A Beginner’s guide to Power BI Custom Visuals

Régis Baccaro

@regbac

About.me

Régis Baccaro

@regbac

regis@baccaro.com

http://theblobfarm.wordpress.com

Founder and lead organizer of SQL Saturday Denmark

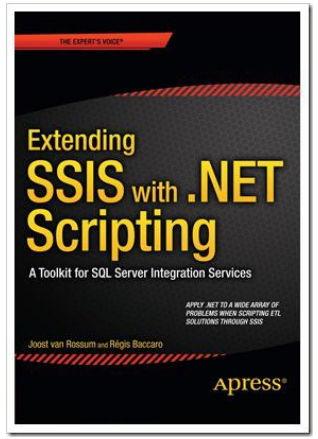
Works for Rehfeld Partners

Passionate about the community

.Net developer, BI guy, SharePoint fellow and accidental DBA

Data Platform MVP

Author



BIG Thanks to SQLSatMadrid Sponsors

4 Sponsor Sessions at 11:40

Don’t miss them, they might be getting distributing some **awesome prizes**!

- HPE
- SolidQ
- KABEL
- TSD Consulting

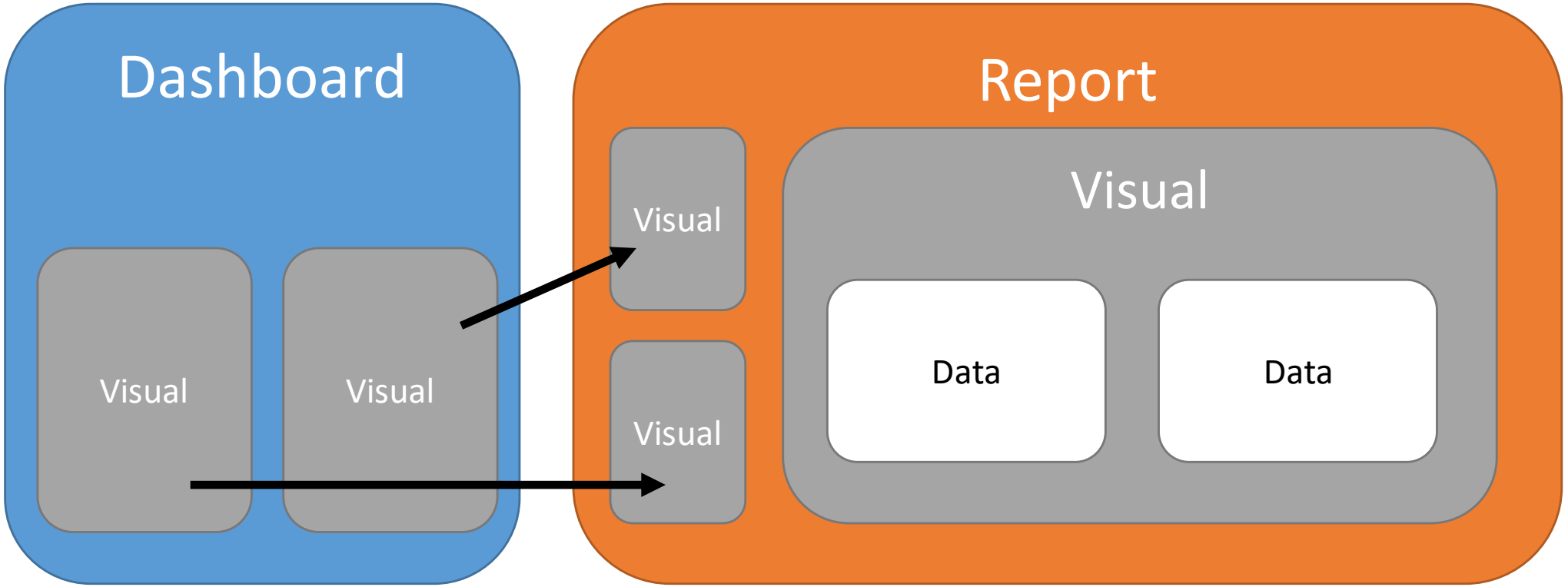
Also **BIG Raffle prizes** at the end of the event provided by:

