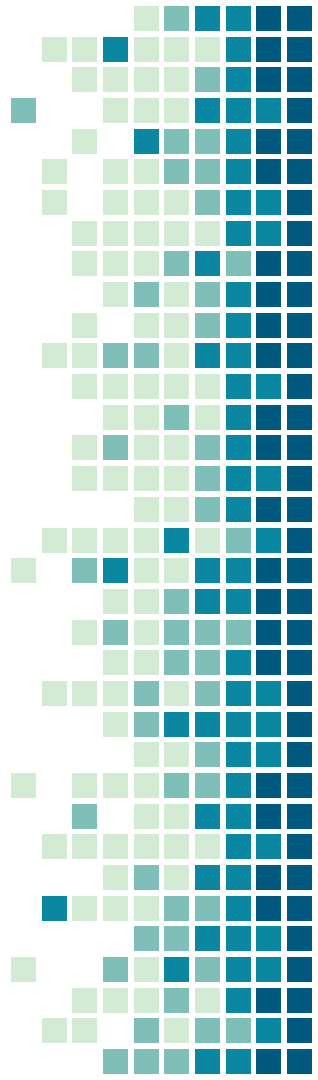
```
npm install invenio-stats-js
```

(soon available on ) 😊

```
// Import as ES2015 module
import { LineGraph, BarGraph } from 'invenio-stats-js';

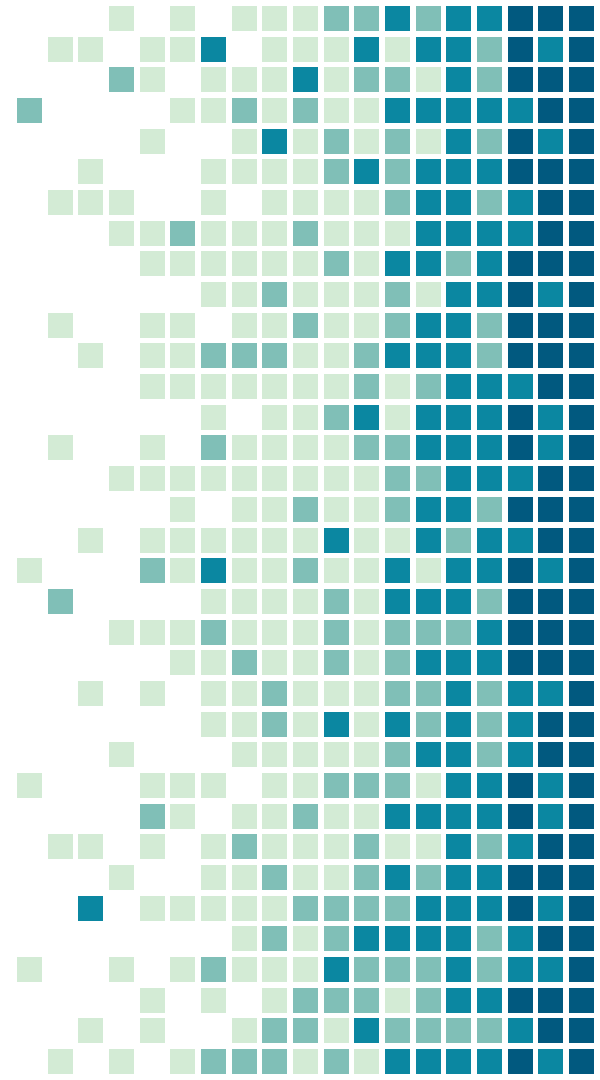
// Create a line chart
new LineGraph(data, 'pageviews').render();

// Create a bar chart
new BarGraph(data, 'downloads').render();
```



