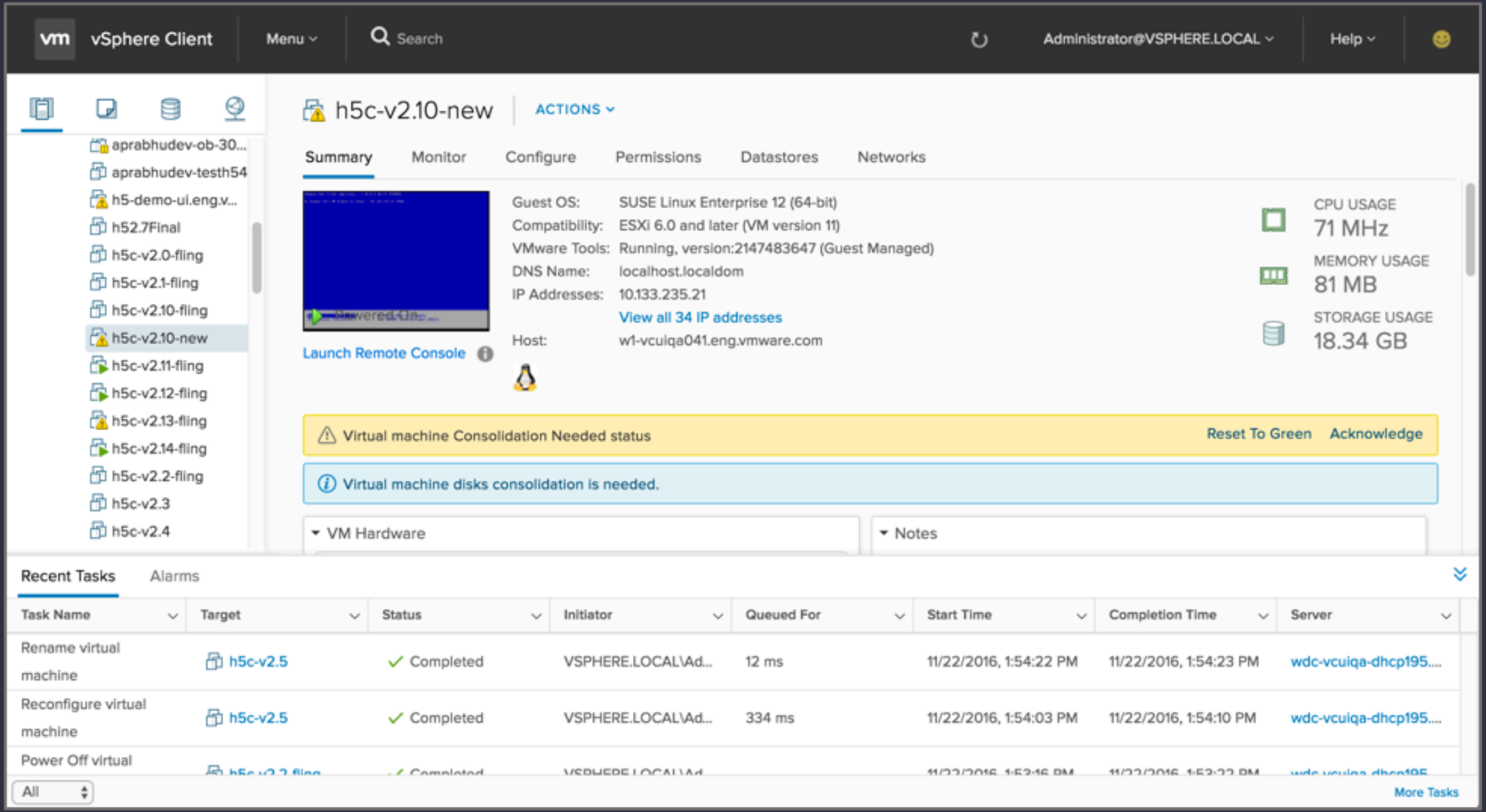
vmware

Clarity Design System

Inspiring builders to create better experiences

How do we empower designers and engineers to build products that are beautiful and functional?



h5c-v2.10-new

Guest OS:	SUSE Linux Enterprise 12 (64-bit)
Compatibility:	ESXi 6.0 and later (VM version 11)
VMware Tools:	Running, version:2147483647 (Guest Managed)
DNS Name:	localhost.localdom
IP Addresses:	10.133.235.21
	View all 34 IP addresses
Host:	w1-vcuiqa041.eng.vmware.com

CPU USAGE 71 MHz
MEMORY USAGE 81 MB
STORAGE USAGE 18.34 GB

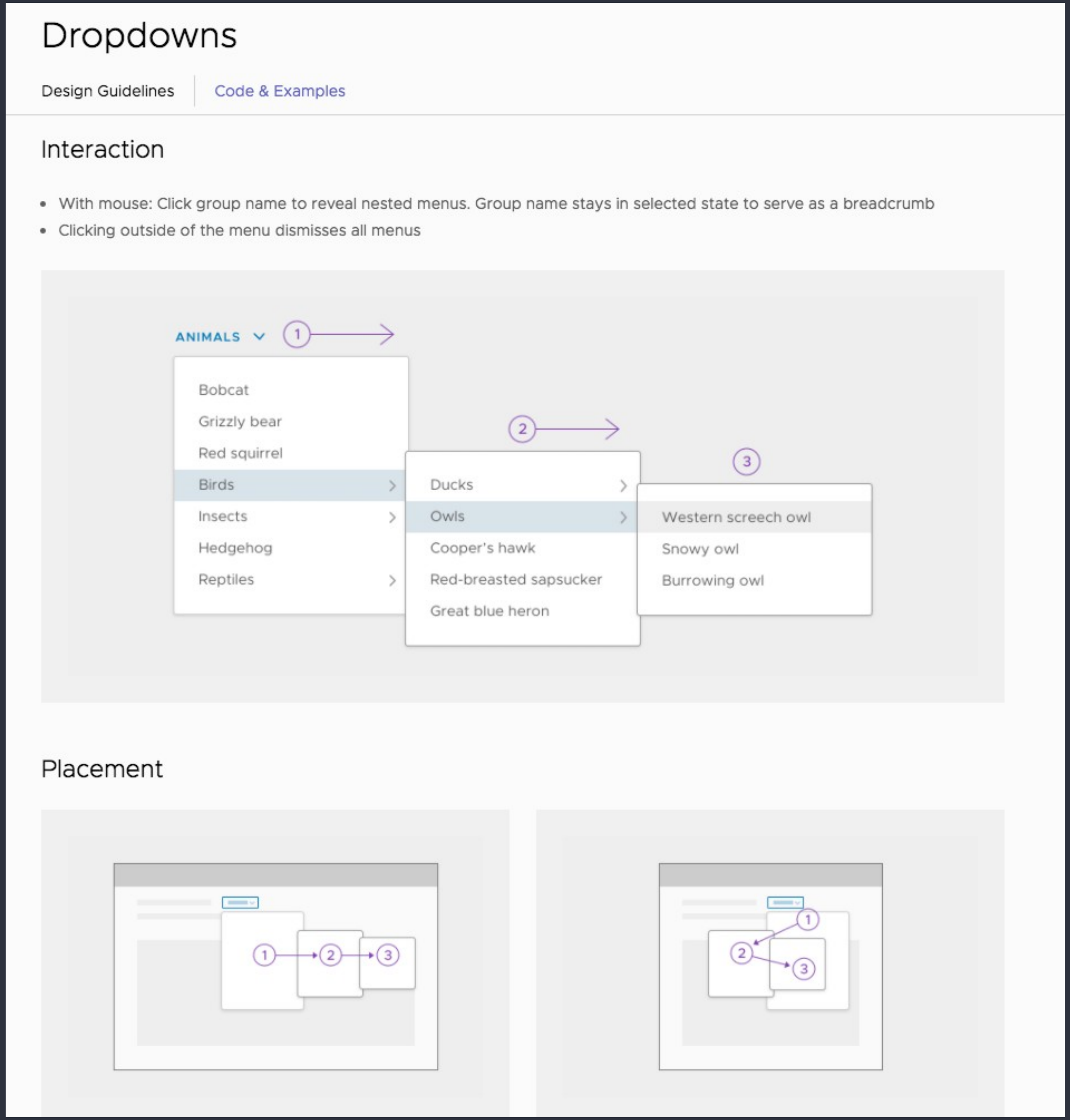
Virtual machine Consolidation Needed status

