

Building Web App with Polymer

Tomislav Priselac

04_2018



About Company



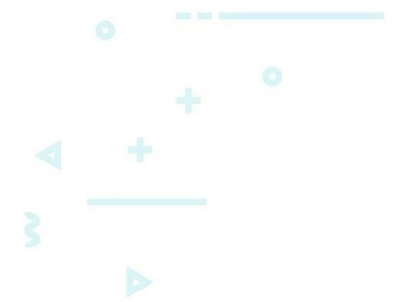
- Deegloo (Digital Igloo)
- Founded in May, 2017
- 9 employees
- Headquarters: Gundulićeva ul. 48, 10000, Zagreb

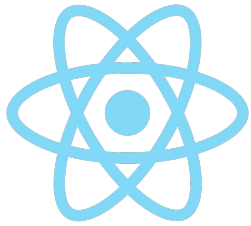


```

<div id="carouselExampleControls" class="carousel slide" data-ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
  <a class="carousel-control-prev" href="#carouselExampleControls" role="button"
data-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
  </a>
  <a class="carousel-control-next" href="#carouselExampleControls" role="button"
data-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
  </a>
</div>

```





```
<my-slider> </my-slider>
```

Disadvantage: **Interoperability**



deegloo
DIGITAL IGLOO

Web Components



deegloo
DIGITAL IGLOO

Standards/APIs

Custom Elements

Shadow DOM

HTML Imports

HTML Templates



deegloo
DIGITAL IGLOO

Custom elements

Define custom HTML tag

<my-element></my-element>

Rules:

- Name must contain a dash
- Can't register same name tag more than once
- Can't be self-closing

```
class MyComponent extends HTMLElement { }  
window.customElements.define("my-component", MyComponent);
```



deegloo
DIGITAL IGLOO

Shadow DOM



Encapsulates component logic in shadows

Nothing leaks out(styling, scripts...)

```
this.attachShadow({ mode: 'open' });
```



deegloo
DIGITAL IGLOO

Shadow DOM



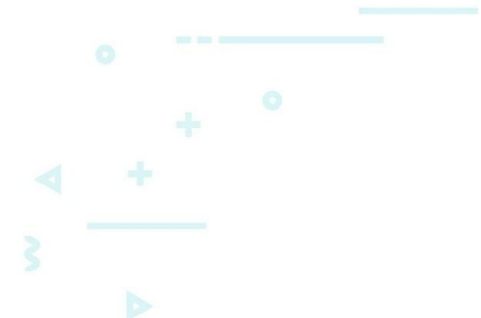
```
▼ <mdd-card _ngcontent-c6 ripple type="indicator">  
  ► #shadow-root (open)  
  </mdd-card>  
  ...
```

```
document.querySelector('mdd-card').querySelector('div') =>  
returns NULL
```

```
document.querySelector('mdd-card').shadowRoot.querySelector  
( 'div' ) => returns DIV element
```



HTML Imports



A way to include and reuse HTML documents in other HTML documents.

```
<link rel="import" href="/imports/my-component.html">
```

HTML Template

It is defined by template tag: `<template> </template>`










Inside this tag you can write HTML & CSS to define how your component should be presented in browser

```
<template>
  <style>
    h1 { color: red; }
  </style>
  <div>
    <h1> </h1>
  </div>
</template>
```



deegloo
DIGITAL IGLOO

Browser Support

	 CHROME	 OPERA	 SAFARI	 FIREFOX	 EDGE
 TEMPLATES	✓ STABLE	✓ STABLE	✓ STABLE	✓ STABLE	✓ STABLE
 CUSTOM ELEMENTS	✓ STABLE	✓ STABLE	✓ STABLE	✓ POLYFILL ● DEVELOPING	✓ POLYFILL ● CONSIDERING
 SHADOW DOM	✓ STABLE	✓ STABLE	✓ STABLE	✓ POLYFILL ● DEVELOPING	✓ POLYFILL ● CONSIDERING
 HTML IMPORTS	✓ STABLE	✓ STABLE	✓ POLYFILL ● ON HOLD	✓ POLYFILL ● ON HOLD	✓ POLYFILL ● CONSIDERING

