Kentico

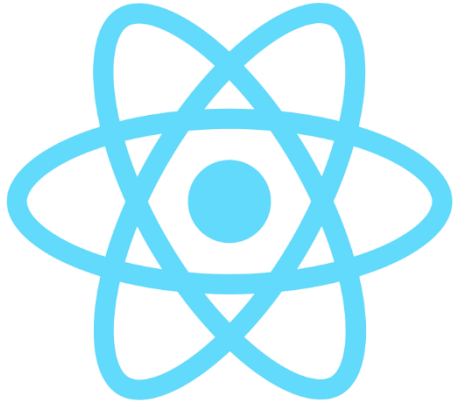
Lecture 10

Hana Navrátilová

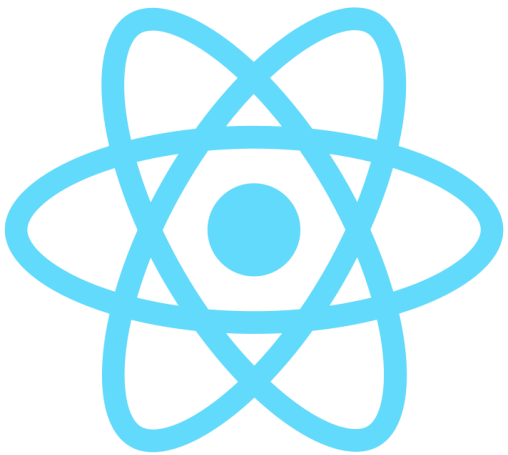

Kentico



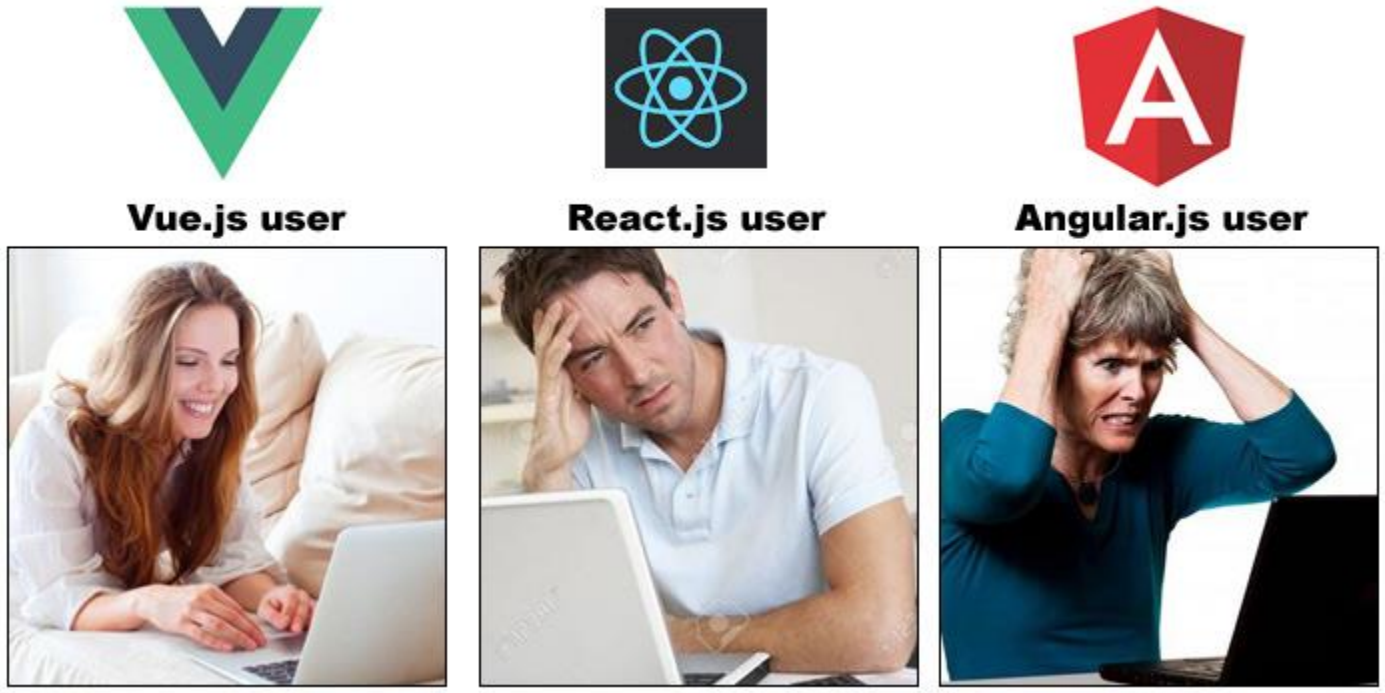


