

28.05.2022

CI/CD DEPLOYMENT FOR SPRINGBOOT APPLICATION

Prepared By: PRIYANKA DAS

Welcome.component.html

```
<div class="card">
  <div class="card-header">
    Login
  </div>
  <div class="card-body">
    <div class="container-fluid">
      <div class="text-center">

        </div>

      <div>
        <form >
          <div class="container2">
            <input type="hidden" name="command"
value="LOGIN" />
            <label>Username : </label>
            <br/>
            <input type="text" placeholder="Enter
Username" name="username" required>
            <br/>
            <label>Password : </label>
            <br/>
            <input type="password" placeholder="Enter
Password" name="password" required>
            <br/>
            <button type="submit">Login</button>
            <br/>
          </div>
        </form>
      </div>
    </div>
  </div>
</div>
```

```

        <a >Don't have an account? Register now!</a>

    </div>
</form>
</div>

</div>
</div>
</div>

```

Product-detail.component.html

```

<div class='card'>
  <div class='card-header'>
    {{pageTitle + ': ' + product?.productName}}
  </div>

  <div class='card-body'
    *ngIf='product'>
    <div class='row'>
      <div class='col-md-8'>
        <div class='row'>
          <div class='col-md-4'>Name:</div>
          <div class='col-md-
8'>{{product.productName}}</div>
        </div>
        <div class='row'>
          <div class='col-md-4'>Code:</div>
          <div class='col-md-8'>{{product.productCode |
lowercase | convertToSpaces: '-'}}</div>

```

```

    </div>
    <div class='row'>
        <div class='col-md-4'>Description:</div>
        <div class='col-md-
8'>{{product.description}}</div>
    </div>
    <div class='row'>
        <div class='col-md-4'>Availability:</div>
        <div class='col-md-
8'>{{product.releaseDate}}</div>
    </div>
    <div class='row'>
        <div class='col-md-4'>Price:</div>
        <div class='col-md-
8'>{{product.price|currency:'USD': 'symbol'}}</div>
    </div>
    <div class='row'>
        <div class='col-md-4'>5 Star Rating:</div>
        <div class='col-md-8'>
            <pm-star [rating]='product.starRating'>
            </pm-star>
        </div>
    </div>
</div>

<div class='col-md-4'>
    <img class='center-block img-responsive'
        [style.width.px]='200'
        [style.margin.px]='2'
        [src]='product.imageUrl'
        [title]='product.productName'>

```

```

        </div>
      </div>
    </div>

    <div class='card-footer'>
      <button class='btn btn-outline-secondary'
        style='width:80px'
        (click)='onBack()'>
        <i class='fa fa-chevron-left'></i> Back
      </button>
    </div>
  </div>

```

Product-detail.component.spect.ts

```

import { ComponentFixture, TestBed } from
'@angular/core/testing';

import { ProductDetailComponent } from './product-
detail.component';

describe('ProductDetailComponent', () => {
  let component: ProductDetailComponent;
  let fixture: ComponentFixture<ProductDetailComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ ProductDetailComponent ]
    })
    .compileComponents();
  });

```

```
beforeEach(() => {
  fixture =
TestBed.createComponent(ProductDetailComponent);
  component = fixture.componentInstance;
  fixture.detectChanges();
});

it('should create', () => {
  expect(component).toBeTruthy();
});
});
```

Product.detail.guard.spec.ts

```
import { TestBed } from '@angular/core/testing';

import { ProductDetailGuard } from './product-
detail.guard';

describe('ProductDetailGuard', () => {
  let guard: ProductDetailGuard;
  beforeEach(() => {
    TestBed.configureTestingModule({});
    guard = TestBed.inject(ProductDetailGuard);
  });
  it('should be created', () => {
    expect(guard).toBeTruthy();
  });
});
```