Data flow



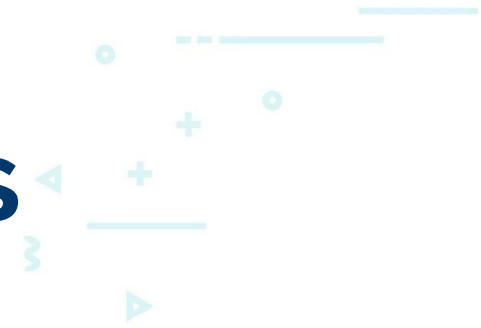
Outside In => Attributes & Parameters

```
<my-component text="Testing"> </my-component>
get text() { return this.getAttribute('text'); }
set text(val) { this.setAttribute('text', val); }
```

Inside Out => Events

```
this.dispatchEvent(new CustomEvent('something-changed', {
  detail: {
    ...
  }
}));
```

Lifecycle Events



- **connectedCallback()**
 - element is attached to the DOM
- **disconnectedCallback()**
 - element is removed from the DOM
- **attributeChangedCallback()**
 - when attribute is changed
 - only fires for observed attributes
 - `static get observedAttributes() { return ['text']; }`



```

<template>
  <style>
    h1 { color: red; font-size: 22px; }
  </style>
  <div> <h1> </h1> </div>
</template>

<script>
  const tmpl = (document.currentScript ||
document._currentScript).ownerDocument.querySelector('template').content;
  class MyHeading extends HTMLElement {
    constructor() {
      super();
      this.attachShadow({ mode: 'open' })
      const content = document.importNode(tmpl, true)
      this.shadowRoot.appendChild(content)
    }
    get text() { return this.getAttribute('text'); }
    set text(val) { this.setAttribute('text', val); }
    static get observedAttributes() { return ['text'] };
    attributeChangedCallback(name, oldVal, newVal) { this.update(); }
    update() { this.shadowRoot.querySelector('h1').textContent = this.text; }
  }
  window.customElements.define("my-heading", MyHeading);
</script>

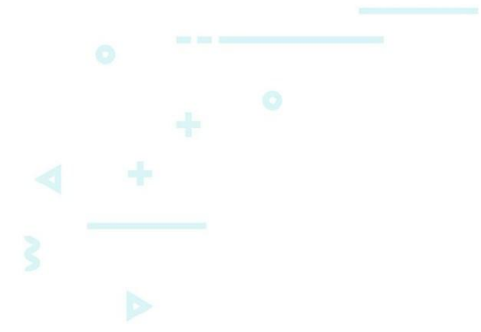
```



deegloo
DIGITAL IGLOO



deegloo
DIGITAL IGLOO



Polymer



deegloo
DIGITAL IGLOO

Polymer



- Google project
- Syntax sugar on top of Web Components
- Lightweight
- Provides set of built-in components
- Bower

#UseThePlatform

Benefits



- Interoperability
- Removes boilerplate code
- Data binding
- Loads polyfills for you

Properties



- type
- value
- reflectToAttribute
- readOnly
- notify
- computed
- observer

```
static get properties() {  
  return {  
    text: {  
      type: String,  
      value: 'some value',  
      reflectToAttribute: true,  
      readOnly: true,  
      notify: true,  
      computed: 'computeFullName(first, last)',  
      observer: '_textChanged'  
    }  
  };  
}
```



Native vs Polymer - 1 (Shadow DOM)

```
<template>
  <style> h1 { color: red; } </style>
  <div> <h1> </h1> </div>
</template>
```

Native

```
const tmpl = document.querySelector('template').content;
this.shadow = this.attachShadow({ mode: 'open' });
const content = document.importNode(tmpl, true);
this.shadowRoot.appendChild(content);
```

Polymer

Out of the box!

Native vs Polymer - 2 (Properties)

Native

```
get text() { return this.getAttribute('text'); }  
set text(val) { this.setAttribute('text', val); }  
static get observedAttributes() { return ['text'] };  
attributeChangedCallback(name, oldVal, newVal) { }
```

Polymer

```
static get properties() {  
  return {  
    text: {  
      type: String,  
      reflectToAttribute: true,  
      observe: '_textChanged'  
    }  
  };  
}
```



Native vs Polymer - 3 (Updating DOM)

Native

```
this.shadowRoot.querySelector('h1').textContent = this.text;
```

Polymer

```
<template>  
  <style> h1 { color: red; } </style>  
  <div> <h1> [[text]] </h1> </div>  
</template>
```

```
<link rel="import" href="../polymer/polymer-element.html">

<dom-module id="my-heading">
  <template>
    <style>
      h1 { color: red; font-size: 22px; }
    </style>
    <h1>[[text]]</h1>
  </template>
  <script>
    class MyHeading extends Polymer.Element {
      static get is() { return 'my-heading'; }
      static get properties() {
        return {
          text: String
        };
      }
    }

    window.customElements.define(MyHeading.is(), MyHeading);
  </script>
</dom-module>
```



deegloo
DIGITAL IGLOO

Web Application

- My Dairy Dashboard
- Angular 4 + Polymer
- USA Dairy Industry

My Dairy
DASHBOARD



deegloo
DIGITAL IGLOO



PDS Manager
a@pds.com



Dashboards

SmartView

Settings



Account



Company



Herds



Billing



Support

My Dairy DASHBOARD

SmartView

Herd

WGD LIVE!