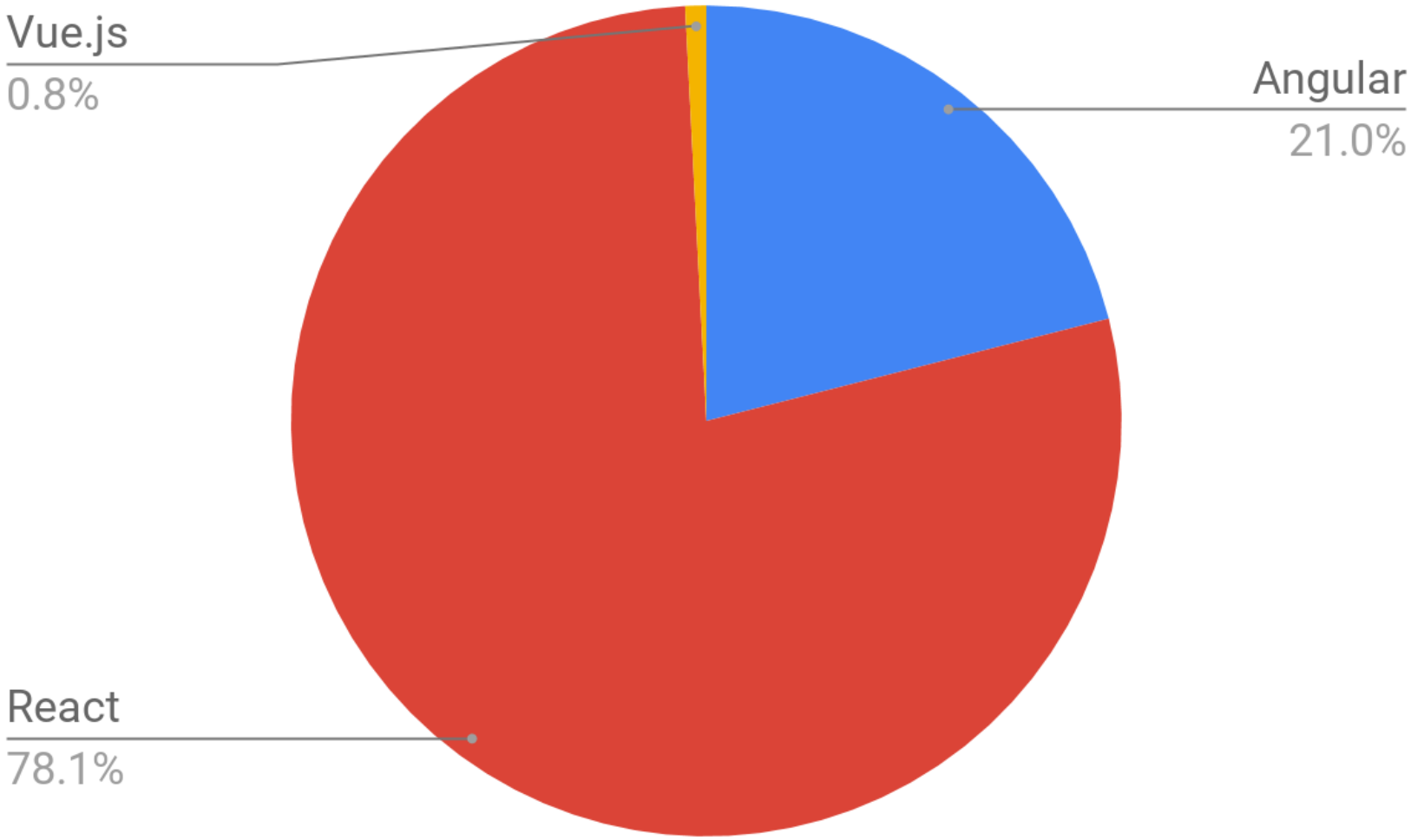
Kentico



Kentico



https://i.warosu.org/data/g/img/0625/33/1506048572584.jpg

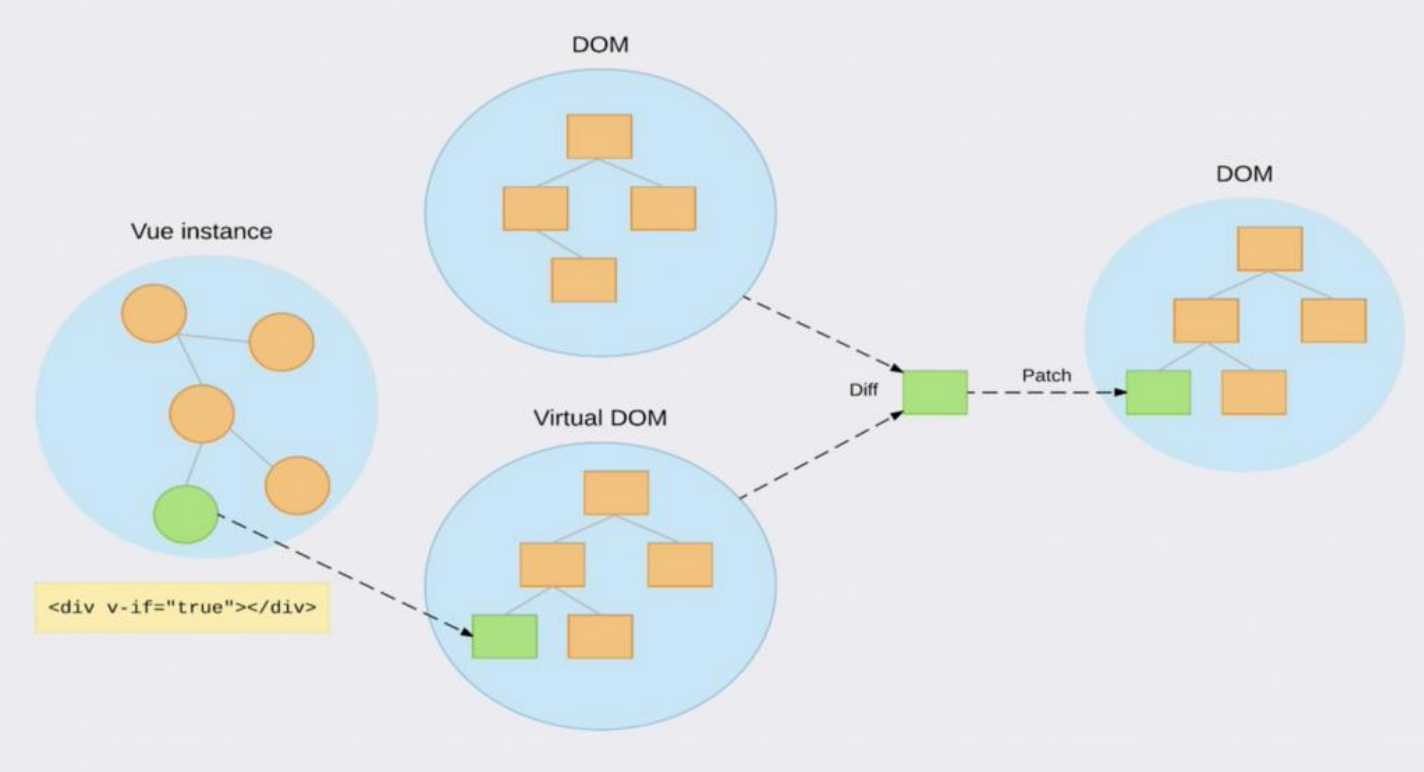


medium.com (Jul 30, 2018): We analyzed the number of open positions worldwide that require a specific knowledge of a certain framework. As a source, we took Indeed.com and got the following distribution according to more than 60,000 job offers.

