

micro-frontend

About me

Paul Williams



I have a vivid aspiration over technology and I use this passion to help team by coaching them in writing better code with better agility.



slack

<http://bit.ly/codejam-slack>

initial code

1



Fork me

<http://bit.ly/codejam-git>

2



gitpod.io/#{link to repo}

prerequisite

```
$ node -v          #version > 10
```

```
$ npm i -g @angular/cli
```

```
$ npm i
```

Did micro-service have all the answers?

Time to market front-end

Though the service(api) were sliced still the front-end team could only deliver as one component and were not independent (build, test & serve). All regressions had to pass.

Can we build and re-use ?

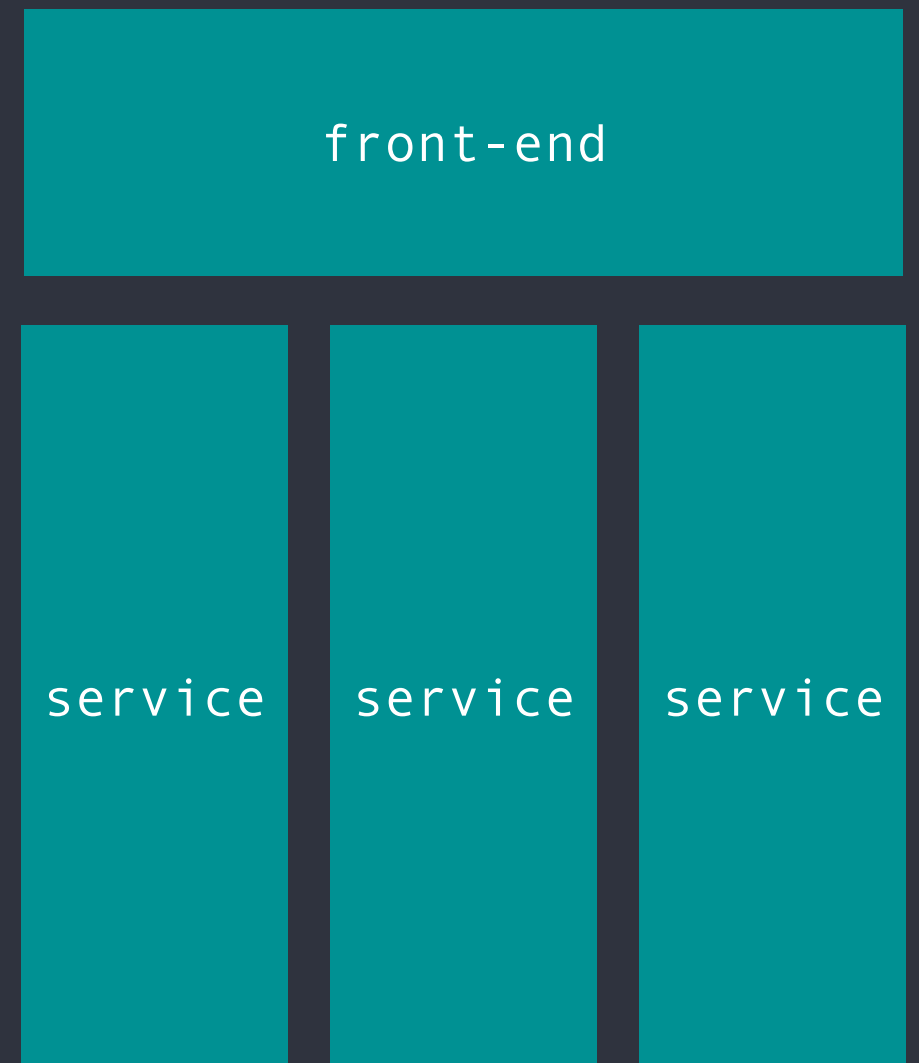
We were limited to use the built component only within the front-end app couldn't re-use it in other app(s) due to dependency of framework.

Progressive upgrades

Upgrading the bootstrap version from v3 to v4 was has to be done for the whole front-end app.