Product-list.component.html

```
<div class='card'>
  <div class='card-header'>
    {{pageTitle}}
  </div>
  <div class='card-body'>
    <div class='row'>
      <div class='col-md-2'>Filter by:</div>
      <div class='col-md-4'>
        <input type='text'
          [(ngModel)]='listFilter' />
      </div>
    </div>
    <div class='row'
      *ngIf='listFilter'>
      <div class='col-md-6'>
        <h4>Filtered by: {{listFilter}}</h4>
      </div>
    </div>
    <div class='table-responsive'>
      <table class='table'
        *ngIf='products.length'>
        <thead>
          <tr>
            <th>
              <button class='btn btn-primary'
                (click)='toggleImage()'>
                {{showImage ? 'Hide' : 'Show'}} Image
              </button>
            </th>
```

```

        <th>Product</th>
        <th>Code</th>
        <th>Available</th>
        <th>Price</th>
        <th>5 Star Rating</th>
    </tr>
</thead>
<tbody>
    <tr *ngFor='let product of filteredProducts'>
        <td>
            <img *ngIf='showImage'
                [src]='product.imageUrl'
                [title]='product.productName'
                [style.width.px]='imageWidth'
                [style.margin.px]='imageMargin'>
        </td>
        <td>
            <a [routerLink]="['/products',
product.productId]">
                {{ product.productName }}
            </a>
        </td>
        <td>{{ product.productCode | lowercase |
convertToSpaces:'-' }}</td>
        <td>{{ product.releaseDate }}</td>
        <td>{{ product.price |
currency:'USD': 'symbol': '1.2-2' }}</td>
        <td>
            <pm-star [rating]='product.starRating'
(ratingClicked)='onRatingClicked($event)'>

```



```

        </pm-star>
      </td>
    </tr>
  </tbody>
</table>
</div>
</div>
</div>
<div *ngIf='errorMessage'
  class='alert alert-danger'>
  Error: {{ errorMessage }}
</div>

```

Products.json

```

[
  {
    "productId": 1,
    "productName": "Leaf Rake",
    "productCode": "GDN-0011",
    "releaseDate": "March 19, 2019",
    "description": "Leaf rake with 48-inch wooden
handle.",
    "price": 19.95,
    "starRating": 3.2,
    "imageUrl": "assets/images/leaf_rake.png"
  },
  {
    "productId": 2,
    "productName": "Garden Cart",
    "productCode": "GDN-0023",

```

```
    "releaseDate": "March 18, 2019",
    "description": "15 gallon capacity rolling garden
cart",
    "price": 32.99,
    "starRating": 4.2,
    "imageUrl": "assets/images/garden_cart.png"
  },
  {
    "productId": 5,
    "productName": "Hammer",
    "productCode": "TBX-0048",
    "releaseDate": "May 21, 2019",
    "description": "Curved claw steel hammer",
    "price": 8.9,
    "starRating": 4.8,
    "imageUrl": "assets/images/hammer.png"
  },
  {
    "productId": 8,
    "productName": "Saw",
    "productCode": "TBX-0022",
    "releaseDate": "May 15, 2019",
    "description": "15-inch steel blade hand saw",
    "price": 11.55,
    "starRating": 3.7,
    "imageUrl": "assets/images/saw.png"
  },
  {
    "productId": 10,
    "productName": "Video Game Controller",
    "productCode": "GMG-0042",
```

```
    "releaseDate": "October 15, 2018",
    "description": "Standard two-button video game controller",
    "price": 35.95,
    "starRating": 4.6,
    "imageUrl": "assets/images/xbox-controller.png"
  }
]
```

Environment.ts

```
export const environment = {
  production: false
};

/*
 * For easier debugging in development mode, you can
 * import the following file
 * to ignore zone related error stack frames such as
 * `zone.run`, `zoneDelegate.invokeTask`.
 *
 * This import should be commented out in production
 * mode because it will have a negative impact
 * on performance if an error is thrown.
 */
// import 'zone.js/plugins/zone-error'; // Included
// with Angular CLI.
```

Main.ts

```
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from
'@angular/platform-browser-dynamic';

import { AppModule } from '../app/app.module';
import { environment } from
'../environments/environment';

if (environment.production) {
  enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule)
  .catch(err => console.error(err));
```