Kentico

Virtual DOM

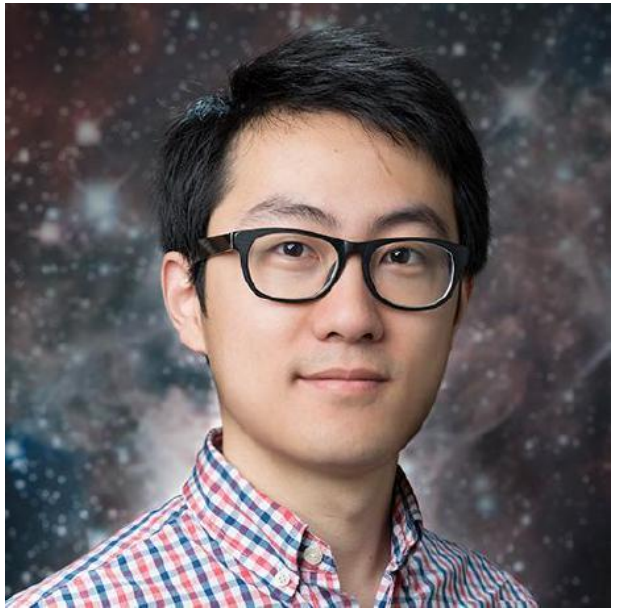
- **What is the Virtual DOM?**
  - Is an abstraction of the HTML DOM (*Document Object Model*)
  - **How it helps?**
    - Manipulating the virtual DOM is much faster than DOM (nothing gets drawn onscreen)
    - Only objects wich have changed are updated on the real DOM



Kentico

VUE.JS

- 2013
- Evan You
- JavaScript Framework
- JS or TS
- Render to VirtualDOM
- 2-way binding
- Vuex



Kentico VUE.JS

Component structure

component.vue

```
<template>
… HTML …
</template>

<script>
… JS(X) …
</script>

<style>
… CSS …
</style >
```

component.vue

```
<template>
… HTML …
</template>

<script lang="ts">
… TS(X) …
</script>

<style>
… CSS …
</style >
```

```
<script src="myscript.j(t)s"/>
```

Kentico VUE.JS

Components

1. `export default { name: '', … }`

2. ```
   @Component({ … })
   export default class name extends Vue { … }
   ```

3. ```
   export default Component({ … })(
     class name extends Vue { … }
   )
   ```

4. ```
   Vue.component('name', {
     template: ''
   })
   ```
   ```

Kentico VUE.JS

Data object = Reacts state

1.

```
<script>
export default {
  …,
  data: function () {
    return {
      text: 'hi'
    }
  },
  …
}
</script>
```

```
<script>
let inputValue = 'text'

export default {
  …,
  data: function () {
    return {
      inputValue
    }
  },
  …
}

</script>
```

2., 3.

```
<script>
let inputValue = 'text'
</script>
```

Kentico VUE.JS

Rendering

One way binding (from data source to view target):

```
<span> {{text}} </span>
```

One way binding (from view target to data source):

```
<button v-on:click="add"> Add </button>
```

Two way binding:

```
<input type="text" v-model="inputValue"/>
```

Filters:

```
<span> {{text | formatText}} </span>
```

Kentico VUE.JS

Rendering - filters

```
<span> {{text | formatText}} </span>
```

1.
```
<script>
export default {
  …,
  filters: {
    formatText: function (value) {
      return …
    }
  },
  …
}
</script>
```

Kentico VUE.JS

Rendering - filters

```
<span> {{text | formatText}} </span>
```

2.
```
function formatText (value) {
  return …
}

@Component({
  filters: {formatText}
})
export default class name extends Vue {}
```

Kentico VUE.JS

Rendering - filters

```
<span> {{text | formatText}} </span>
```

3.
```
function formatText (value) {
  return …
}

export default Component({
  filters: {formatText}
})( class name extends Vue {})
```

Kentico VUE.JS

Conditionals

```
<template>
  <span v-if="seen">Now you see me</span>
  <h1 v-else>You don't see the first text.</h1>
</template>
```

```
export default {
  data: {
    seen: true
  }
}
```

Kentico VUE.JS

Loops

```
<template>
  <li v-for="(item, index) in items">
    <span{{index}}. {{item.text}}</span>
  </li>
</template>

<script>
export default {
  data: function () {
    return {
      items: [{
        text: 'item1'
      },{
        text: 'item2'
      }]
    }}}
</script>
```

Kentico VUE.JS

Component registration

```
<template>
  <item />
</template>
```

1.
```
import item from './item

export default {
  components: [
    item
  ], ... }
```

2.
```
import item from './item

@Component({
  components: { item }
})
```

3.
```
import item from './item

export default Component ({
  components: { item }
})
```

Kentico VUE.JS

Props

```
<template>
  <item :index="index + 1" :item="item"/>
</template>
```

1.
```
export default {
 name: item',
 props: [
    'index',
    'item'
  ],
  …
}
```

Kentico VUE.JS

Props

```
<template>
  <item :index="index + 1" :item="item"/>
</template>
```

2.
```
@Component({props: [
  'item',
  'index'
]})
export default class item extends Vue { … }
```