Plainconcepts, SolidQ, Kabel, TSD Consulting, Pyramid Analytics & sqlpass.es

Agenda

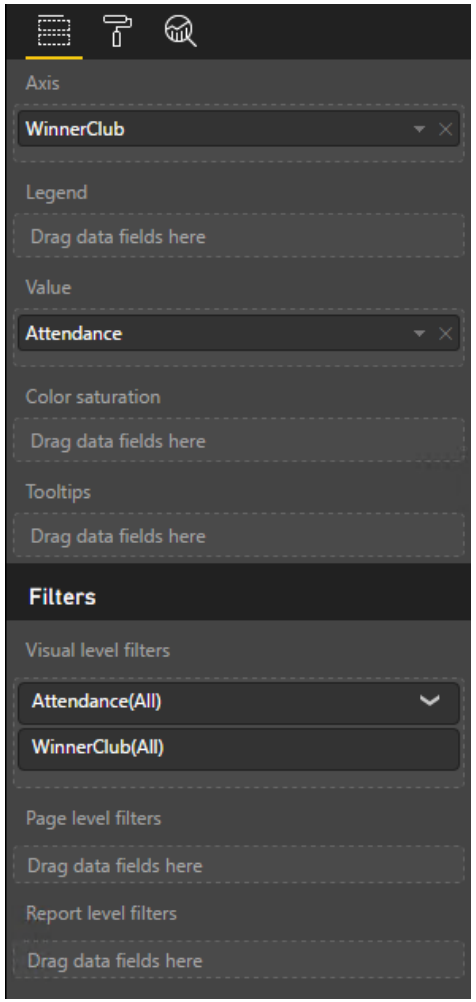
Intro to visuals
Why it is a good idea !
Getting custom visual
Setting up Environment
Visual lifecycle
Data binding
Building Bar Chart
Formatting

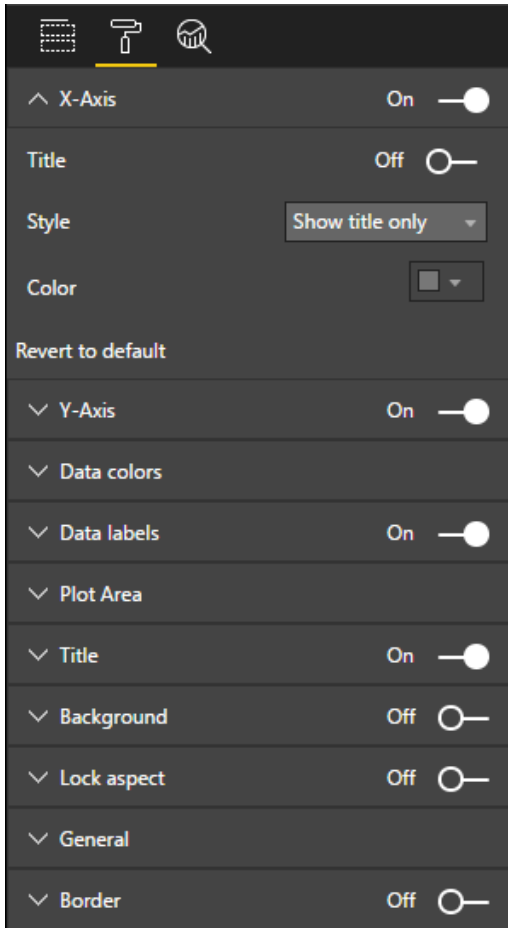
Power BI contains dashboards, reports, visuals and data



Visual Show info, allow interaction and provide insights

- Some types are bar charts, pie charts, maps, tables…
- Switch between types
- Modify the data and properties the visual uses