Virtual machine disks consolidation is needed.

Task Name	Target	Status	Initiator	Queued For	Start Time	Completion Time	Server
Rename virtual machine	h5c-v2.5	✓ Completed	VSPHERE.LOCAL\Ad...	12 ms	11/22/2016, 1:54:22 PM	11/22/2016, 1:54:23 PM	wdc-vcuiqa-dhcp195....
Reconfigure virtual machine	h5c-v2.5	✓ Completed	VSPHERE.LOCAL\Ad...	334 ms	11/22/2016, 1:54:03 PM	11/22/2016, 1:54:10 PM	wdc-vcuiqa-dhcp195....
Power Off virtual	h5c-v2.2-fling	✓ Completed	VSPHERE.LOCAL\Ad...		11/22/2016, 1:53:16 PM	11/22/2016, 1:53:22 PM	wdc-vcuiqa-dhcp195

vSphere Client (HTML5)

Clarity - Inspiring builders to create better experiences



Dropdowns

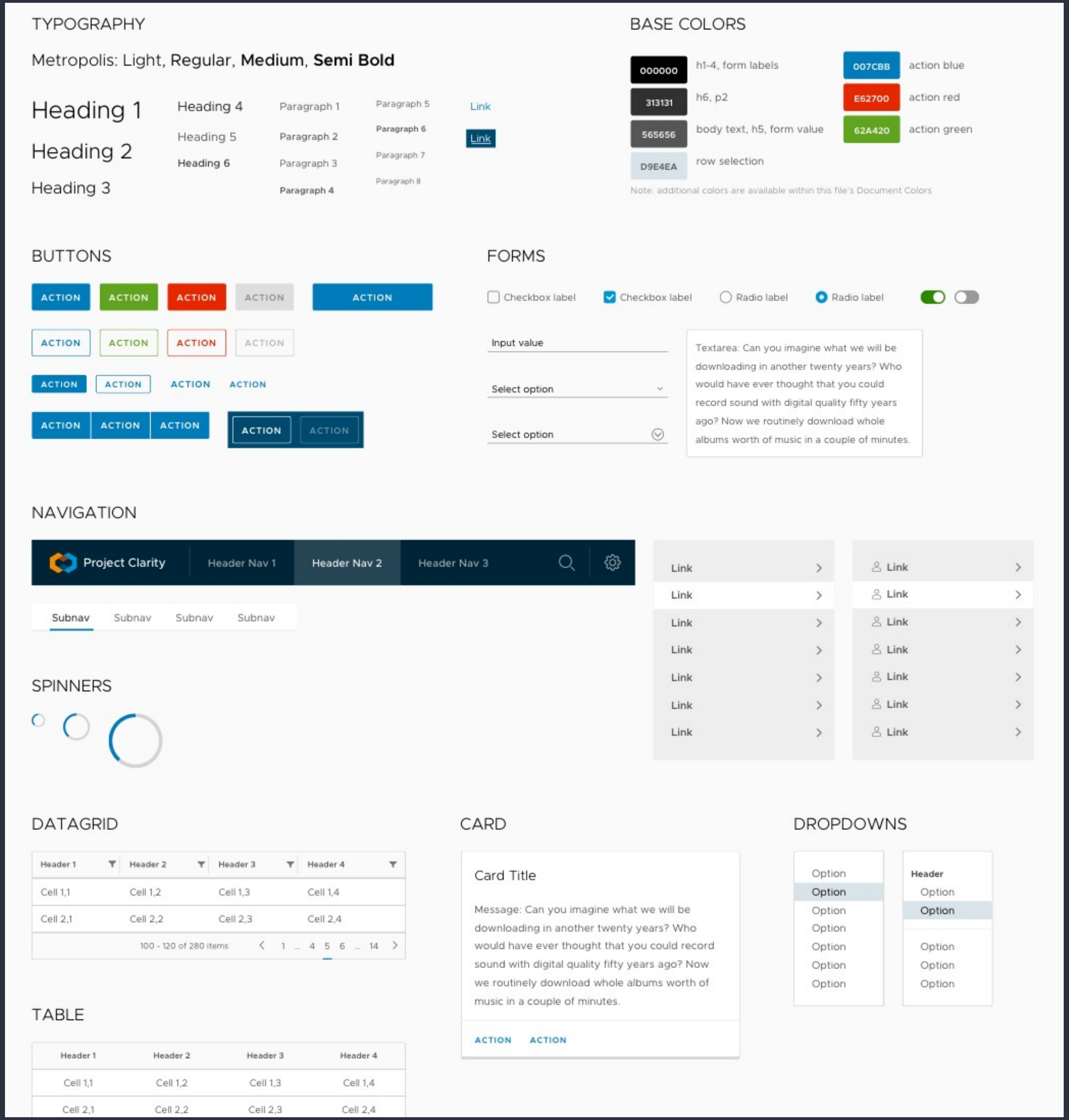
Design Guidelines | Code & Examples

Interaction

- With mouse: Click group name to reveal nested menus. Group name stays in selected state to serve as a breadcrumb
- Clicking outside of the menu dismisses all menus

Placement

Design Guidelines



TYPOGRAPHY

Metropolis: Light, Regular, **Medium**, **Semi Bold**

BASE COLORS

000000	h1-4, form labels	007CBB	action blue
313131	h6, p2	E62700	action red
565656	body text, h5, form value	62A420	action green
D9E4EA	row selection		

Note: additional colors are available within this file's Document Colors

BUTTONS

FORMS

Textarea: Can you imagine what we will be downloading in another twenty years? Who would have ever thought that you could record sound with digital quality fifty years ago? Now we routinely download whole albums worth of music in a couple of minutes.