Technology agnostic

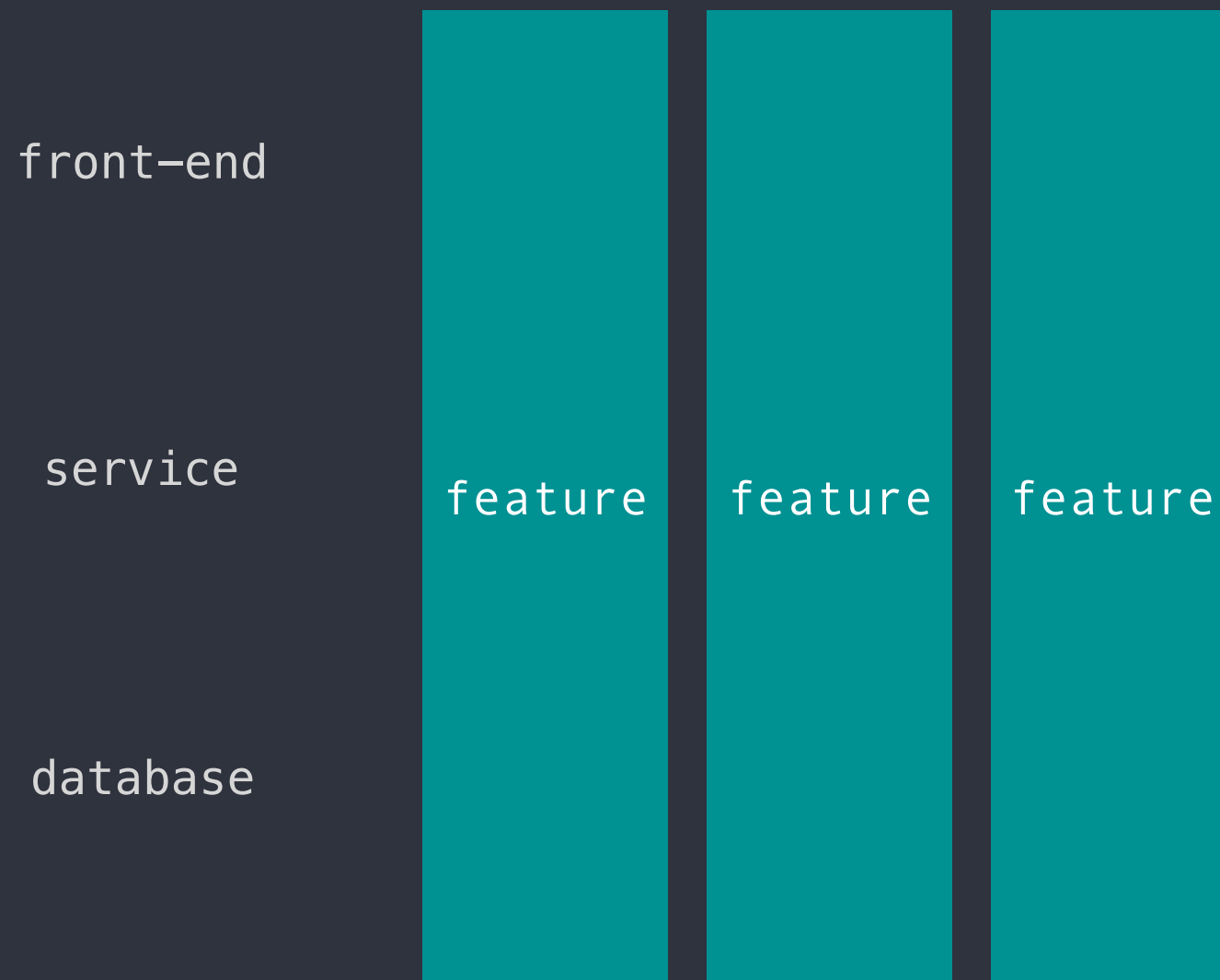
Could not use various frameworks like angular, vuejs, preact etc



What are we trying to do ?