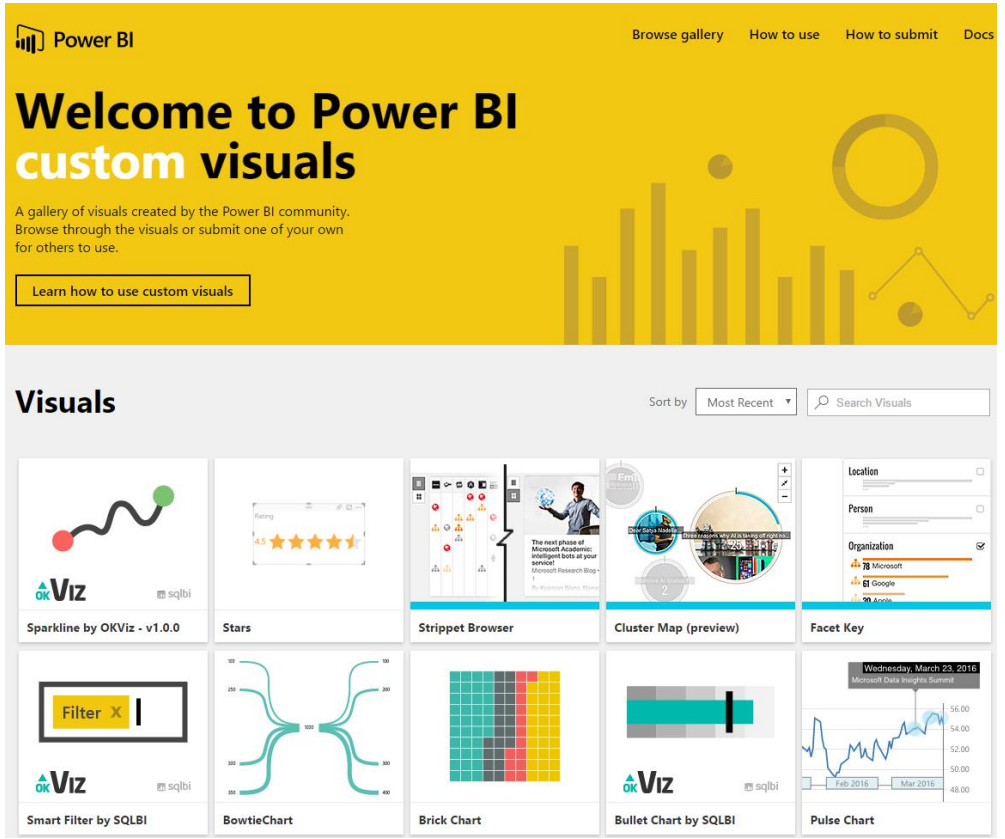
Make your own

- Microsoft open sourced the code for visuals
- You can use them in own reports and dashboards
- You can submit your visuals for inclusion in Power BI for others

Getting Custom Visuals

Available at the Power BI custom visuals gallery

http://app.powerbi.com/visuals



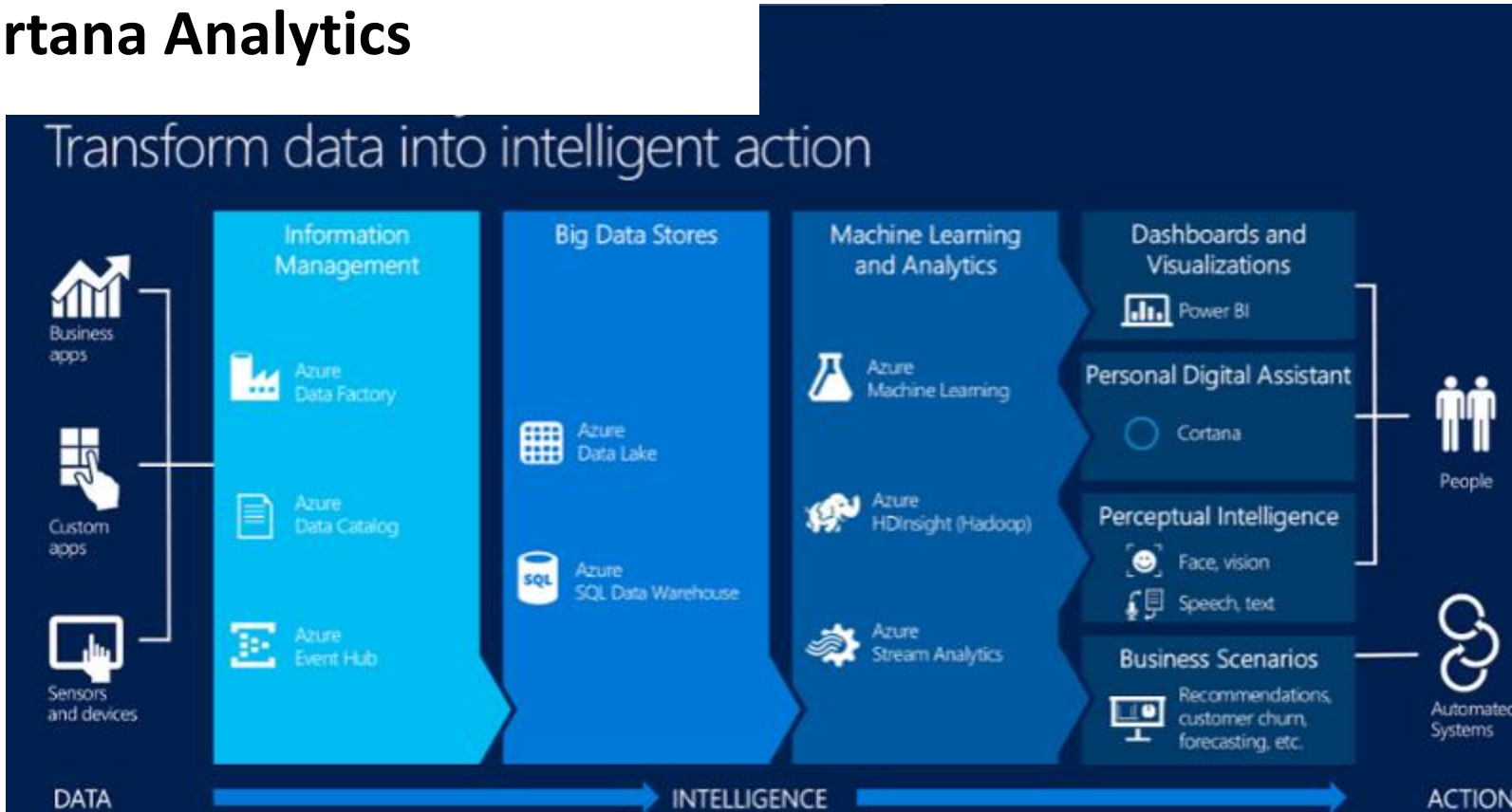
Is it a good idea ?