NAVIGATION

SPINNERS

DATAGRID

Header 1	Header 2	Header 3	Header 4
Cell 1,1	Cell 1,2	Cell 1,3	Cell 1,4
Cell 2,1	Cell 2,2	Cell 2,3	Cell 2,4

100 - 120 of 280 items 1 ... 4 5 6 ... 14

TABLE

Header 1	Header 2	Header 3	Header 4
Cell 1,1	Cell 1,2	Cell 1,3	Cell 1,4
Cell 2,1	Cell 2,2	Cell 2,3	Cell 2,4

CARD

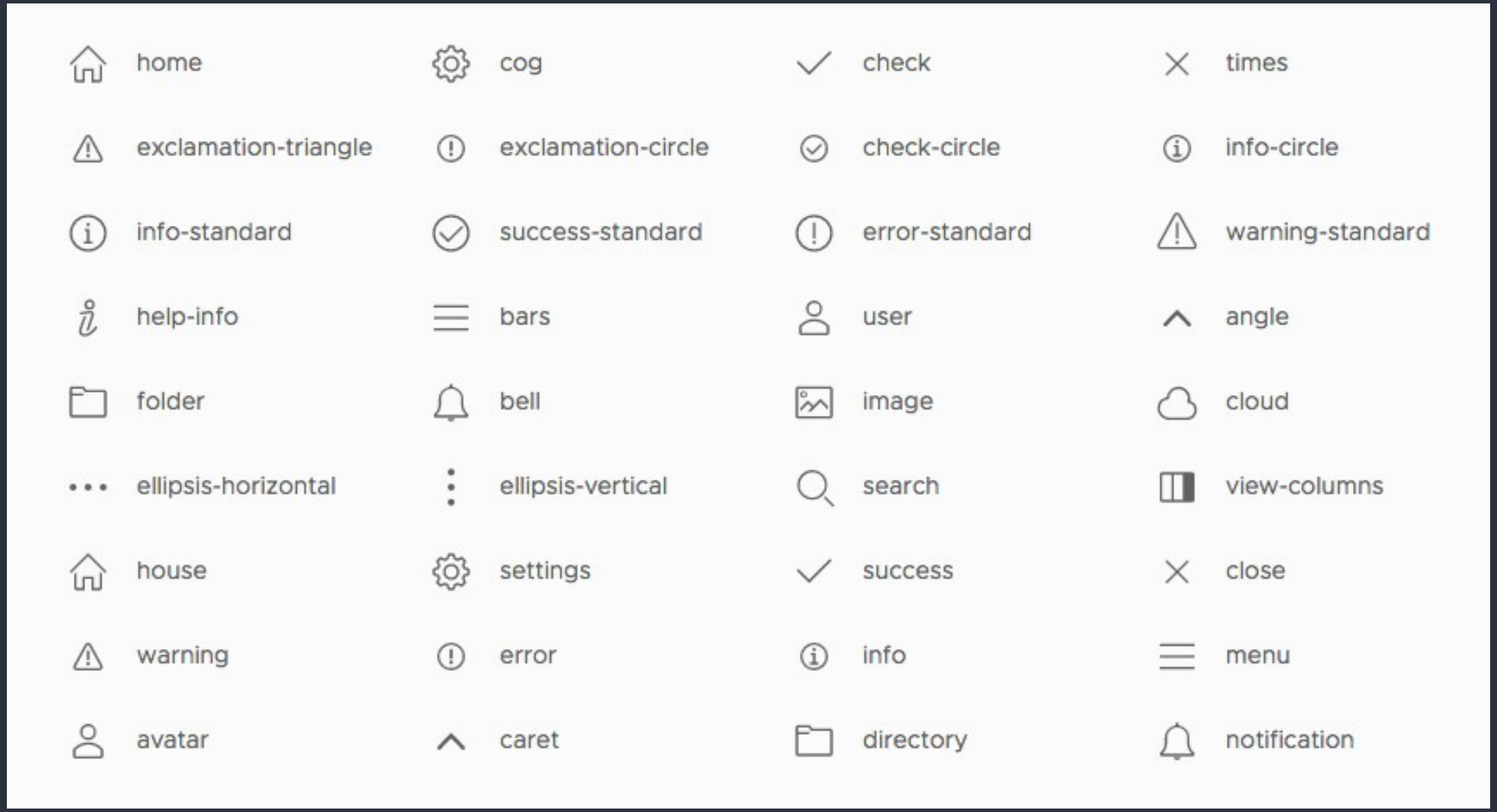
Card Title

Message: Can you imagine what we will be downloading in another twenty years? Who would have ever thought that you could record sound with digital quality fifty years ago? Now we routinely download whole albums worth of music in a couple of minutes.