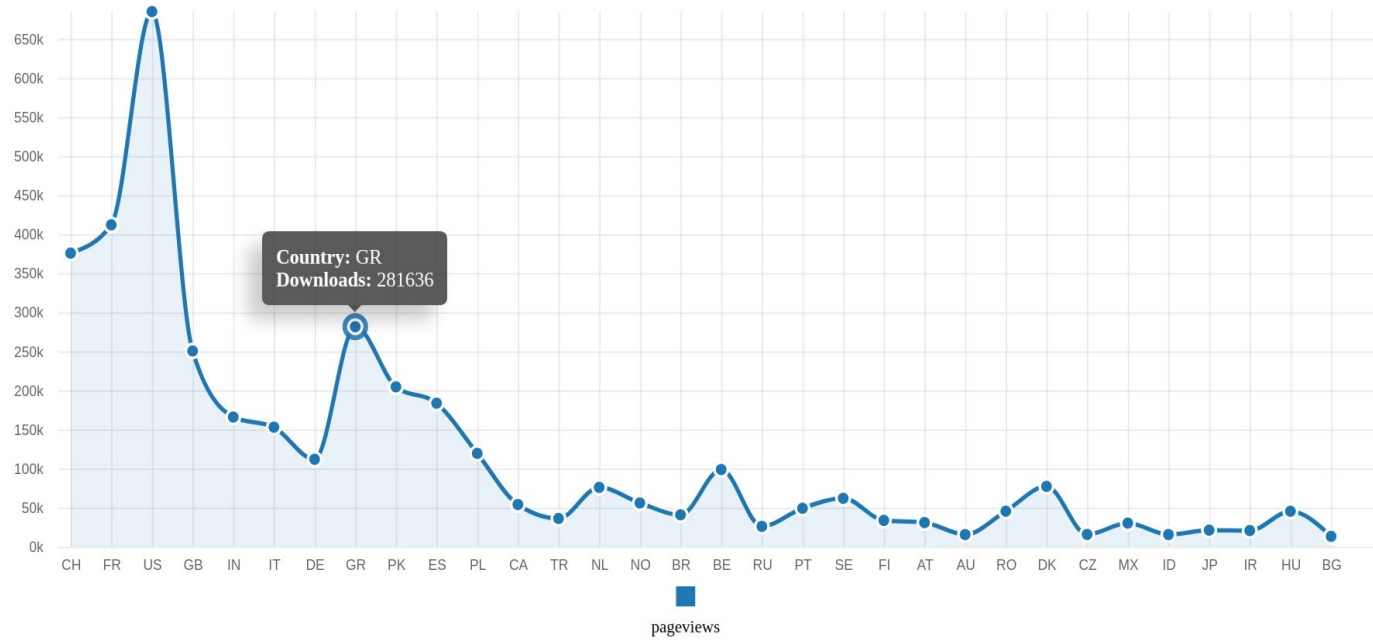
EARLY RESULTS

Alpha version of charts