Kentico VUE.JS

Props

```
<template>
  <item :index="index + 1" :item="item"/>
</template>
```

3.
```
export default Component({props: [
  'item',
  'index'
]})(
  class activeItem extends Vue { … }
```

Kentico VUE.JS

Callbacks

listItem.vue

```
<template>
  <button v-on:click="saveItem">
    Save item
  </button>
</template>

<script>
export default {
  name: 'listItem', … ,
  methods: {
    saveItem : function () {
      this.$emit('onSaveItem',
        this.item.id, text)
 }}}
</script>
```

list.vue

```
<template>
  <listItem @onSaveItem="editItem" />
</template>

<script>
import Vue from 'vue'
import { Component } from 'vue-property-
decorator'
import listItem from './listItem'

@Component({
  components: { listItem }
})
export default class list extends Vue {
  editItem (id, text){
    …
  }
</script>
```

Kentico VUE.JS

Vue task

1. PhoneBook component
2. Person component
3. Format number pipe
4. Delete person emit

Name	Surname	Phone	
Karel	Novak	786 581 235	Delete
Jan	Stastny	748 528 965	Delete
Klara	Zemanova	738 596 125	Delete
Petr	Kral	785 965 842	Delete
Kamil	Zednik	749 685 123	Delete
Tereza	Novakova	796 584 158	Delete
Karolina	Fialova	793 254 186	Delete
Petr	Novotny	741 245 863	Delete

```
[phoneNumber.slice(0, 3), ' ', phoneNumber.slice(3)].join('')
```

Kentico

ANGULAR

- 2010
- Google
- New version of AngularJS
- (JavaScript) Framework
- TS
- Render to DOM
- 2-way binding
- NgRX store

https://me.me/i/youre-a-fairy-here-to-grant-and-im-you-a-18387195

Google

Kentico ANGULAR

Component structure

text.component.ts

```ts
import { Component } from '@angular/core';

@Component({
  selector: ' app-text ',
  templateUrl: './text.component.html',
  styleUrls: ['./text.component.css']
})

export class TextComponent{
}
```

text.component.html

```html
<div>
  <span>My component!</span>
</div>
```

text.component.css

```css
span {
}
```

Kentico

ANGULAR

Data = Reacts state

```
import …

@Component({ … })

export class Component{
  inputValue:  = 'text';

  items: ListItem[] = getDefaultList();
}
```

Rendering

One way binding (from data source to view target):

```
<span> {{text}} </span>
```

One way binding (from view target to data source):

```
<button (click)="delete()">Delete </button>
```

Two way binding:

```
<input type="text" [(ngModel)]="inputValue"/>
```

Pipe operators:

```
<h2>{{name | uppercase}}</h2>
```

Kentico

ANGULAR

Rendering - pipes

```
<h2>{{text | myPipe}}</h2>
```

```
import {Pipe, PipeTransform} from '@angular/core';

@Pipe({
  name:'myPipe'
})
export class MyPipe implements PipeTransform {
  transform(value: any, ...args: any[]): any {
    return …;
  }
}
```

```
@NgModule({
  declarations: [
    MyPipe
  ]
})
export class Module { }
```

Conditionals

```
<div *ngIf=„condition; then thenBlock else elseBlock "> </div>
<ng-template #thenBlock> Yes </ng-template>
<ng-template #elseBlock> No </ng-template>
```

```
<div *ngIf=„condition; else elseBlock ">Yes</div>
<ng-template #elseBlock> No </ng-template>
```

Loops

```
<ul>
  <li *ngFor="let item of items; let i = index">
     <span> {{item.text}} </span>
  </li>
</ul>
```

Kentico ANGULAR

Component registration

app.module.ts

```
import …;

@NgModule({
  declarations: [
    AppComponent, …
  ],
  imports: [
    …,
    ListModule
  ],
  bootstrap: [AppComponent]
})
export class AppModule { };
```

list.module.ts

```
import …;
@NgModule({
  imports: [
    CommonModule,
    FormsModule
  ],
  exports:[
    ListComponent
  ],
  declarations: [
    ListComponent,
    AddItemComponent,
    MyPipe,
  ]
})
export class ListModule { };
```

main.ts

```
platformBrowserDynamic()
  .bootstrapModule(AppModule);
```

Kentico

ANGULAR

Props

```
<app-inactive-item [index]="i" [item]="item" />
```

```
import …;

@Component({…})
export class InactiveItemComponent {
  @Input() index: number;
  @Input() item: ListItem;

  …
}
```