Separates Power BI from the Competition

Empowers Users Through the Gallery

Energizes the Power BI Development Community

Reflects Favorably on Cortana Analytics



Anatomy of a Visual Project

assets/

dist/

src/

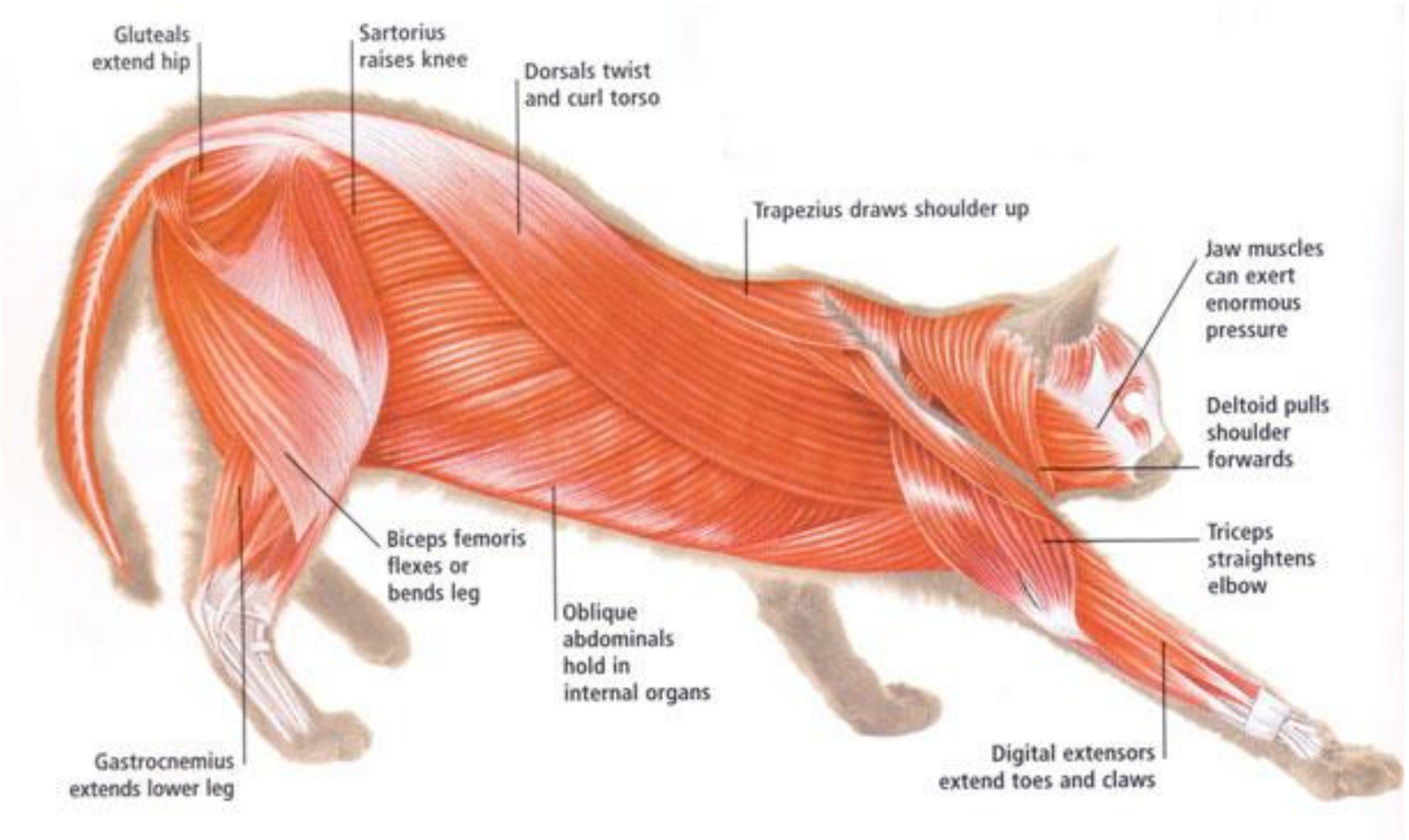
style/

.gitignore

capabilities.json

pbiviz.json

tsconfig.json



Setting up the environment – old API

Install Node.js

Install gulp if not present: `npm install -g gulp`

Clone a copy of the repo:

```
git clone https://github.com/Microsoft/PowerBI-visuals-core.git
```