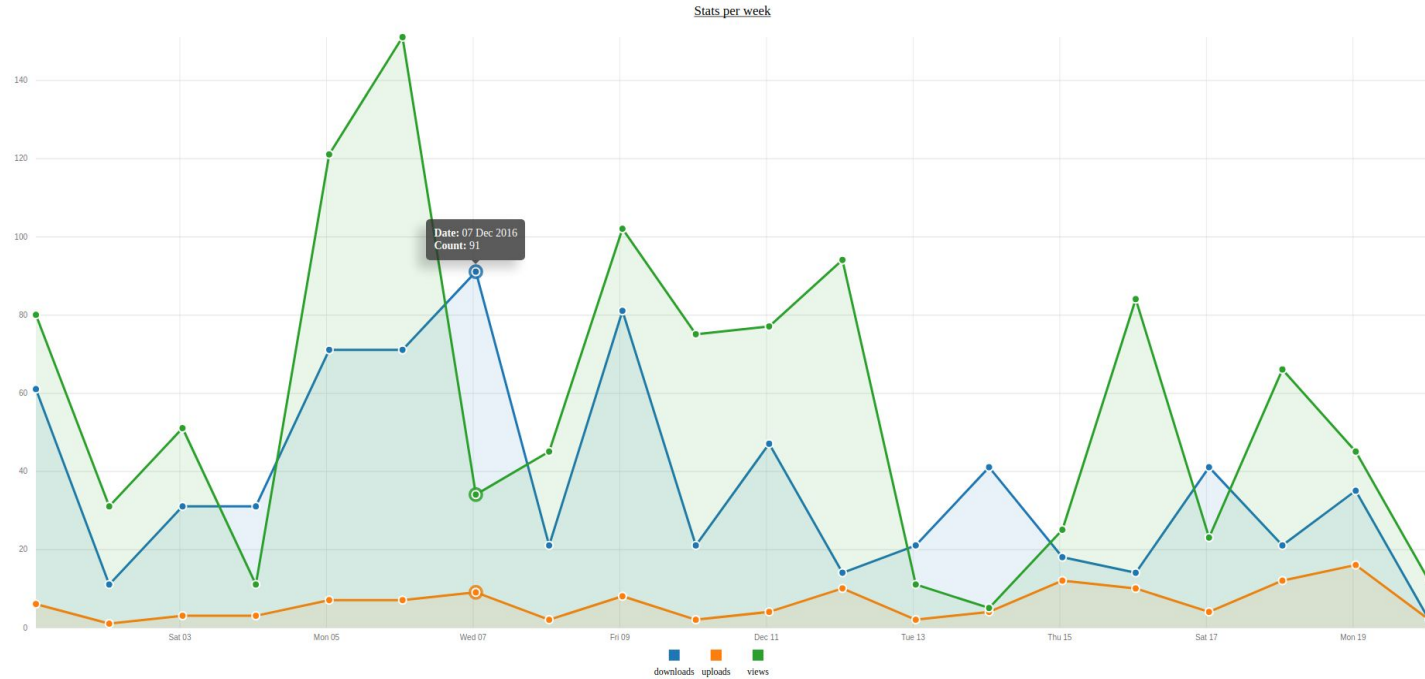


Simple Line/Area Chart

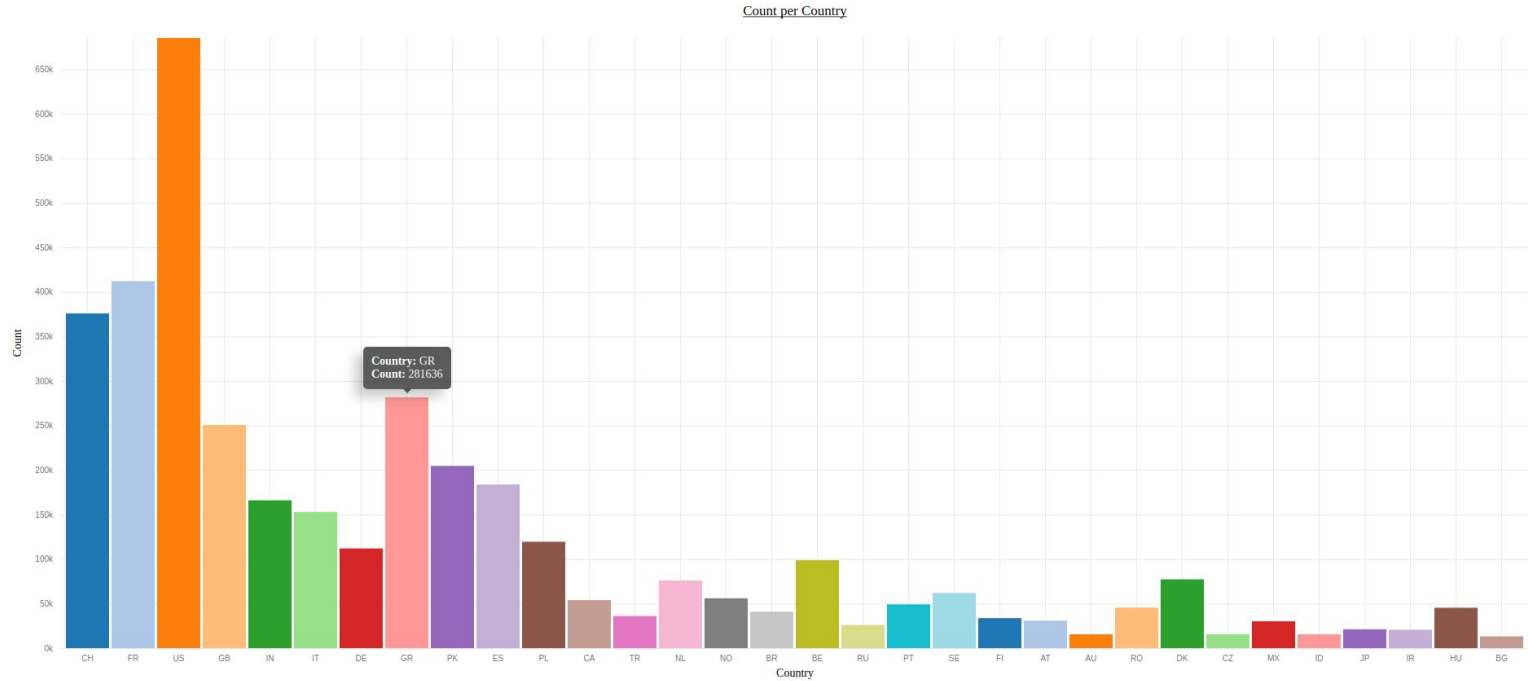