Build, serve & deploy independently

Should be able to build, test, serve, lint and deploy independently without impacting other features. **The key is to slice it as features.**



What are we trying to do ?

Build as web-components

Make it more re-usable (like api)



WEB COMPONENTS

TEMPLATES

```
<template id="">  
</template>
```

SHADOW DOM

```
div  
  #document-fragment  
  span
```

HTML IMPORTS

```
<link rel="import"  
      href="part.html">
```

CUSTOM ELEMENTS

```
<my-elem>  
</my-elem>
```


What are we trying to do ?

Using toggles for progressive upgrades

Release features without impacting other features or regression through toggle mechanism



on ☒

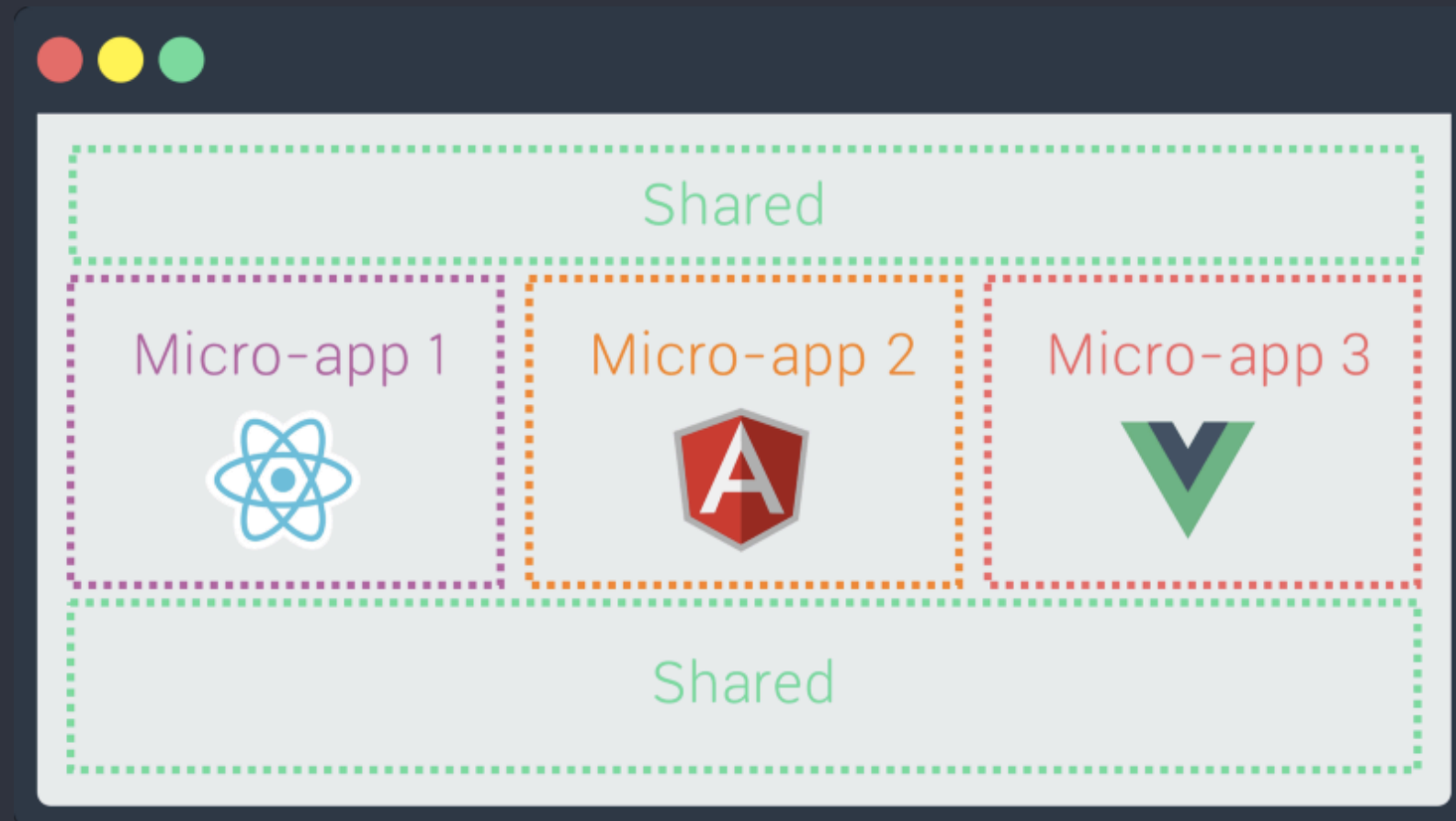


☐ off

What are we trying to do ?

Technology agnostics

Use any framework (angular, vuejs, preact etc)



talk is cheap.. show me the code

todo app



What are we trying to do ?



Build & serve independently

Should be able to build, test, serve, lint and deploy independently without impacting other features. **The key is to slice it as features.**



Build as web-components

Make it more re-usable (like api)



Using toggles for progressive upgrades

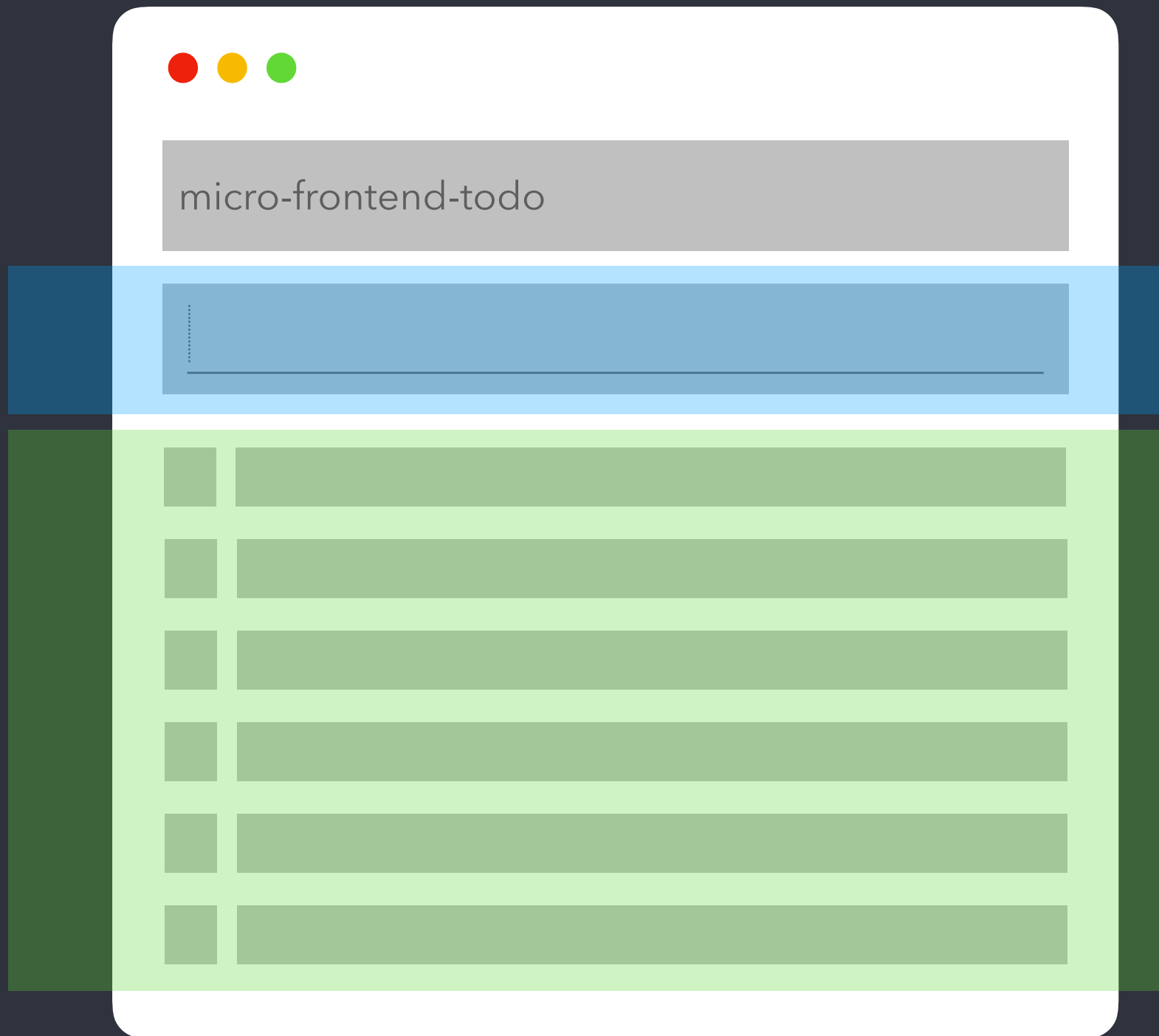
Release features without impacting other features or regression through toggle mechanism