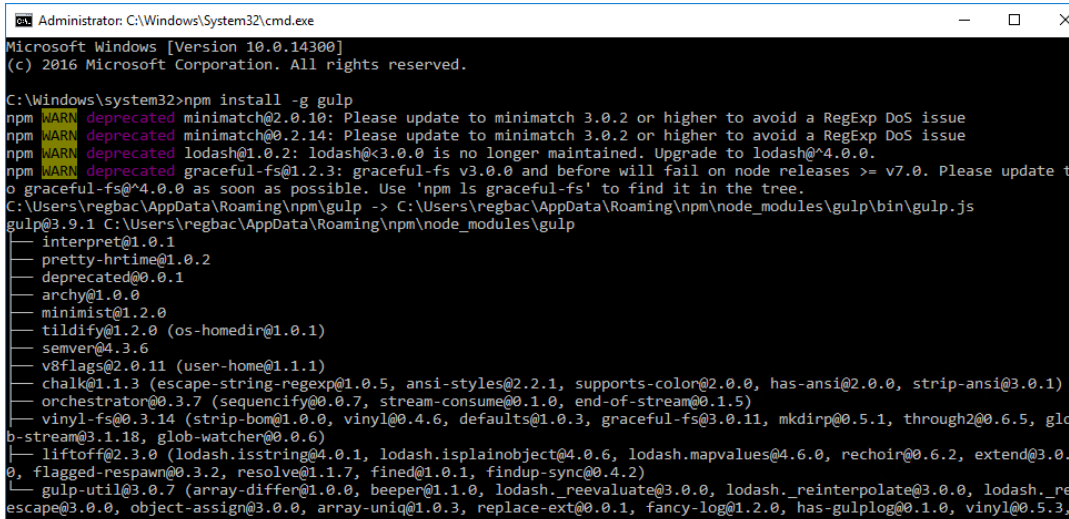
Change to the PowerBI-visuals-core directory:

```
cd PowerBI-visuals-core
```

Install dev dependencies:

```
npm install
```

Open Visual Studio Project



Running playground – old API

Open `src\PowerBIVisuals.sln`

In `src\Clients\PowerBIVisualsPlayground` select `standalone.html` as 'Set As Start Page'.

PowerBIVisualsPlayground→ 'Property Pages‘ → 'Build' →'Before running startup page' →'No Build'.

PowerBIVisualsPlayground →'Set as Startup Project'

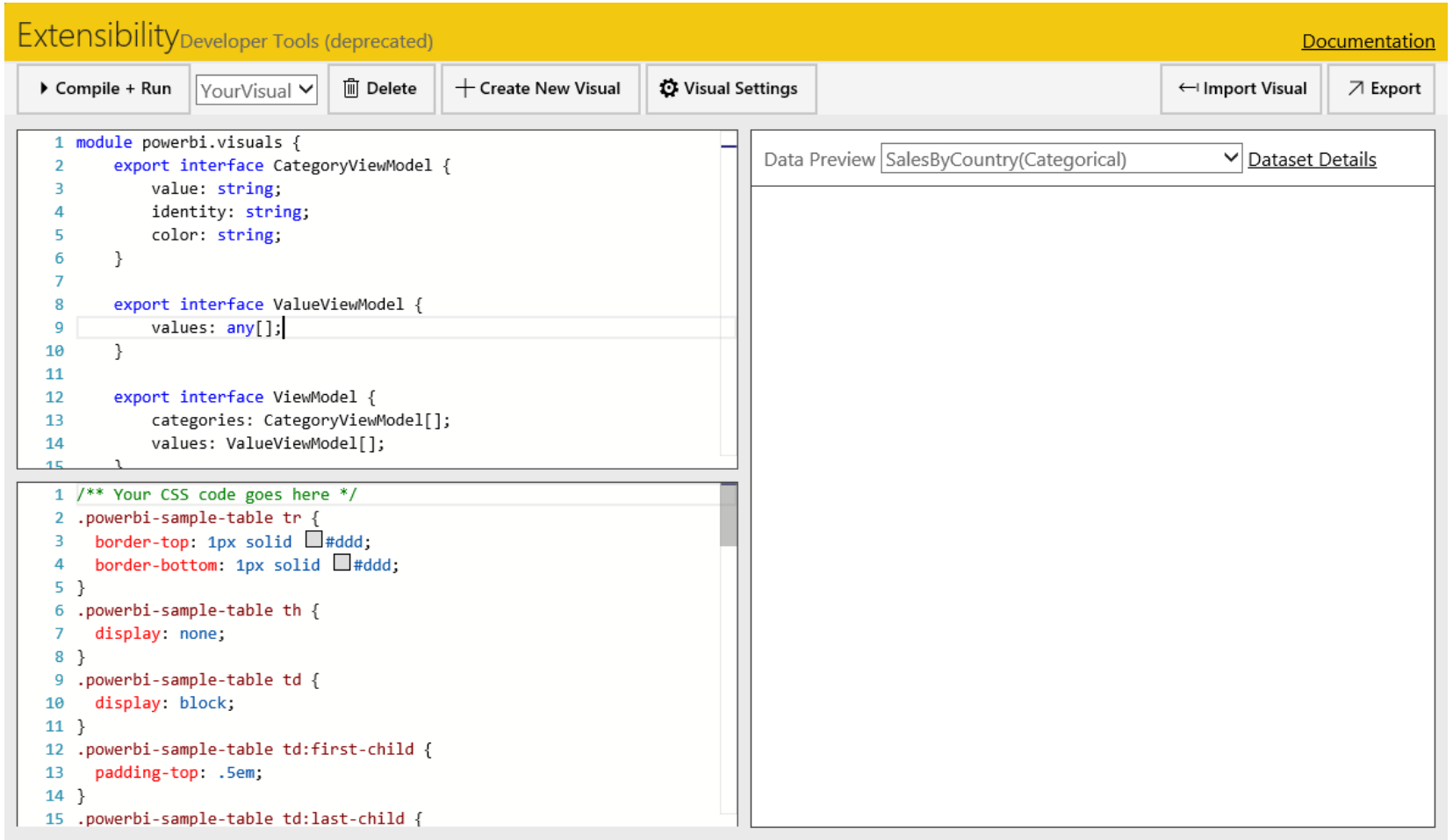
Run "build" task from "Task Runner Explorer" window.