Stats per country



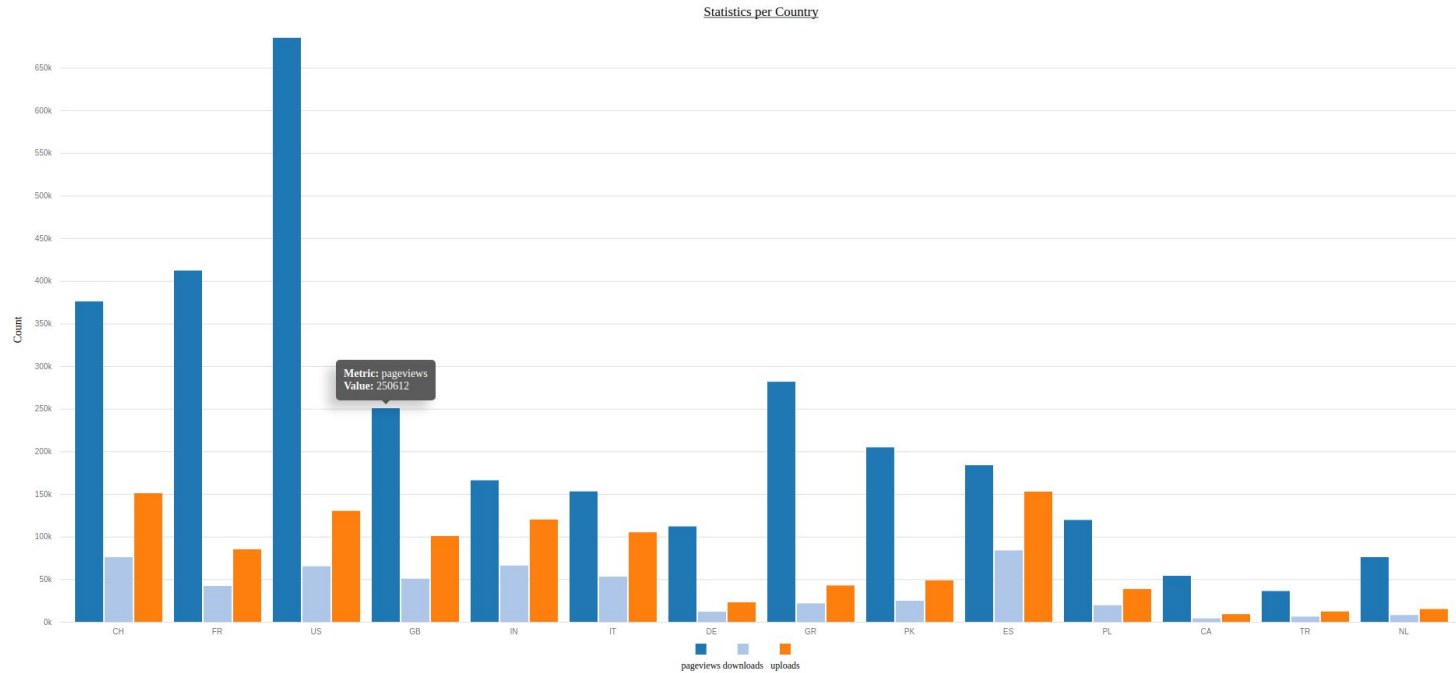
Multi Line/Area Chart