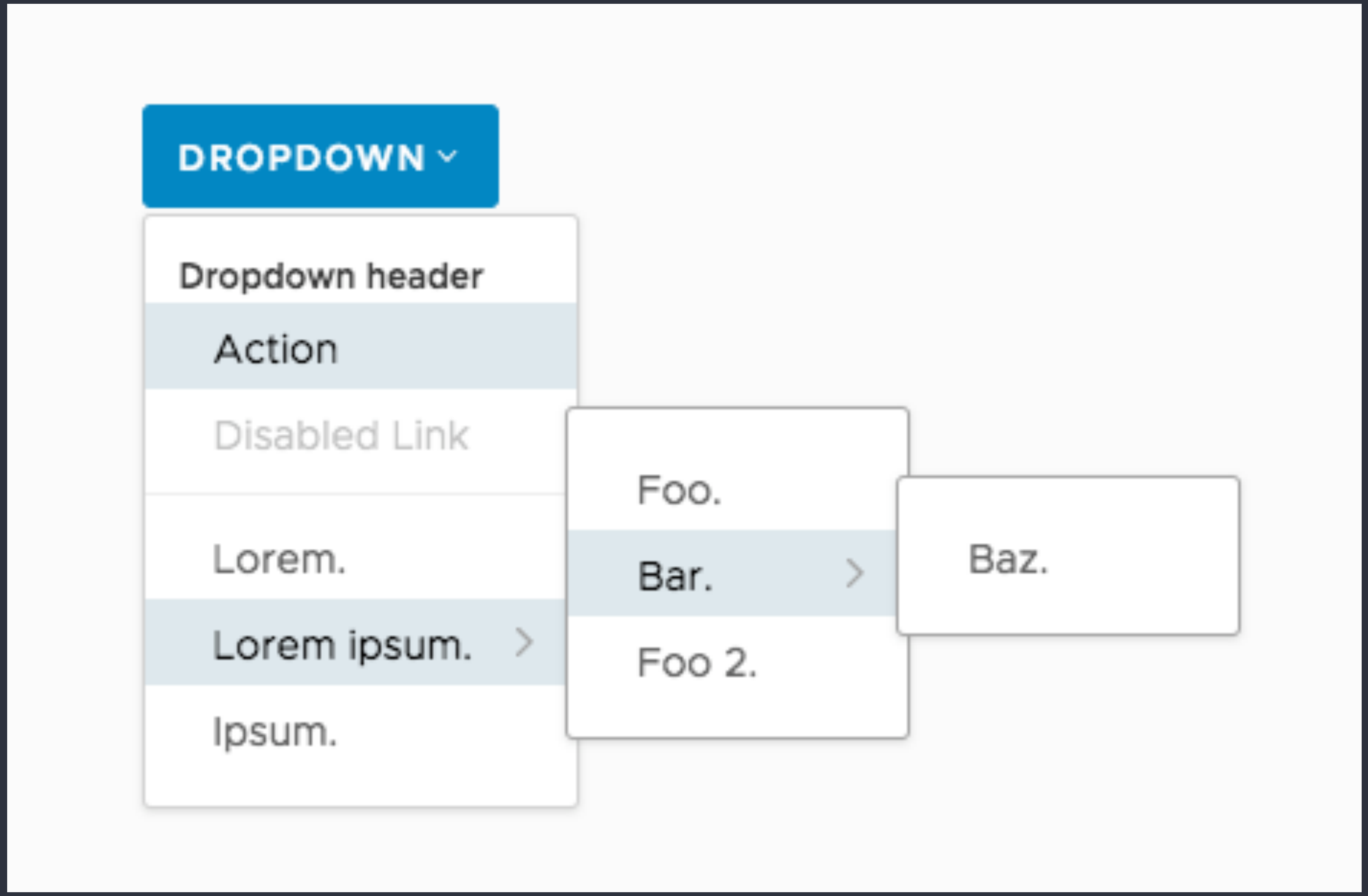
DROPDOWNS

Sketch Templates

Clarity - Inspiring builders to create better experiences



SVG Icons

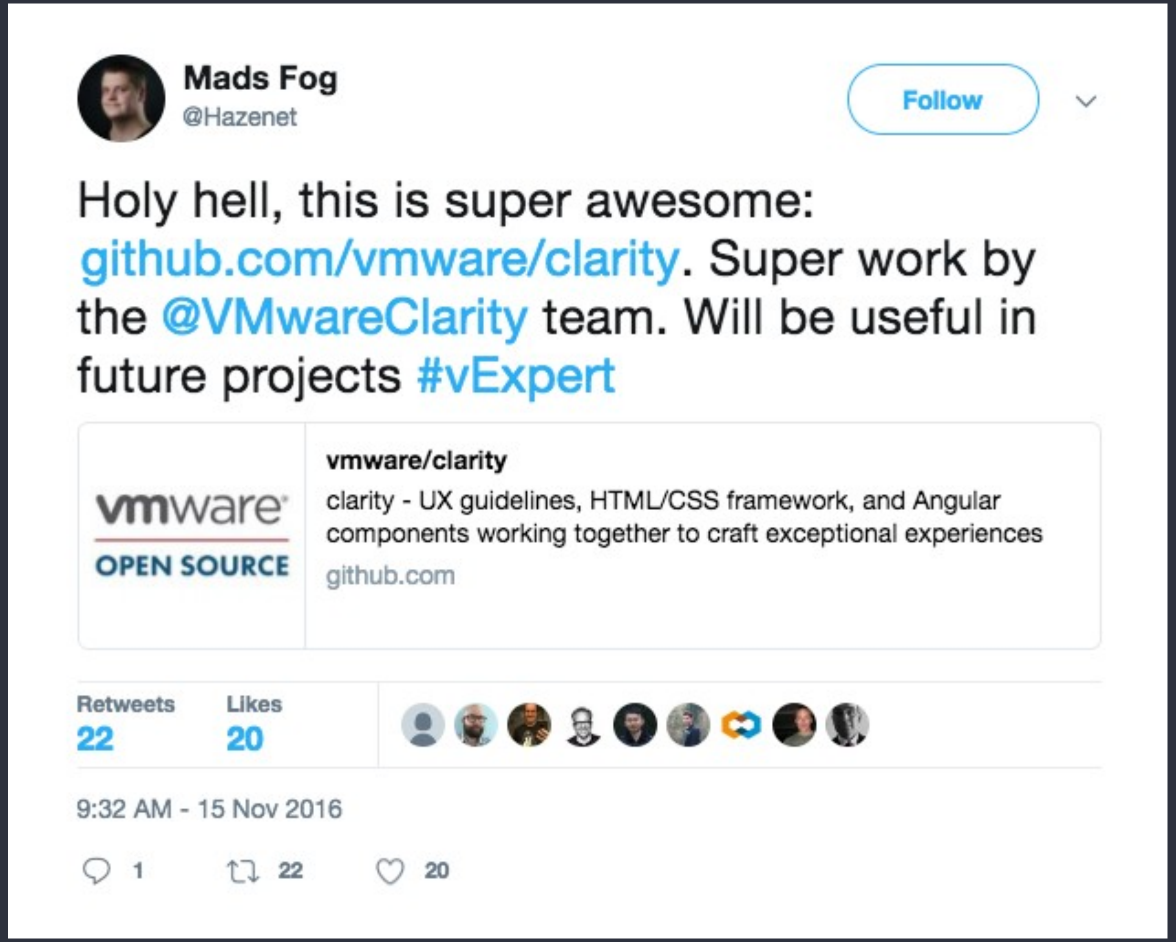


Components

In November of 2016,
Clarity decided to **go public**.

Clarity's Open Source Journey

Since November 15th 2016...