Technology agnostics

Use any framework (angular, vuejs, preact etc)

feature teams



user should be able to create
todo items

user should be able to view all
the todo items created

gist

create-todo

<http://bit.ly/codejam-create-todo-gist>

view-todo

<http://bit.ly/codejam-view-todo-gist>

generate the micro-app

```
$ ng g application {create|view}-todo  
  --style=scss  
  --prefix={create|view}  
  --viewEncapsulation=ShadowDom
```

main.ts

```
platformBrowserDynamic().bootstrapModule(CreateTodoAppModule, {  
  defaultEncapsulation: ViewEncapsulation.ShadowDom  
})  
  .catch(err => console.error(err));
```

cleanup...

Delete all the **files** under
projects/{create|view}-todo/src/app



generate module, component, service

Navigate to **projects/{create|view}-todo/src/app**

```
$ ng g module {create|view}-todo --flat=true
```

```
$ ng g component {create|view}-todo  
--module={create|view}-todo  
--selector={create|view}-todo  
--entryComponent=true  
--flat=true
```

```
$ ng g service {create|view}-todo --flat=true
```

serve

```
$ ng serve
```

```
$ ng serve create-todo --port=4201
```

```
$ ng serve view-todo --port=4202
```

integrate with the main app

home.module.ts

```
@NgModule({  
  declarations: [HomeComponent],  
  imports: [  
    CommonModule,  
    CreateTodoModule, ViewTodoModule  
  ],  
})  
export class HomeModule {  
}
```

home.component.html

```
<div class="container-fluid">  
  <create-todo></create-todo>  
  <view-todo></view-todo>  
</div>
```

What are we trying to do ?



Build & serve independently

Should be able to build, test, serve, lint and deploy independently without impacting other features. **The key is to slice it as features.**



Build as web-components

Make it more re-usable (like api)



Using toggles for progressive upgrades

Release features without impacting other features or regression through toggle mechanism



Technology agnostics

Use any framework (angular, vuejs, preact etc)

What are we trying to do ?

Build as web-components

Make it more re-usable (like api)



WEB COMPONENTS

TEMPLATES

```
<template id="">  
</template>
```

SHADOW DOM

```
div  
  #document-fragment  
  span
```

HTML IMPORTS

```
<link rel="import"  
      href="part.html">
```

CUSTOM ELEMENTS

```
<my-elem>  
</my-elem>
```

make it a web component

```
$ ng add @angular/elements --project={create|view}-todo
```

create-todo.module.ts

```
@NgModule({
  declarations: [ CreateTodoComponent ],
  providers: [ CreateTodoService ],
  exports: [ CreateTodoComponent ],
  imports: [ CommonModule, BrowserModule ],
  entryComponents: [ CreateTodoComponent ],
  bootstrap: [CreateTodoComponent],
})
export class CreateTodoModule {
  constructor(private injector: Injector) {
  }

  ngDoBootstrap() {
    const createTodoElement = createCustomElement(CreateTodoComponent,
    {injector: this.injector});
    customElements.define('create-todo-element', createTodoElement);
  }
}
```