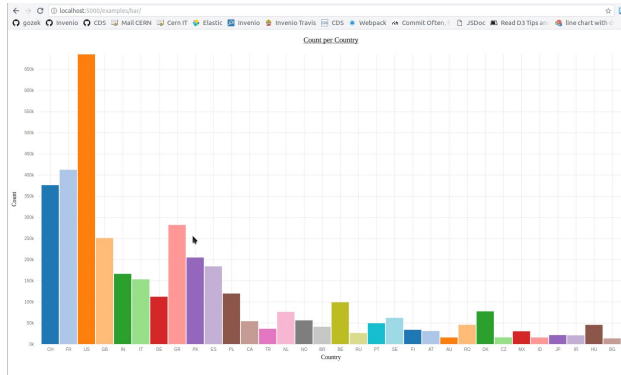
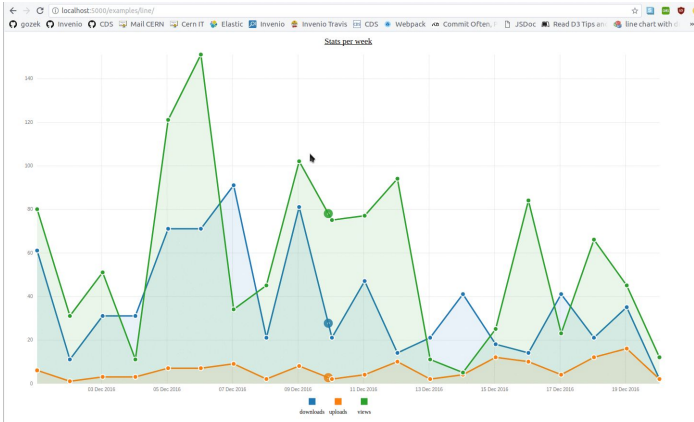
Bar Chart