Mads Fog @Hazenet Follow

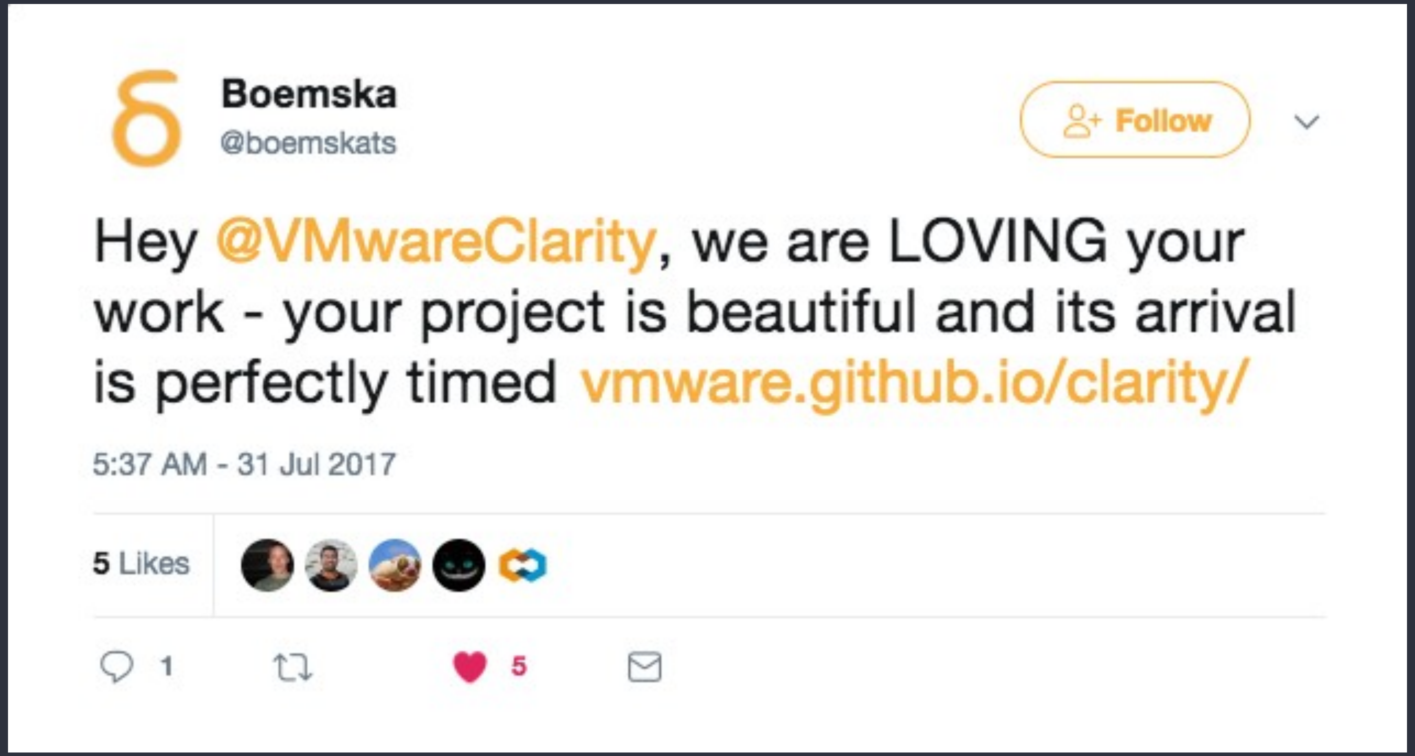
Holy hell, this is super awesome: github.com/vmware/clarity. Super work by the @VMwareClarity team. Will be useful in future projects #vExpert

vmware OPEN SOURCE | **vmware/clarity** clarity - UX guidelines, HTML/CSS framework, and Angular components working together to craft exceptional experiences github.com

Retweets 22 Likes 20

9:32 AM - 15 Nov 2016

1 22 20

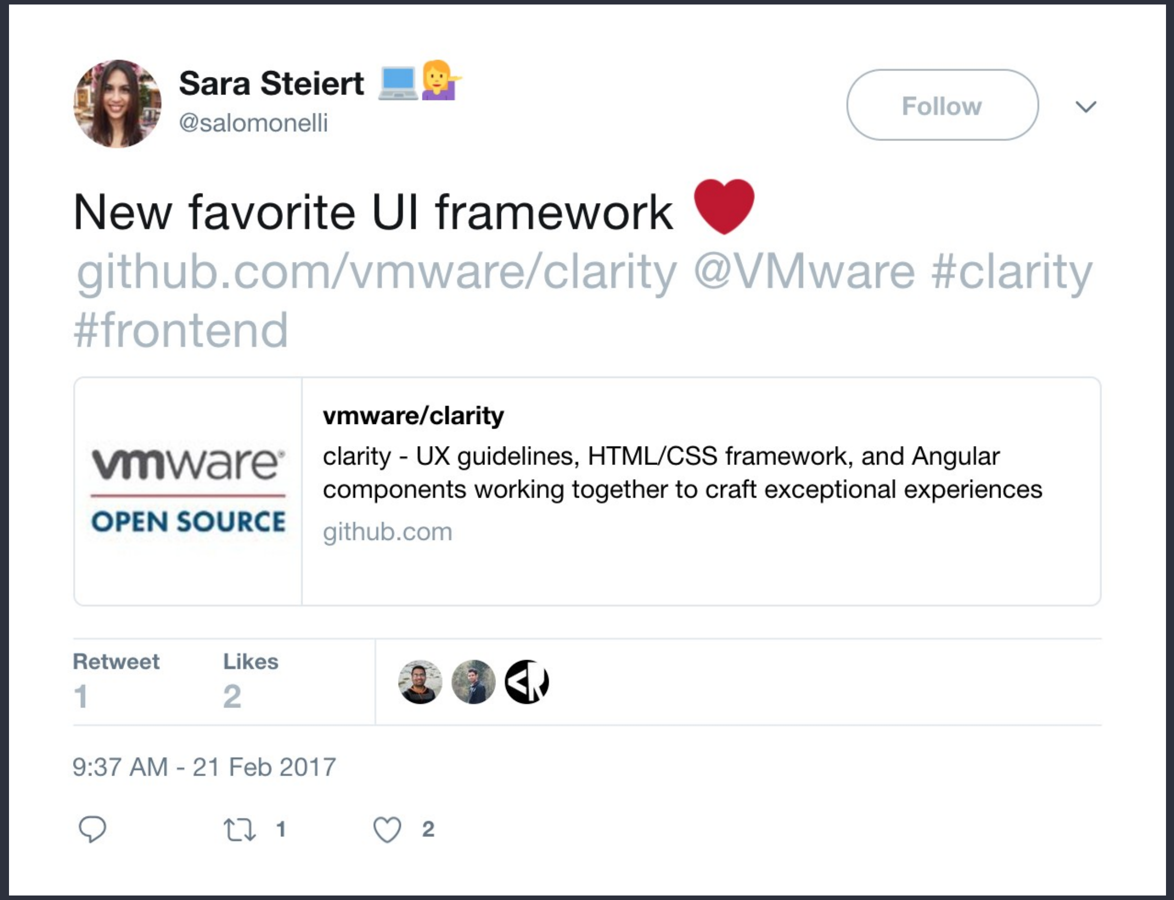

Boemska @boemskats Follow

Hey @VMwareClarity, we are LOVING your work - your project is beautiful and its arrival is perfectly timed vmware.github.io/clarity/