Kentico ANGULAR

Callbacks

list.component.html

```
<app-active-item (onSaveItem)="editItem(item.id, $event)"/>
```

activeItem.component.html

```
<button(click)="saveItem()"> Save </button>
```

activeItem.component.ts

```
import …;

@Component({…})
export class ActiveItemComponent {
  @Output() onSaveItem: EventEmitter<string>
      = new EventEmitter<string>();

  saveItem = () =>
     this.onSaveItem.emit(this.inputValue);
}
```

list.component.ts

```
import …;

@Component({…})
export class ListComponent {
  editItem(id: Uuid, text: string) {
     …
  }
}
```

Kentico ANGULAR

Services

```
@Injectable({
  providedIn: 'root',
})
export class Service {
    method(): string{
      return …;
    }
}

export class Component {
  text: string;

  constructor(private  service: Service) {
    this.text = this.service.method();
  }
}
```

Kentico

ANGULAR

Angular task

1. CitiesList component
2. City component
3. Degrees pipe
4. City module

`ngModelChange`

City	Temperature
Amsterdam	6 °C
Ankara	11 °C
Bangkok	29 °C
Barcelona	13 °C
Bejing	5 °C
Cape Town	29 °C
Chicago	0 °C
Dublin	10 °C
Havana	26 °C
Helsinki	3 °C
Kuala Lumpur	28 °C
London	11 °C
Miami	26 °C
Montréal	-6 °C
New York	4 °C
Paris	7 °C
Prague	3 °C
Singapore	28 °C
Sydney	18 °C
Vienna	3 °C
Zürich	0 °C

B

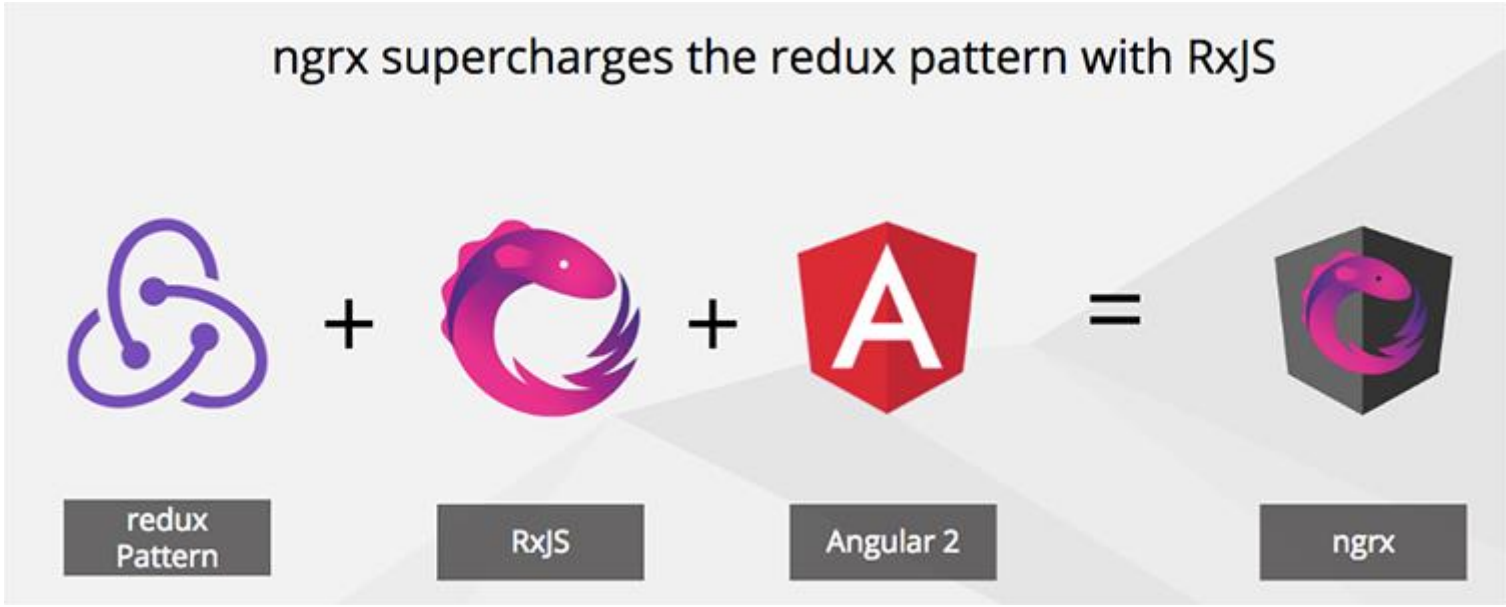
City	Temperature
Bangkok	29 °C
Barcelona	13 °C
Bejing	5 °C