Ctrl + F5 to launch the Playground.

Setting up the environment – web only API 1.0 (deprecated)

Enable dev tools on PowerBI.com

Try it out



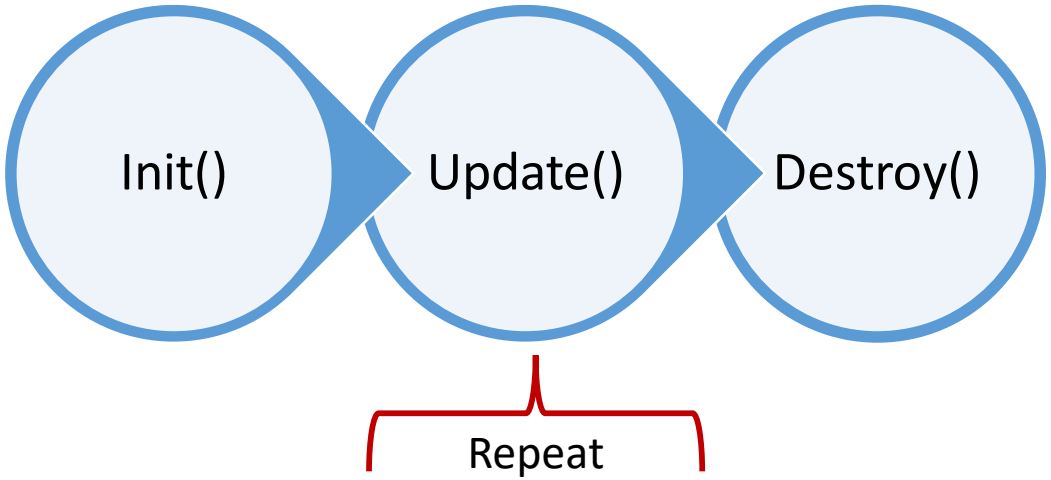
Three functions do the work

Init() – Sets up the visual when it is first created

Update() – Makes changes to view, data or style

Destroy() – Cleans up any remaining items when the visual is removed

Demo !!



Important objects provided in IVisual

DataView : Object model used to store a representation of the data

DataRoles : What fields the visual expects

DataViewMappings : What the fields look like

Static converter pattern : maps the DataView to a D3-friendly ViewModel “data” object

Setting up the environment - new

Install PowerBI visuals CLI tool - requires NodeJS
Install SSL certifications to enable live preview of visuals:

```
pbiviz --install-cert
```

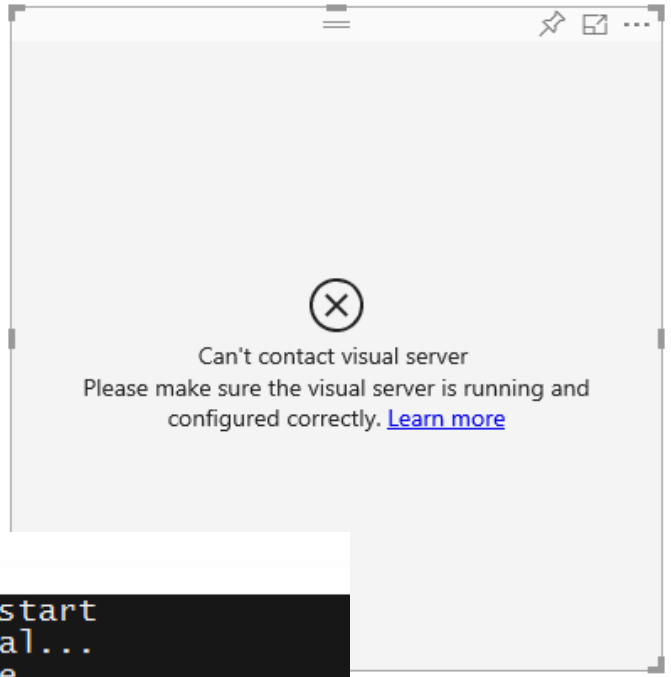
Enable developer tools in PowerBI
Create new PowerBI Visual Project