5:37 AM - 31 Jul 2017

5 Likes

1 5


Sara Steiert @salomonelli Follow

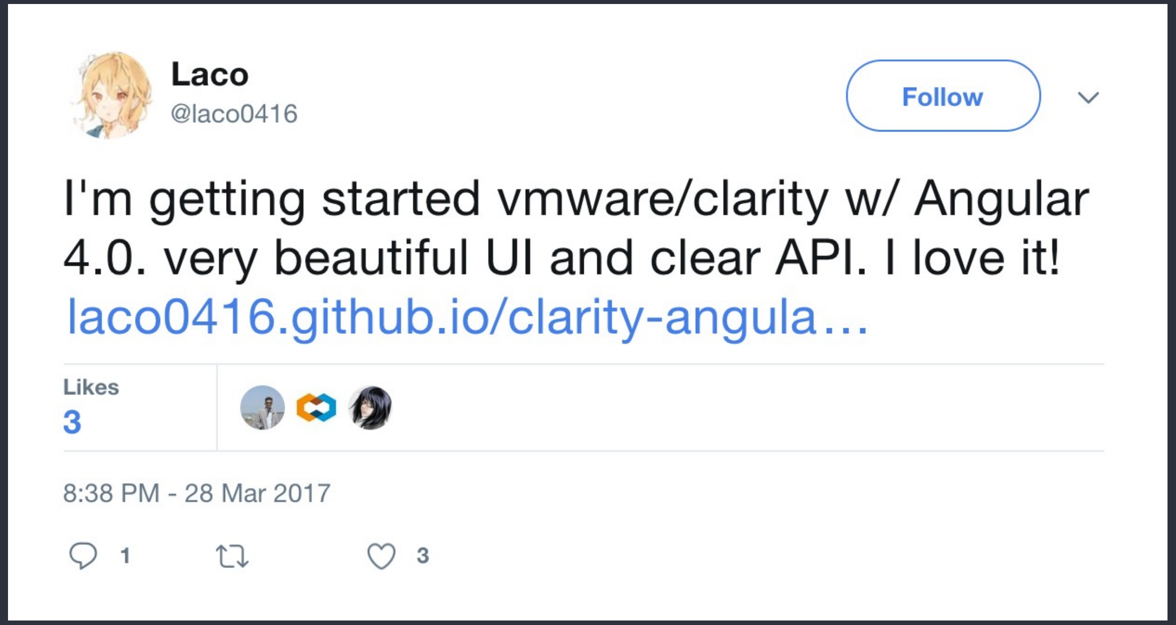
New favorite UI framework ❤ github.com/vmware/clarity @VMware #clarity #frontend

vmware OPEN SOURCE | **vmware/clarity** clarity - UX guidelines, HTML/CSS framework, and Angular components working together to craft exceptional experiences github.com

Retweet 1 Likes 2

9:37 AM - 21 Feb 2017

1 2

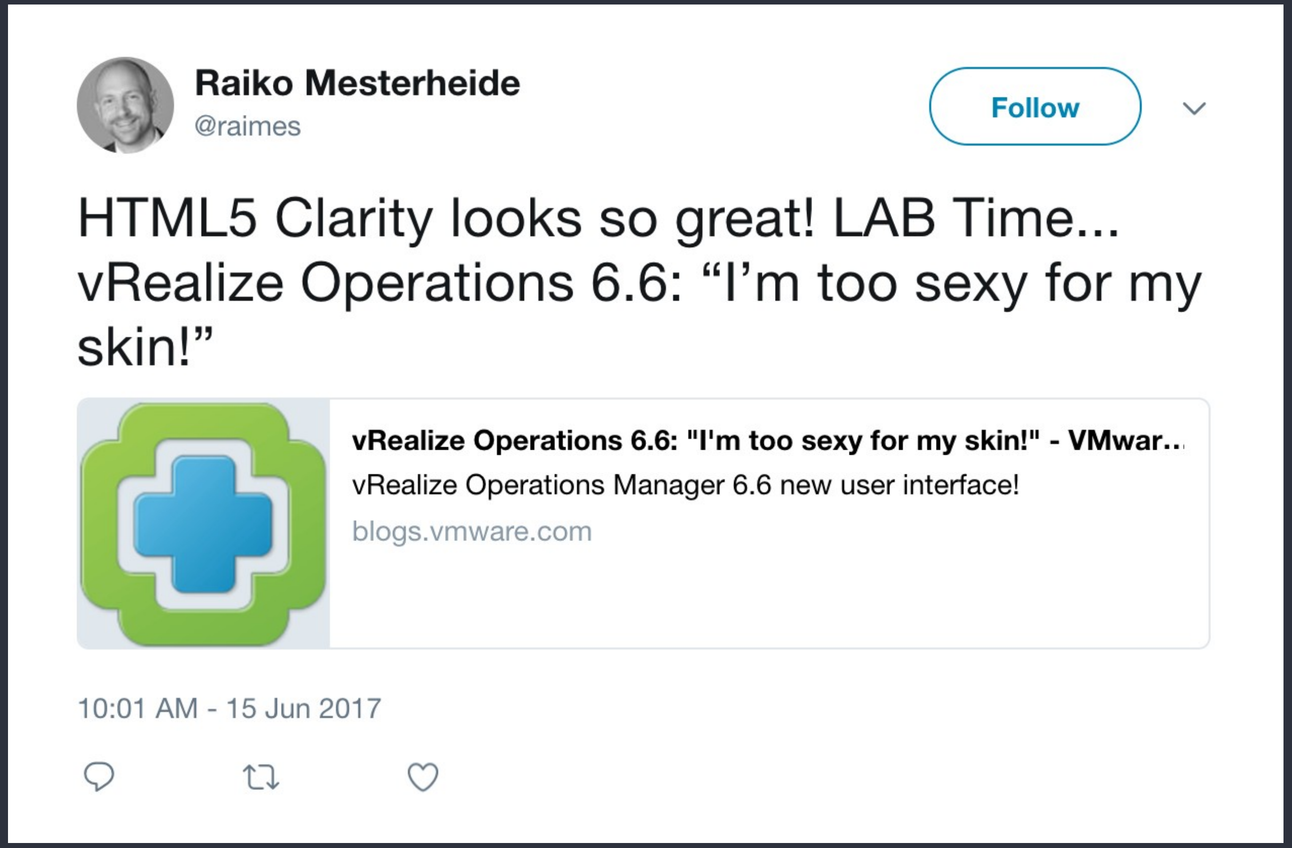

Laco @laco0416 Follow

I'm getting started vmware/clarity w/ Angular 4.0. very beautiful UI and clear API. I love it! laco0416.github.io/clarity-angula…