build the web component

```
$ npm i --save-dev concat fs-extra del
```

create-todo/build/bundle.js

```
const fs = require('fs-extra');
const concat = require('concat');
const del = require('del');

(async function build() {
  const files = [
    'dist/create-todo/runtime-es2015.js',
    'dist/create-todo/polyfills-es2015.js',
    'dist/create-todo/main-es2015.js'
  ];
  await del(['dist/elements/create-todo.js']);
  await fs.ensureDir('dist/elements');
  await concat(files, 'dist/elements/create-todo.js');
})();
```

bundle the web component

package.json

```
"build:create-todo:elements": "ng build create-todo --prod --output-hashing none && node projects/create-todo/build/bundle.js"
```

```
"build:view-todo:elements": "ng build view-todo --prod --output-hashing none && node projects/view-todo/build/bundle.js"
```


refer the web component

dist/elements/index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Web component</title>
</head>
<body>
<create-todo-element></create-todo-element>
<script type='application/javascript' src='create-todo.js'></script>
</body>
</html>
```



fixing the issue with web-pack

```
$ npm i --save @angular-builders/custom-webpack
```

angular.json

```
"build": {  
  "builder": "@angular-builders/custom-webpack:browser",  
  "options": {  
    ...  
    ,  
    "customWebpackConfig": {  
      "path": "projects/create-todo/build/extra-webpack.config.js",  
      "mergeStrategies": {  
        "externals": "replace"  
      }  
    }  
  }  
}
```

fixing the issue with web-pack

create-todo/build/extra-webpack.config.js

```
module.exports = {  
  output: {  
    jsonpFunction: 'webpackJsonpCreateTodo',  
    library: 'createTodo'  
  }  
};
```

hosting the web component in app

home.component.html

```
<div class="todo-create-container">
  <create-todo-element></create-todo-element>
  <create-todo></create-todo>
</div>
```

home.module.ts

```
@NgModule({
  declarations: [HomeComponent],
  imports: [CommonModule, CreateTodoAppModule],
  bootstrap: [HomeComponent],
  entryComponents: [HomeComponent],
  schemas: [CUSTOM_ELEMENTS_SCHEMA]
})
export class HomeModule {
}
```

What are we trying to do ?



Build, serve & deploy independently

Should be able to build, test, serve, lint and deploy independently without impacting other features. **The key is to slice it as features.**



Build as web-components

Make it more re-usable (like api)



Using toggles for progressive upgrades

Release features without impacting other features or regression through toggle mechanism



Technology agnostics

Use any framework (angular, vuejs, preact etc)

What are we trying to do ?

Using toggles for progressive upgrades

Release features without impacting other features or regression through toggle mechanism



feature toggle in app

home.component.html