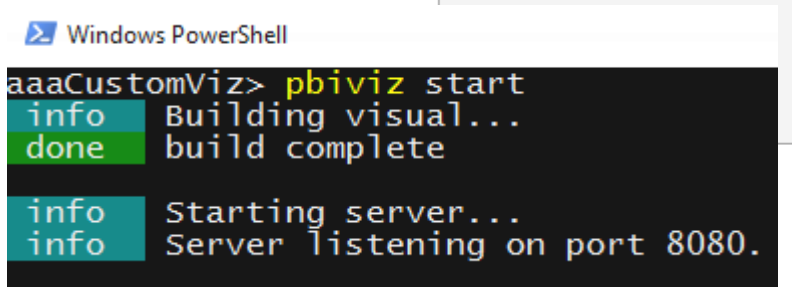
```
pbiviz new <visualName>
```

Start development server

```
pbiviz start
```

Updating
Adding external libraries
Installing typing for Libraries





We

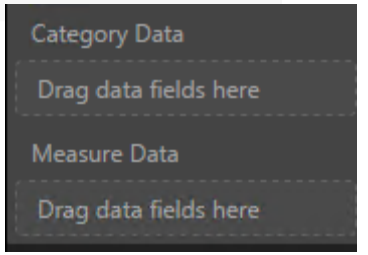
Cap

```
public enumerateObjectInstances(options: EnumerateVisualObjectInstancesOptions): VisualObjectInstanceEnumeration {
    let objectName = options.objectName;
    let objectEnumeration: VisualObjectInstance[] = [];

    switch(objectName) {
        case 'enableAxis':
            objectEnumeration.push({
                objectName: objectName,
                properties: {
                    show: this.barChartSettings.enableAxis.show,
                },
                selector: null
            });
            break;
        case 'colorSelector':
            for(let barDataPoint of this.barDataPoints) {
                objectEnumeration.push({
                    objectName: objectName,
                    displayName: barDataPoint.category,
                    properties: {
                        fill: {
                            solid: {
                                color: barDataPoint.color
                            }
                        }
                    },
                    selector: barDataPoint.selectionId.getSelector()
                });
            }
            break;
    };

    return objectEnumeration;
}
```

```
[ ... ],
ppings": [ ... ],
{ ... },
ghlight": true|false,
[ ... }
```



ings

```
Mappings": [

conditions": [ ... ],
categorical": { ... },
table": { ... },
single": { ... },
matrix": { ... }
```