Likes 3

8:38 PM - 28 Mar 2017

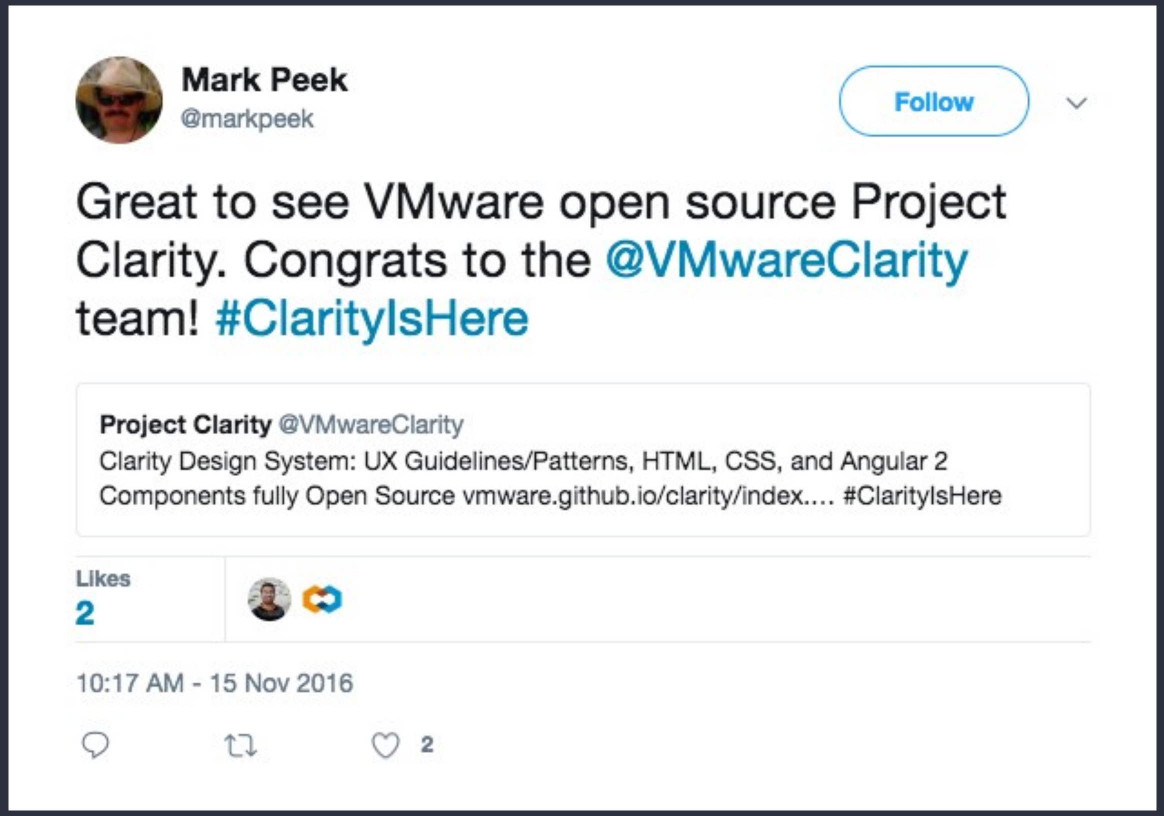
1 3


Raiko Mesterheide @raimes Follow

HTML5 Clarity looks so great! LAB Time... vRealize Operations 6.6: "I'm too sexy for my skin!"

vRealize Operations 6.6: "I'm too sexy for my skin!" - VMwar.. vRealize Operations Manager 6.6 new user interface! blogs.vmware.com

10:01 AM - 15 Jun 2017


Mark Peek @markpeek Follow

Great to see VMware open source Project Clarity. Congrats to the @VMwareClarity team! #ClarityIsHere

Project Clarity @VMwareClarity
Clarity Design System: UX Guidelines/Patterns, HTML, CSS, and Angular 2 Components fully Open Source vmware.github.io/clarity/index.... #ClarityIsHere

Likes 2

10:17 AM - 15 Nov 2016

2

Clarity’s Open Source Journey

Since November 15th 2016...

1900+
Stars on GitHub

500+
Pull Requests

40+
Weekly Release

1,000,000+
Page Views

174+
Countries

120+
Languages

Engage with Us!

http://clarity.design

@VMwareClarity

Hands-on: Building an application with Clarity

The picture can't be displayed.

Hands-on: Building an application with Clarity

In this training session you'll learn how to..

- Create an Angular +Clarity application
- Work with navigation and routing
- Create components
- Write and inject services
- Retrieve data using HTTP

Getting started with Clarity seed

- **Prerequisite**: Install Node.js, IDE such as Visual Studio Code (Optional)

```
# install angular-cli globally
npm install -g @angular/cli

# clone the latest version of Clarity seed
git clone https://github.com/vmware/clarity-seed.git
cd clarity-seed

# install the project's dependencies
npm install

# start the application - it will watch for file changes for live-reload
ng serve
```

Angular-cli

- Command line interface tool for speedy Angular code generation

```
# install angular-cli globally
npm install -g @angular/cli

# generating a new component
ng g component my-new-component

# generating a new service
ng g service my-new-service

# to learn more about Angular-CLI commands and theirusages
ng help
```

Creating a component

```
# generating a new component
   ng g component test

# add a new route in app.routing.ts
   export const ROUTES: Routes = [
      ...
      {path: 'test', component: TestComponent}
   ];

# add a new navigation link in the template app.component.html
   <div class="header-nav" [clr-nav-level]="1">
     ...
     <a class="nav-link" href="#" [routerLink]="['/test']" routerLinkActive="active">
        <span class="nav-text">Test</span></a>
   </div>
```

Setting up the page

- A **grid** provides a structure of rows and columns for aligning content
- Clarity uses a 12-column, responsive grid
- Replace the content of home.component.html with the following:

```
<div class="row">
    <div class="col-sm-12 col-md-8">
        <span>span 1</span>
    </div>
    <div class="col-sm-12 col-md-4">
        <span>span 2</span>
    </div>
</div>
```