lb Milk/Cow

102

yesterday

% Fat

4.24

yesterday

% Protein

3.30

yesterday

Lactating DMI

53

on Mar 26

Close-up DMI

35

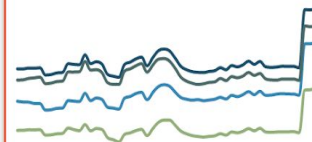
on Mar 26

SCC yesterday



65

Milk/Cow & ECM/Cow



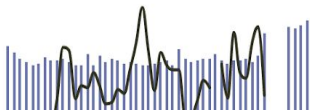
102 lb Milk/Cow
yesterday

Udder Health



2.16 % New Infection
last month

Close-up DMI



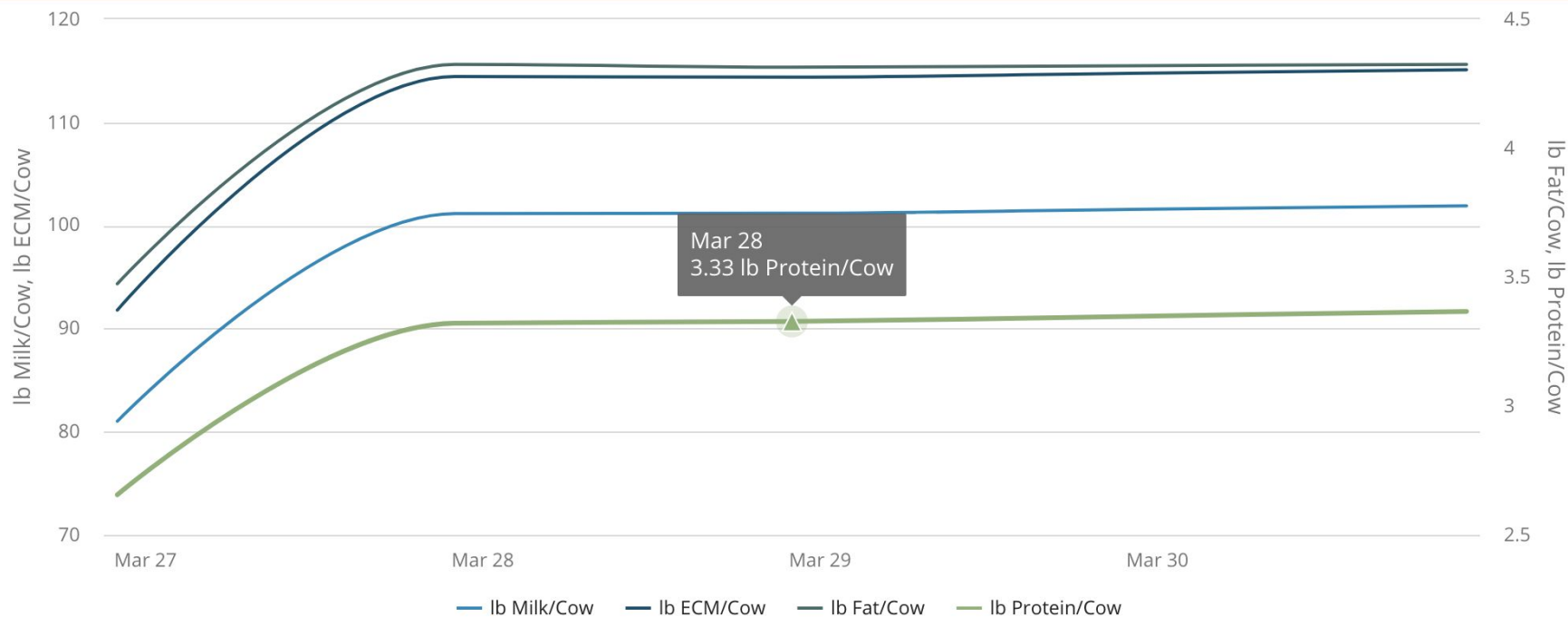
Peak Milk



deegloo
DIGITAL IGLOO



Milk/Cow & ECM/Cow



Date range

Last Days Days ago

Aggregation

4-Day Average

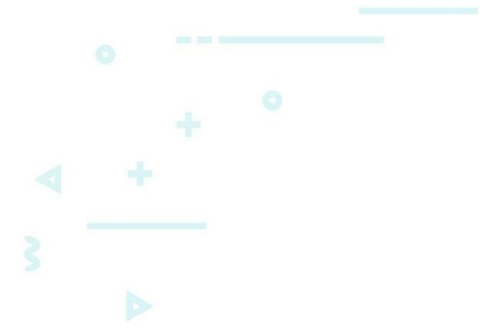
SAVE CHANGES

SAVE AS COPY



deegloo
DIGITAL IGLOO

Why Polymer?



Business requirement

- **Possibility of embedding dashboard into other applications**

Architecture



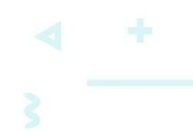
2 repositories

- **mdd-portal-ui**
 - Angular 4 + Polymer
- **mdd-elements**
 - custom built web components using Polymer



deegloo
DIGITAL IGLOO

Add custom components through bower



```
1 {
2   "name": "mdd-portal",
3   "private": true,
4   "dependencies": {
5     "polymer": "Polymer/polymer#^2.0.0",
6     "app-layout": "PolymerElements/app-layout#^2.0.0",
7     "paper-icon-button": "PolymerElements/paper-icon-button#^2.0.0",
8     "iron-icons": "PolymerElements/iron-icons#^2.0.0",
9     "paper-card": "PolymerElements/paper-card#^2.0.0",
10    "paper-styles": "PolymerElements/paper-styles#^2.0.0",
11    "paper-dialog": "PolymerElements/paper-dialog#^2.0.0",
12    "paper-button": "PolymerElements/paper-button#^2.0.0",
13    "paper-dialog-scrollable": "PolymerElements/paper-dialog-scrollable#^2.1.0",
14    "paper-item": "PolymerElements/paper-item#^2.0.0",
15    "mdd-elements": "git+ssh://git-codecommit.us-east-1.amazonaws.com/v1/repos/mdd-elements#develop"
16  },
17  "version": "1.4.1"