```
<div class="container-fluid" style="width:100%">
  <div *ngIf="!loadElements">
    <create-todo></create-todo>
  </div>
  <div *ngIf="loadElements">
    <create-todo-element></create-todo-element>
  </div>
</div>
```

reducing the css payload

styles.scss

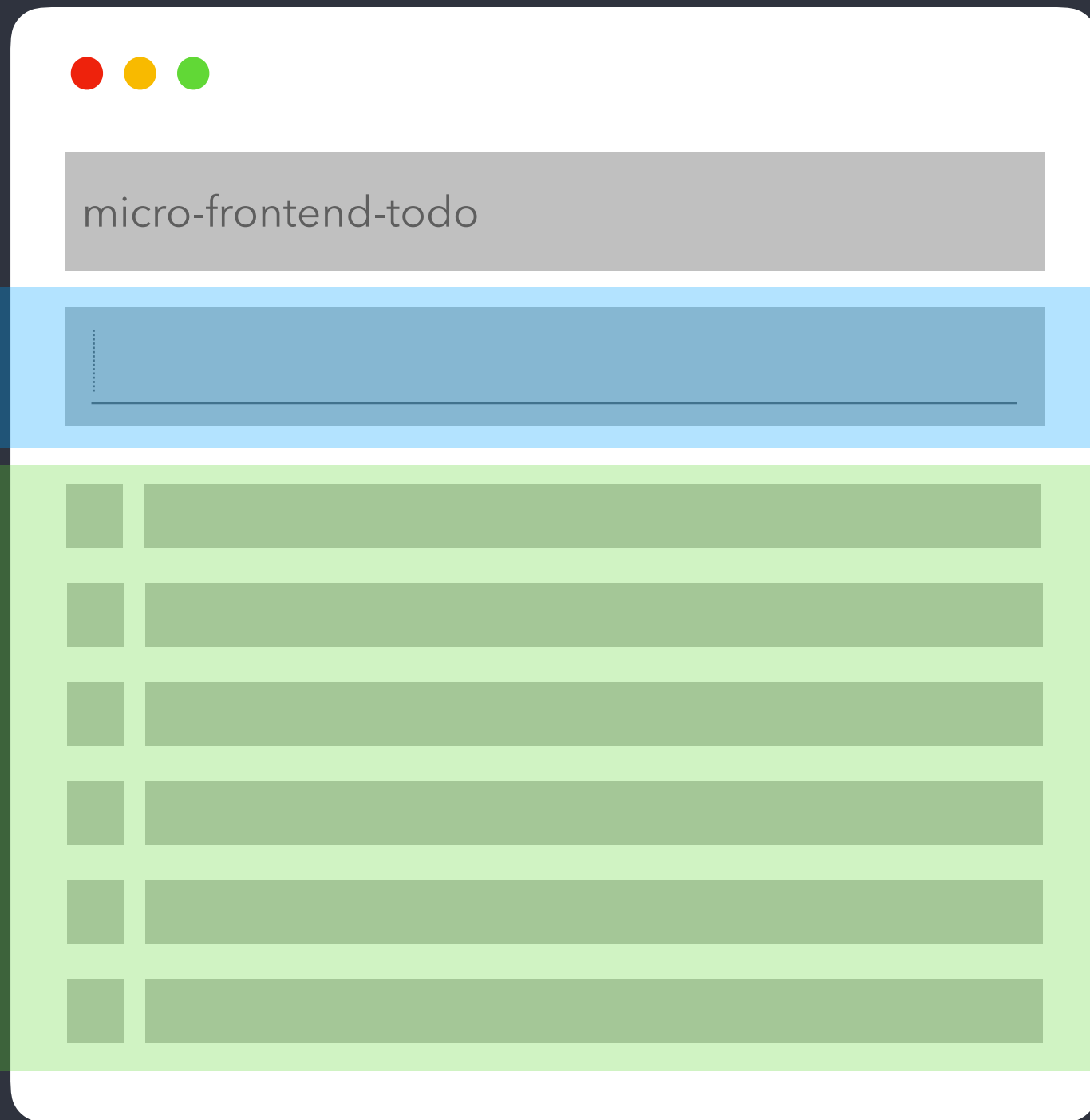
```
@import url('https://stackpath.bootstrapcdn.com/bootstrap/3.3.7/css/  
bootstrap.min.css');
```


progressive upgrade of bootstrap

home.component.scss