Walkthro

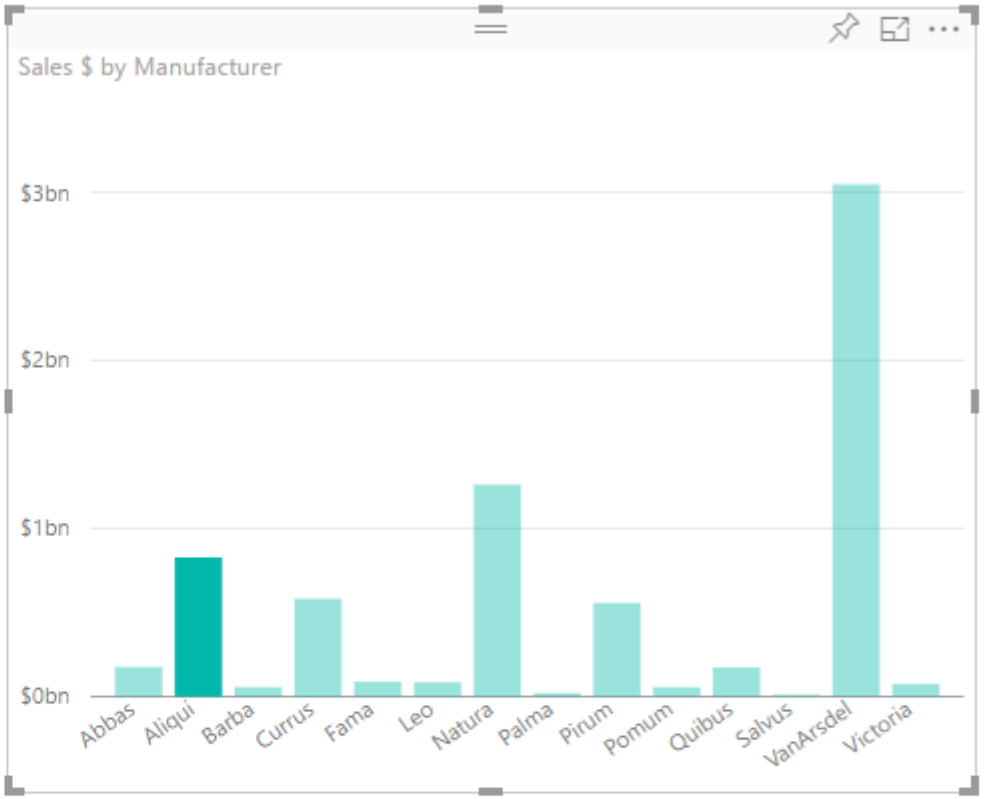
Capabilities

Handle p


```
⊖[
  ⊖{
      "metadata": ⊕{...},
      "categorical": ⊖{
          "categories": ⊕[ ... ],
          "values": ⊖[
              ⊖{
                  "source": ⊕{...},
                  "values": ⊖[
                      829176158.79
                  ],
                  "minLocal": 829176158.79,
                  "maxLocal": 829176158.79
              }
          ]
      },
      "tree": ⊕{...},
      "table": ⊕{...},
      "matrix": ⊕{...}
  }
]
```



```
⊖[
  ⊖{
      "metadata": ⊕{...},
      "categorical": ⊖{
          "categories": ⊕[ ... ],
          "values": ⊖[
              ⊖{
                  "source": ⊕{...},
                  "values": ⊖[
                      176647835.07,
                      829176158.79,
                      57150048.06,
                      583361169.3,
                      88824675.87,
                      85720374.18,
                      1263093451.2,
                      21241165.68,
                      558489013.11,
                      56201031.81,
                      173725443.36,
                      4586284.08,
                      3051157508.91,
                      75035665.53
                  ],
                  "minLocal": 4586284.08,
                  "maxLocal": 3051157508.91,
                  "highlights": ⊖[
                      null,
                      829176158.79,
                      null,
                      null,
                      null,
                      null,
                      null,
                      null,
                      null,
                      null,
                      null,
                      null,
                      null,
                      null,
                      null,
                      null
                  ]
              }
          ]
      },
```

Walkthrough – building a bar chart

Defining and Using visualTransform

Convert DataView into a view model

Adding color

```
function visualTransform(options: VisualUpdateOptions, host: IVisualHost): BarChartViewModel {
    let dataViews = options.dataViews;
    let defaultSettings: BarChartSettings = {
/**
 * Interface for BarChart data points.
 *
 * @interface
 * @property {number} value             - Data value for point.
 * @property {string} category          - Corresponding category of data value.
 * @property {string} color             - Color corresponding to data point.
 * @property {ISelectionId} selectionId - Id assigned to data point for cross filtering
 *                                        and visual interaction.
 */
interface BarChartDataPoint {
    value: number;
    category: string;
    color: string;
    selectionId: ISelectionId;
};
```

Walkthrough – building a bar chart

Adding Selection and Interactions with Other Visuals

- Adding Selection to Each Data Point
- Assigning Selection Ids to Each Data Point
- Interacting with your Data Points

```
let selectionManager = this.selectionManager;

//This must be an anonymous function instead of a lambda because
//d3 uses 'this' as the reference to the element that was clicked.
bars.on('click', function(d) {
    selectionManager.select(d.selectionId).then((ids: ISelectionId[]) => {
        bars.attr({
            'fill-opacity': ids.length > 0 ? BarChart.Config.transparentOpacity : BarChart.Config.solidOpacity
        });

        d3.select(this).attr({
            'fill-opacity': BarChart.Config.solidOpacity
        });
    });

    (<Event>d3.event).stopPropagation();
});
```

Debugging

Using breakpoints

Showing Exceptions

Log exceptions

Break on exceptions

Edge

Chrome

```
public update(options: VisualUpdateOptions) {
                console.log('Visual update', options);
                debugger;
                this.target.innerHTML = ‘<p>Update count: <em>${(this.updateCount++)}</em></p>’;
}
```

```
module powerbi.extensibility.visual {
    export function logExceptions(): MethodDecorator {
        return function (target: Object, propertyKey: string, descriptor: TypedPropertyDescriptor<Function>)
        : TypedPropertyDescriptor<Function> {

            return {
                value: function () {
                    try {
                        return descriptor.value.apply(this, arguments);
                    } catch (e) {
                        console.error(e);
                        throw e;
                    }
                }
            }
        }
    }
}
```

```
@logExceptions()
public update(options: VisualUpdateOptions) {
```

Packaging for distribution

```
pbiviz package
```

References

Github for Custom Visuals : https://github.com/Microsoft/PowerBI-visuals

Dev tools (the deprecated) : https://app.powerbi.com/devTools

Custom Visuals gallery : http://app.powerbi.com/visuals

Questions