18 }
19
```



Register custom elements schema to module

```
import { CUSTOM_ELEMENTS_SCHEMA, NgModule } from '@angular/core';

@NgModule({
  schemas: [CUSTOM_ELEMENTS_SCHEMA],
})

export class SomeModule { }
```



Import custom component

bower.json **card-sheet.component.ts** ✕

```
1  import {
2    ChangeDetectionStrategy,
3    ChangeDetectorRef,
4    Component,
5    ElementRef,
6    HostListener,
7    OnInit,
8    ViewChild
9  } from '@angular/core';
10 import { ActivatedRoute, Router } from '@angular/router';
11 import { DashboardCard, DashboardCardSeries } from '@mdd/api';
12 import debounce from 'lodash-es/debounce';
13 import { BehaviorSubject } from 'rxjs/BehaviorSubject';
14
15 import 'app-layout/app-toolbar/app-toolbar.html';
16 import 'mdd-elements/lib/highcharts.html';
17 import 'mdd-elements/mdd-chart/mdd-chart.html';
18 import 'mdd-elements/mdd-dynamic-date-filter/mdd-dynamic-date-filter.html';
19 import 'paper-button/paper-button.html';
```



deegloo
DIGITAL IGLOO

Render component

```
<app-toolbar class="auto-height">
  <mdd-dynamic-date-filter *ngIf="(dynamicDateFilter | async)" class="inline-for-large-displays"
    label="Date range"
    [offset]="(dynamicDateFilter | async)?.offset"
    [count]="(dynamicDateFilter | async)?.count"
    [countLevel]="(dynamicDateFilter | async)?.countLevel"
    [offsetLevel]="(dynamicDateFilter | async)?.offsetLevel"
    (mdd-filter-changed)="dateFilterChanged($event.detail)"
    [projected]="isCardProjected()">
  </mdd-dynamic-date-filter>
</app-toolbar>
```



Date range

Last Days starting Days ago

Aggregation

SAVE CHANGES

SAVE AS COPY



deegloo
DIGITAL IGLOO

Polymer 3



Bower => NPM

HTML Imports => ES6 Modules



deegloo
DIGITAL IGLOO



deegloo
DIGITAL IGLOO

Thank you