Style.css

```
@import "~bootstrap/dist/css/bootstrap.min.css";
@import "~font-awesome/css/font-awesome.min.css";

div.card-header {
  font-size: large;
}

div.card {
  margin-top: 10px
}

.table {
  margin-top: 10px
```

```
}

a {
  text-decoration: none;
}
a:hover {
  text-decoration: underline;
}

button {
  justify-content: center;
  background-color: #4CAF50;
  width: 100%;
  color: white;
  padding: 15px;
  margin: 10px 0px;
  border: none;
  cursor: pointer;
}

form {

  transform: translate(50%, 1%);
  display: inline-block;
  border: 1.4px solid black;
  width: 45%;

}
input[type=text], input[type=password] {
```

```
justify-content: center;
width: 100%;
margin: 8px 0;
padding: 12px 20px;
display: inline-block;
border: 2px solid green;
box-sizing: border-box;
}
button:hover {
  opacity: 0.7;
}
```

```
.container2 {

  justify-content: center;
  padding: 15px;
  background-color: #FFF8DC;
}
```

Angular.json

```
{
  "$schema":
    "./node_modules/@angular/cli/lib/config/schema.json",
  "cli": {
    "analytics": "25966e0b-559a-45fe-b9d4-ae4bd482836a"
  },
  "version": 1,
  "newProjectRoot": "projects",
}
```

```
"projects": {
  "apm": {
    "projectType": "application",
    "schematics": {
      "@schematics/angular:application": {
        "strict": true
      }
    },
    "root": "",
    "sourceRoot": "src",
    "prefix": "pm",
    "architect": {
      "build": {
        "builder": "@angular-devkit/build-
angular:browser",
        "options": {
          "outputPath": "dist/apm",
          "index": "src/index.html",
          "main": "src/main.ts",
          "polyfills": "src/polyfills.ts",
          "tsConfig": "tsconfig.app.json",
          "assets": [
            "src/favicon.ico",
            "src/assets",
            "src/api"
          ],
          "styles": [
            "src/styles.css"
          ],
          "scripts": []
        },

```

```
"configurations": {
  "production": {
    "budgets": [
      {
        "type": "initial",
        "maximumWarning": "500kb",
        "maximumError": "1mb"
      },
      {
        "type": "anyComponentStyle",
        "maximumWarning": "2kb",
        "maximumError": "4kb"
      }
    ],
    "fileReplacements": [
      {
        "replace":
"src/environments/environment.ts",
        "with":
"src/environments/environment.prod.ts"
      }
    ],
    "outputHashing": "all"
  },
  "development": {
    "buildOptimizer": false,
    "optimization": false,
    "vendorChunk": true,
    "extractLicenses": false,
    "sourceMap": true,
    "namedChunks": true
  }
}
```



```
    },
    },
    "defaultConfiguration": "production"
  },
  "serve": {
    "builder": "@angular-devkit/build-angular:dev-
server",
    "configurations": {
      "production": {
        "browserTarget": "apm:build:production"
      },
      "development": {
        "browserTarget": "apm:build:development"
      }
    },
    "defaultConfiguration": "development"
  },
  "extract-i18n": {
    "builder": "@angular-devkit/build-
angular:extract-i18n",
    "options": {
      "browserTarget": "apm:build"
    }
  },
  "test": {
    "builder": "@angular-devkit/build-
angular:karma",
    "options": {
      "main": "src/test.ts",
      "polyfills": "src/polyfills.ts",
      "tsConfig": "tsconfig.spec.json",
```

```
        "karmaConfig": "karma.conf.js",
        "assets": [
            "src/favicon.ico",
            "src/assets"
        ],
        "styles": [
            "src/styles.css"
        ],
        "scripts": []
    }
}
},
"defaultProject": "apm"
}
```

Jenkinsfile

```
node(){
    stage('Cloning Git') {
        checkout scm
    }

    stage('Install dependencies') {
        nodejs('nodejs') {
            sh 'npm install'
            echo "Modules installed"
        }
    }
}
```

```
stage('Build') {
    nodejs('nodejs') {
        sh 'npm run build'
        echo "Build completed"
    }
}

stage('Package Build') {
    sh "tar -zcvf bundle.tar.gz dist/apm/"
}

stage('Artifacts Creation') {
    fingerprint 'bundle.tar.gz'
    archiveArtifacts 'bundle.tar.gz'
    echo "Artifacts created"
}

stage('Stash changes') {
    stash allowEmpty: true, includes:
'bundle.tar.gz', name: 'buildArtifacts'
}
}

node('awsnode') {
    echo 'Unstash'
    unstash 'buildArtifacts'
    echo 'Artifacts copied'

    echo 'Copy'
```

```
sh "yes | sudo cp -R bundle.tar.gz /var/www/html &&  
cd /var/www/html && sudo tar -xvf bundle.tar.gz"  
echo 'Copy completed'  
}
```