Kentico




Pete Hunt, one of the early contributors to React, says:

„You'll know when you need Flux. If you aren't sure if you need it, you don't need it.“



https://medium.com/stratajet-tech/a-beginners-guide-to-ngrx-store-bc2184d6d7f0

Kentico

	Flux	Redux	Vuex	NgRX
Multiple stores	YES	NO*	NO	NO*
State is mutable	YES	NO	YES	NO
Dispatching process	Actions are written in the store itself	Sparate actions and reducers and store	Actions are written in the store itself	Same as Redux

*There can be multiple stores, but they don't know about each other.

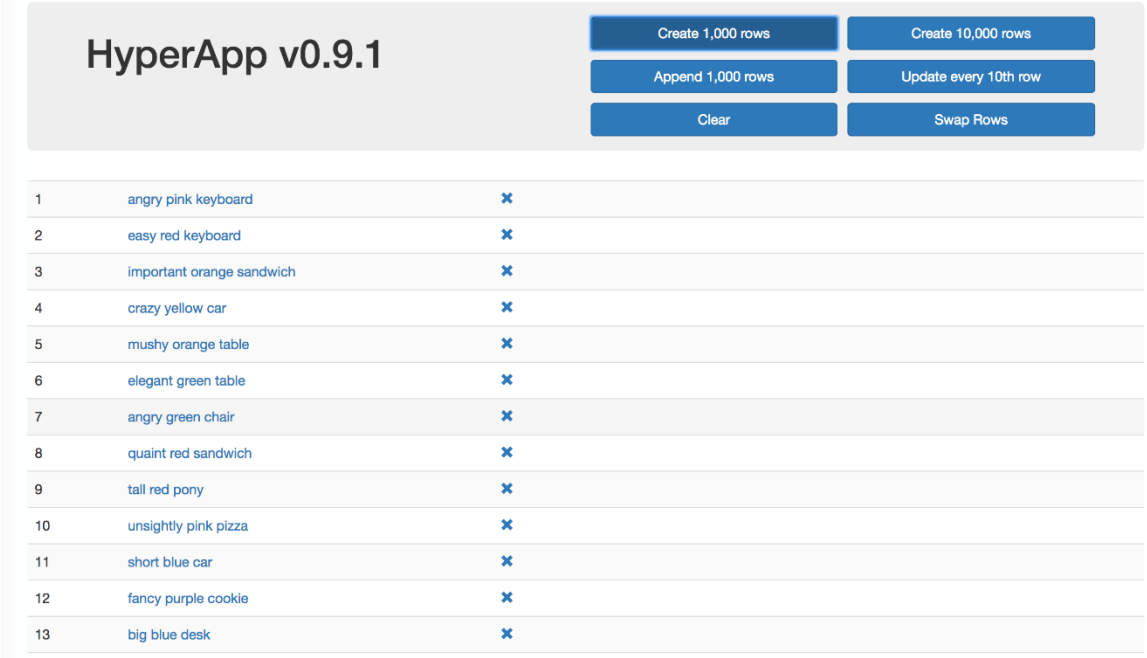
https://edgecoders.com/the-difference-between-flux-and-redux-71d31b118c1
https://vuex.vuejs.org/