Grouped Bar Chart

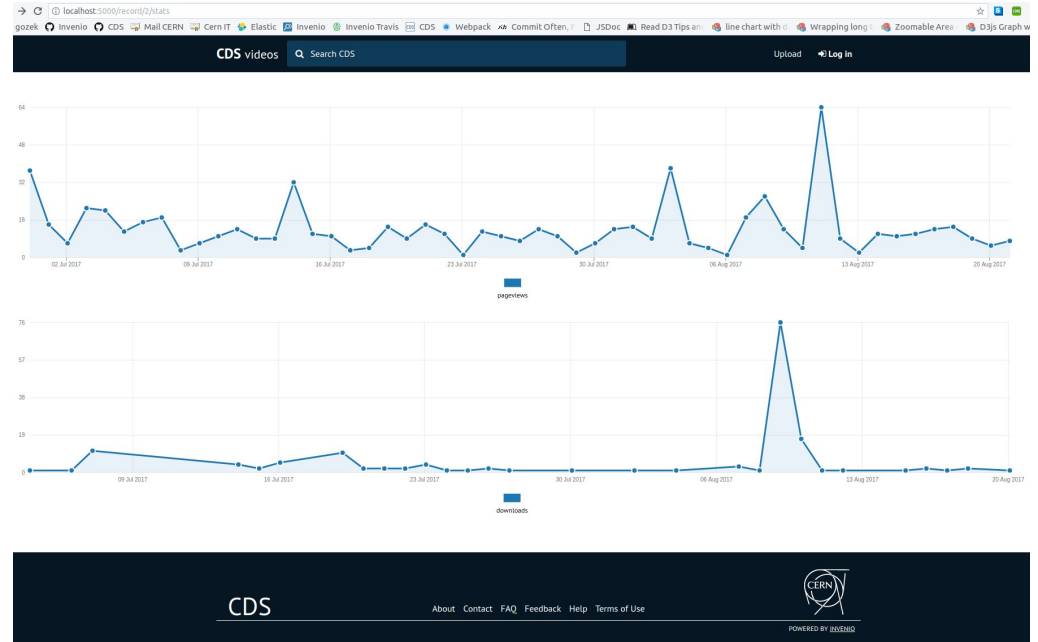


Zoomable, responsive, interactive



Current integration

- Set up instance of **cds-labs**
- Add a legacy **stats module*** to query the ES production cluster :
/api/stats/pageviews
/api/stats/downloads
- Digest response
- Extend **records module** with a testing page for statistics :
/record/<record_id>/stats



Documentation, tutorials and tests

Class: Graph

Graph(config, input, classElement)

Class representing an abstract Graph.

Constructor

new Graph(config, input, classElement)

Create a graph.

Parameters:

Name	Type	Description
config	Object	The configuration of the graph.
input	Array<Object>	The input data.
classElement	String	The class of the SVG element.

Source: [graph/graph.js](#), line 31

Methods

enableZoom()

Enable zoom in the graph.

Source: [graph/graph.js](#), line 605

getType() → (string)

Get the type of the graph.

Source: [graph/graph.js](#), line 68

Returns:

The type value.

Type
string



465 lines (432 sloc) 13 KB

Raw Blame History

Simple Line Graph with default configuration

In this example we are going to create a simple line graph using the default configuration.

1. Before rendering

This is how the HTML document looks before rendering the line graph. In this example, the `div` element classed as `simple-line-graph` is the placeholder of our `svg` graph and has fixed dimensions.

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <script src="../../node_modules/d3/build/d3.min.js"></script>
  <script src="../../node_modules/d3-svg-legend/d3-legend.min.js"></script>
  <script src="../../node_modules/d3-tip/index.js"></script>
  <script src="../../node_modules/odash/odash.min.js"></script>
  <style type="text/css"></style>
</head>
<body>
  <div class="simple-line-graph" style="width: 600px; height: 450px;"></div>
  <script src="/simple-line-graph.js"></script>
</body>
</html>
```

2. Fetching the data

Assuming that we are working with `invenio-stats` as our backend, we have the following API and we'd like to display retrieved data using the `invenio-statistics-js`.

All files

88.97% statements (280/315) 88.47% branches (26/30) 92.22% functions (2/22) 98.88% lines (280/285)

File	Statements	Branches	Functions	Lines
example/cluster	100%	100%	0%	100%
example/groupedBar	100%	100%	0%	100%
example/line	100%	100%	0%	100%
src	100%	100%	0%	100%
utils	75%	0%	44.76%	70%
webpack	88.44%	100%	24.0%	100%
webpackDev	87.54%	0%	82.34%	100%
webpack	88.57%	0%	46.88%	100%



#TODOS

- Additional styling
- Package and publish on NPM
- Pie, Country charts (?)



Questions?

Special thanks to:

Harris Tzovanakis

Sebastian Witowski

More info:

[https://github.com/CERNDocumentServer/
invenio-stats-js](https://github.com/CERNDocumentServer/invenio-stats-js)

androulidakis.ioannis@gmail.com

This template is free to use under [Creative Commons Attribution license](https://creativecommons.org/licenses/by/4.0/).