Adding a datagrid

- Add the following to the first column div in home.component.html:

```
<h3>Pick your Star Wars captain</h3>
<clr-datagrid>
  <clr-dg-column>User ID</clr-dg-column>
  <clr-dg-column>Name</clr-dg-column>

  <clr-dg-row *ngFor="let user of users">
    <clr-dg-cell>{{user.id}}</clr-dg-cell>
    <clr-dg-cell>{{user.name}}</clr-dg-cell>
  </clr-dg-row>

  <clr-dg-footer>{{users.length}} users</clr-dg-footer>
</clr-datagrid>
```

- Define users in home.component.ts:

```
users =[ {id: 1,name: “Alice" },{id: 2, name: “Bob" }];
```

Fetching data through a service

- We'll be using The Star Wars API to fetch data: https://swapi.co/

```
# generating a new service
   ng g service people
```

```
# use Angular's Http module to call this API in people.service.ts
   import {Injectable }from '@angular/core';
   import {Http }from "@angular/http";
   import "rxjs/add/operator/map";

   @Injectable()
   export class PeopleService {
     constructor(private http: Http) {}

     get(page: number =1) {
         let restUrl = `https://swapi.co/api/people/?page=${page}`;
         return this.http.get(restUrl).map(data =>data.json());
     }
```

Consuming our service in the component

- In home.component.ts:

```
import {PeopleService} from "../people.service";

@Component({
        ...
    providers: [PeopleService]
})
export class HomeComponent {
    currentPage = 1;
    people = [];

    constructor(private peopleService: PeopleService) {
        this.peopleService.get(this.currentPage).subscribe( data =>{
                console.log(data);
                this.people =data.results;
        });
    }
}
```

Populating datagrid

- Modify home.component.html to display data from the service call:

```
<clr-datagrid>
  <clr-dg-column>Name</clr-dg-column>
  <clr-dg-column>Birth Year</clr-dg-column>
  <clr-dg-column>Gender</clr-dg-column>

  <clr-dg-row *ngFor="let person of people">
    <clr-dg-cell>{{person.name}}</clr-dg-cell>
    <clr-dg-cell>{{person.birth_year}}</clr-dg-cell>
    <clr-dg-cell>{{person.gender}}</clr-dg-cell>
  </clr-dg-row>

  <clr-dg-footer>{{people.length}} people</clr-dg-footer>
</clr-datagrid>
```

Adding pagination -footer

- Modify the footer in home.component.html:

```
<clr-dg-footer>
  {{pg.firstItem +1}}- {{pg.lastItem +1}}of {{pg.total}} people
  <clr-dg-pagination #pg [(clrDgPage)]="currentPage"
      [clrDgPageSize]="10" [clrDgTotalItems]="total">
  </clr-dg-pagination>
</clr-dg-footer>
```

- Define total in home.component.ts:

```
…
total =0;

constructor(private peopleService: PeopleService) {
   this.peopleService.get(this.currentPage).subscribe( data =>{
      …
      this.total =data.count;
   });
}
```

Fetching data for another page - template

- Add datagrid’s properties in home.component.html:

```
<clr-datagrid
        (clrDgRefresh)="refresh($event)" [clrDgLoading]="loading">
  ...
  <clr-dg-row *ngFor="let person of people">
    <clr-dg-cell>{{person.name}}</clr-dg-cell>
    ...
  </clr-dg-row>
  ...
</clr-datagrid>
```

Data direction	Syntax
data source (class) to view target (html)	{{expression}} or [target]=“expression”
view target (html) to data source (class)	(target)=“statement”
Two-way	[(target)]=“expression”

Fetching data for another page - component

- Define variables and refresh function in home.component.ts:

```
export class HomeComponent {
    ...
    selected;
    loading = true;
    constructor(private peopleService: PeopleService) {}
    refresh(state: State) {
      this.loading = true;
      this.peopleService.get(this.currentPage).subscribe( data => {
        ...
        this.loading = false;
      });
    }}
```

Enabling selection

- Add datagrid’s properties for selection in home.component.html:

```
<clr-datagrid [(clrDgSingleSelected)]="selected"
        (clrDgRefresh)=“refresh($event)" [clrDgLoading]="loading">
   ...
  <clr-dg-row *ngFor="let person of people" [clrDgItem]=“person">
     ...
  </clr-dg-row>
   ...
</clr-datagrid>
```

Displaying selection (1)

- Add a card to the other column in home.component.html:

```
<ng-container *ngIf="selected">
    <h3>You've selected</h3>
    <div class="card">
        <div class="card-header">
            {{selected.name}}
        </div>
        <div class="card-block">
            …
        </div>
        <div class="card-footer">
            …
        </div>
    </div>
</ng-container>
```

Displaying selection (2)

- Fill out card-block with data in home.component.html:

```
<div class="card">
  ...
   <div class="card-block">
      <ul>
         <li>Height: {{selected.height}}</li>
         <li>Mass: {{selected.mass}}</li>
         <li>Hair Color: {{selected.hair_color}}</li>
         <li>Eye Color: {{selected.eye_color}}</li>
      </ul>
   </div>
   ...
</div>
```

Displaying selection (3)

- Fill out card-block with data in home.component.html:

```
<div class="card">
    ...
    <div class="card-footer">
        <div class="row">
            <div class="col-sm-6 col-md-6">
                <clr-icon shape="car"></clr-icon>
                {{selected.vehicles.length}} vehicles
            </div>
            <div class="col-sm-6 col-md-6">
                <clr-icon shape="plane"></clr-icon>
                {{selected.starships.length}} starships
            </div>
        </div>
    </div>
</div>
```

Extra - adding a tooltip

- A tooltip can provide more information to the user:

```
<clr-dg-
   column>
   Birth Year
   <clr-tooltip>
      <clr-icon clrTooltipTrigger shape="info-circle"></clr-icon>
      <clr-tooltip-content clrPosition="right" clrSize="md" *clrIfOpen>
         <span>BBY - Before the Battle of Yavin</span>
      </clr-tooltip-content>
   </clr-tooltip>
</clr-dg-column>
```

Extra - adding an alert inside the card

```
<div class="card-block">
    <clr-alert *ngIf="selected.starships.length == 0"
          [clrAlertSizeSmall]="true" [clrAlertType]="'warning'">
        <div class="alert-item">
            <span class="alert-text">
                {{selected.name}} has no starships. May be difficult to get around.
            </span>
        </div>
    </clr-alert>
    ...
</div>
```

The Final Product

The picture can't be displayed.

Resources

The picture can't be displayed.

git.io/vmworld-clarity

bit.ly/vmworld-clarity

Thank You!

Q&A