Kentico

Summary

Kentico

Rendering time



Name	react-v16.1.0-keyed	angular-v5.0.0-keyed	vue-v2.5.3-keyed
create rows Duration for creating 1000 rows after the page loaded.	187.6 ± 4.3 (1.1)	185.7 ± 7.8 (1.1)	169.2 ± 3.6 (1.0)
replace all rows Duration for updating all 1000 rows of the table (with 5 warmup iterations).	165.2 ± 7.0 (1.0)	179.3 ± 6.5 (1.1)	161.8 ± 3.9 (1.0)
partial update Time to update the text of every 10th row (with 5 warmup iterations) for a table with 10k rows.	93.6 ± 5.6 (1.3)	73.5 ± 4.9 (1.0)	168.1 ± 7.4 (2.3)
select row Duration to highlight a row in response to a click on the row. (with 5 warmup iterations).	12.4 ± 4.1 (1.0)	7.6 ± 4.0 (1.0)	9.8 ± 2.5 (1.0)
swap rows Time to swap 2 rows on a 1K table. (with 5 warmup iterations).	19.6 ± 4.7 (1.0)	20.1 ± 4.2 (1.0)	21.8 ± 4.5 (1.1)
remove row Duration to remove a row. (with 5 warmup iterations).	51.5 ± 2.0 (1.1)	46.1 ± 2.6 (1.0)	52.5 ± 1.8 (1.1)
create many rows Duration to create 10,000 rows	2033.7 ± 32.0 (1.3)	1682.0 ± 53.1 (1.1)	1521.4 ± 55.7 (1.0)
append rows to large table Duration for adding 1000 rows on a table of 10,000 rows.	271.8 ± 9.9 (1.1)	257.6 ± 11.1 (1.0)	338.4 ± 10.3 (1.3)
clear rows Duration to clear the table filled with 10.000 rows.	224.4 ± 6.0 (1.0)	360.3 ± 16.4 (1.6)	240.9 ± 11.4 (1.1)
startup time Time for loading, parsing and starting up	49.4 ± 0.7 (1.0)	88.8 ± 2.9 (1.8)	48.4 ± 2.4 (1.0)
slowdown geometric mean	1.09	1.15	1.15

November 20, 2017

https://medium.com/fundbox-engineering/react-vs-vue-vs-angular-163f1ae7be56
https://www.stefankrause.net/wp/?p=454
https://www.stefankrause.net/js-frameworks-benchmark7/table.html

Kentico

What should I choose?

- If you work at Google: **Angular**
- If you love TypeScript: **Angular (or React)**
- If you love object-orientated-programming (OOP): **Angular**
- If you work at Facebook: **React**
- If you like flexibility: **React**
- If you love big ecosystems: **React**
- If you like choosing among dozens of packages: **React**
- If you love JS & the "everything-is-Javascript-approach": **React**
- If you like really clean code: **Vue**
- If you want the easiest learning curve: **Vue**
- If you want the most lightweight framework: **Vue**
- If you want separation of concerns in one file: **Vue**
- If you are working alone or have a small team: **Vue (or React)**
- If your app tends to get really large: **Angular (or React)**
- If you want to build an app with react-native: **React**
- If you want to have a lot of developers in the pool: **Angular or React**
- If you work with designers and need clean HTML files: **Angular or Vue**
- If you like Vue but are afraid of the limited ecosystem: **React**

- If you can't decide, first learn **React**, then **Vue**, then **Angular**.

https://medium.com/unicorn-supplies/angular-vs-react-vs-vue-a-2017-comparison-c5c52d620176

Kentico


Hmm, I can't decide on my new year's resolution...
I could choose a single new JS Framework, and I work with it all year to really learn it properly...
Jquery for example?
Or maybe I should commit to trying a different JS framework every week for the rest of the year, so I stay up-to-date
Hum...
The fact that both ideas seem entirely reasonable demonstrates just how f*cked front-end dev is right now
CommitStrip.com

Kentico

- https://vuejs.org/
- https://angular.io/
- https://medium.com/unicorn-supplies/angular-vs-react-vs-vue-a-2017-comparison-c5c52d620176
- https://medium.com/@TechMagic/reactjs-vs-angular5-vs-vue-js-what-to-choose-in-2018-b91e028fa91d
- https://reactkungfu.com/2015/10/the-difference-between-virtual-dom-and-dom/
- https://medium.com/javascript-in-plain-english/i-created-the-exact-same-app-in-react-and-vue-here-are-the-differences-e9a1ae8077fd
- https://egghead.io/lessons/vue-js-simplify-vue-components-with-vue-class-component
- https://github.com/loganfsmyth/babel-plugin-transform-decorators-legacy
- https://blog.angular-university.io/angular-2-smart-components-vs-presentation-components-whats-the-difference-when-to-use-each-and-why/
- https://angular.io/api/core/NgModule

- https://github.com/voronianski/flux-comparison
- https://github.com/ngrx/store
- https://vuex.vuejs.org/
- https://angular.io/guide/rx-library
- https://edgecoders.com/the-difference-between-flux-and-redux-71d31b118c1
- https://medium.com/@Musclenun/redux-vs-vuex-9b682529c36
- https://blog.angular-university.io/angular-2-redux-ngrx-rxjs/