Karma.conf.js

Karma configuration file, see link for more information
// <https://karma-runner.github.io/1.0/config/configuration-file.html>

```
module.exports = function (config) {  
  config.set({  
    basePath: '',  
    frameworks: ['jasmine', '@angular-devkit/build-angular'],  
    plugins: [  
      require('karma-jasmine'),  
      require('karma-chrome-launcher'),  
      require('karma-jasmine-html-reporter'),  
      require('karma-coverage'),  
      require('@angular-devkit/build-angular/plugins/karma')  
    ],  
    client: {  
      jasmine: {  
        // you can add configuration options for Jasmine  
        here  
        // the possible options are listed at  
        https://jasmine.github.io/api/edge/Configuration.html  
      }  
    }  
  });  
}
```

```
    // for example, you can disable the random
    execution with `random: false`
    // or set a specific seed with `seed: 4321`
  },
  clearContext: false // leave Jasmine Spec Runner
output visible in browser
},
jasmineHtmlReporter: {
  suppressAll: true // removes the duplicated traces
},
coverageReporter: {
  dir: require('path').join(__dirname,
'./coverage/apm'),
  subdir: '.',
  reporters: [
    { type: 'html' },
    { type: 'text-summary' }
  ]
},
reporters: ['progress', 'kjhtml'],
port: 9876,
colors: true,
logLevel: config.LOG_INFO,
autoWatch: true,
browsers: ['Chrome'],
singleRun: false,
restartOnFileChange: true
});
};
```

Package.json

```
{
  "name": "apm",
  "version": "0.0.0",
  "scripts": {
    "ng": "ng",
    "start": "ng serve -o",
    "build": "ng build",
    "watch": "ng build --watch --configuration
development",
    "test": "ng test"
  },
  "private": true,
  "dependencies": {
    "@angular/animations": "~12.0.0",
    "@angular/common": "~12.0.0",
    "@angular/compiler": "~12.0.0",
    "@angular/core": "~12.0.0",
    "@angular/forms": "~12.0.0",
    "@angular/platform-browser": "~12.0.0",
    "@angular/platform-browser-dynamic": "~12.0.0",
    "@angular/router": "~12.0.0",
    "bootstrap": "^5.0.1",
    "font-awesome": "^4.7.0",
    "rxjs": "~6.6.0",
    "tslib": "^2.1.0",
    "zone.js": "~0.11.4"
  },
  "devDependencies": {
    "@angular-devkit/build-angular": "~12.0.0",
```

```
"@angular/cli": "~12.0.0",
"@angular/compiler-cli": "~12.0.0",
"@types/jasmine": "~3.6.0",
"@types/node": "^12.11.1",
"jasmine-core": "~3.7.0",
"karma": "~6.3.0",
"karma-chrome-launcher": "~3.1.0",
"karma-coverage": "~2.0.3",
"karma-jasmine": "~4.0.0",
"karma-jasmine-html-reporter": "^1.5.0",
"typescript": "~4.2.3"
}
}
```

Tsconfig.json

```
{
  "compileOnSave": false,
  "compilerOptions": {
    "baseUrl": "./",
    "outDir": "./dist/out-tsc",
    "forceConsistentCasingInFileNames": true,
    "strict": true,
    "noImplicitReturns": true,
    "noFallthroughCasesInSwitch": true,
    "sourceMap": true,
    "declaration": false,
    "downlevelIteration": true,
    "experimentalDecorators": true,
    "moduleResolution": "node",
```

```
"importHelpers": true,  
"target": "es2017",  
"module": "es2020",  
"lib": [  
  "es2018",  
  "dom"  
]  
},  
"angularCompilerOptions": {  
  "enableI18nLegacyMessageIdFormat": false,  
  "strictInjectionParameters": true,  
  "strictInputAccessModifiers": true,  
  "strictTemplates": true  
}  
}
```

X=====X=====X