```
@import url('https://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/  
bootstrap.min.css');
```

feature teams



user should be able to create
todo items

user should be able to view all
the todo items created

demo

<http://bit.ly/codejam-git-final>

What are we trying to do ?



Build, serve & deploy independently

Should be able to build, test, serve, lint and deploy independently without impacting other features. **The key is to slice it as features.**



Build as web-components

Make it more re-usable (like api)



Using toggles for progressive upgrades

Release features without impacting other features or regression through toggle mechanism



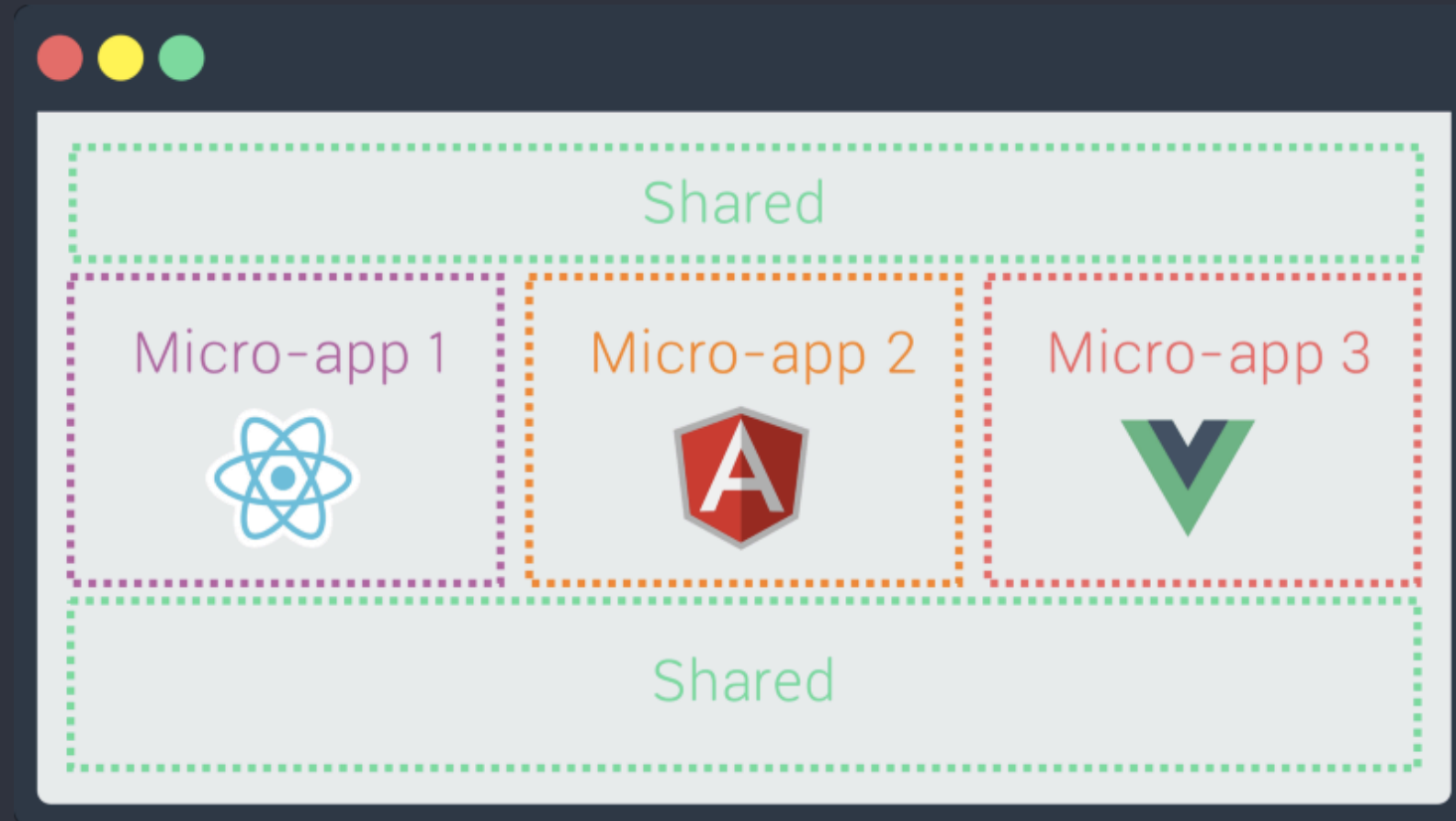
Technology agnostics

Use any framework (angular, vuejs, preact etc)

What are we trying to do ?

Technology agnostics

Use any framework (angular, vuejs, preact etc)



demo

where we could improve

Bundle size

Too much. Refer article in [medium](#) and ng-conf19. Ivy to the rescue

Shukriya !!!

HAPPY CODING