VW Multi-Tenant Portal POC Plan/task

Wednesday, September 5, 2018 9:47 AM

Phase 1: Setup FE for multi-tenant portal app

- [✓] 1. Create a design for Multi-tenant Portal (8/29, 8/30)
- [✓] 2. Setup a react app with ES6 build, Lint and SASS compiler (8/30)
- [✓] 3. Create a basic layout to place different component (8/04)
- [✓] 4. Create a home page with header, Navigation, footer, Carousel and content components (8/04)
- [✓] 5. Configure basic component/images to change based on tenant (8/05)
- [✓] 6. Configure component to change **style** (placement) according to different tenant
  - [✓] a. Setup style data integration
  - [✓] b. Configure header component

- [✓] 7. Configure component to show/hide element according to different tenant
  - [✓] a. add all component images and related text to element API
  - [✓] b. Setup element data integration
  - [✓] c. Configure collage component to render all slides (images/text) based on data from element Api
  - [✓] d. Configure banner component to render all banners (images/text) based on data from element Api

- [✓] 8. Configure components to show **verbiage** according to selected tenant
  - [✓] a. Setup verbiage data integration
  - [✓] b. Configure toast messages

- [ ] 9. Implement toast notification to show Verbiage changes
- [✓] 10. Configure component to change verbiage according to different tenant
- [ ] 11. Create small JSON editor to modify style configuration (React JSON Editor - https://codepen.io/sandeep821/pen/wEXYKZ)
  - [ ] a. Create post methods to update style configuration without DB
  - [ ] b. Create UI to use style configuration post method
  - [ ] c. Configure database for style configuration end point

- [ ] 12. Verify everything together

Phase 2: Setup API for FE app

- [✓] 1. Setup a Nodes application to have API for Style, element and verbiage configuration (8/30)
- [✓] 2. Create API for Style, element and verbiage configuration
- [✓] 3. Setup proxy in react app to get data from API
- [✓] 4. In react app get data from API and replace static configurations
- [ ] 5. Verify everything together (with Phase 1)

Phase 3: Create Login

- [✓] 1. Create basic Login page with static values
- [✓] 2. Configure react app to change is behavior based on login values and tenant name
- [ ] 3. Add Logo and banner components in login page
- [ ] 4. Configure login page to show different logo and banner based on different tenant URL
- [ ] 5. Verify everything together (with Phase 2 and Phase 1)

Phase 4: Implement CMS (JSON editor) for all three configuration

- [ ] 1. Create post method to update style configuration
- [ ] 2. Create UI (JSON editor) for Style post method

☐ 3. Create post method to update verbiage configuration
☐ 4. Create UI (JSON editor) for verbiage post method
☐ 5. Create post method to update element configuration
☐ 6. Create UI (JSON editor) for element post method
☐ 7. Verify everything together (with Phase1, Phase2 and Phase3)

Phase 5: Add CMS for toast notification UI elements

☐ 1. Configure toast UI using JSON values
☐ 2. Create API endpoint for toast within Element data
☐ 3. Create post method to update toast data using DB
☐ 4. Create UI to update toast post method
☐ 5. Test

Note: Implement REDUX to move forward

Phase 6: Build dynamic theme which each tenant can configure using UI

☐ 1. Configure them style using JSON values
☐ 2. Create API endpoint for theme configuration
☐ 3. Create post method to update theme configuration data using DB
☐ 4. Create UI to update configuration data
☐ 5. Integrate theme with login credential
☐ 6. Test

Phase 7: Implement internalization / localization

☐ 7. Use 'react-intl' to implement internalization / localization
☐ 8. Use at-least 2 locales
☐ 9. Every component and all static date should have defined locale key
☐ 10. Test

Phase 8: Deploy whole app in AWS

☐ 1. Deploy API
☐ 2. Deploy Db
☐ 3. Deploy FE App

Phase 9: Create CICD pipeline

☐ 4. Create CICD pipeline for FE repo
☐ 5. Create CICD pipeline for BE repo

Phase 10:

☐ 6. Demo

POC URL - http://10.136.66.66:3000/

Audi:
user : audi, password: demo

Bugatti:
user : bugatti, password: demo

Feature:
Multitenancy, Multitenancy configuration CMS, UI CMS, Dynamic theme, theme CMS,

Initialization/Localization, Cloud configuration

Tech stack: HTML5, ES6, SASS (node-sass-chokidar), React, Redux, JEST, react-intel, bootstrap, Webpack, VS Code, NodeJS, ExpressJS, DynamoDB, Git, GitHub, Concourse CI, Docker, AWS cloudFront (S3), AWS Lambda

Repos: FE, CMS, BE for UI Configuration

Demo Slides

Tuesday, September 18, 2018 10:06 AM

Slides:

1. Feature list with brief (1)
2. Tech stack with usage in table (1)
3. Architecture (3)
4. Feature details with above modules and bullet points (3)
5. App Demo

GIT Readme file

Tuesday, September 18, 2018 10:07 AM

Readme file:

1. About app
2. Install both repo (FE and BE)
3. Setup your own AWS database and replace DB in repo
4. How to run
5. Architecture
6. Tech stack
7. Feature list

App Features

Thursday, September 20, 2018 11:01 AM

Features:

1. Multitenancy
2. Multitenancy configuration CMS
3. UI CMS
4. Dynamic theme
5. Theme CMS
6. Initialization/Localization (react-intl)
7. Toast (react-toastify)
8. SASS (node-sass-chokidar)
9. Serverless API using Nodejs o AWS

Tech stack

Thursday, September 20, 2018 11:04 AM

Tech stack: HTML5, ES6, SASS (node-sass-chokidar), React, Redux, JEST, react-intel, bootstrap, Webpack, VS Code, NodeJS, ExpressJS, DynamoDB, Git, GitHub, Concourse CI, Docker, AWS cloudFront (S3), AWS Lambda

------ Installed dependencies-------------

Thursday, September 20, 2018 11:33 AM

```
 1. "axios": "^0.18.0",
 2. "bootstrap": "^4.1.3",
 3. "codemirror": "^5.40.0",
 4. "node-sass-chokidar": "^1.3.3",
 5. "npm": "^6.4.0",
 6. "npm-run-all": "^4.1.3",
 7. "react": "^16.4.2",
 8. "react-bootstrap": "^0.32.3",
 9. "react-codemirror2": "^5.1.0",
10. "react-dom": "^16.4.2",
11. "react-jsonschema-form": "^1.0.4",
12. "react-router-dom": "^4.3.1",
13. "react-scripts": "1.1.4",
14. "react-toastify": "^4.3.0"
```

Internationalize (react-intl)

Thursday, September 20, 2018   11:05 AM

Internationalize React apps. This library provides React components and an API to format dates, numbers, and strings, including pluralization and handling translations.

From <https://www.npmjs.com/package/react-intl>

Toast (react-toastify)

Thursday, September 20, 2018 11:28 AM

React-Toastify allow you to add toast notification to your app with ease.

- Can display a react component inside the toast !
- Don't rely on findDOMNode
- Has onOpen and onClose hooks. Both can access the props passed to the react component rendered inside the toast
- Can be positioned
- Define behavior per toast
- Easy to setup
- Super easy to customize

From <https://www.npmjs.com/package/react-toastify/v/1.4.3>

UI Componet Lib (react-bootstrap)

Thursday, September 20, 2018 11:32 AM

CSS Lib (bootstrap)

Thursday, September 20, 2018 11:32 AM

Http calls (axios)

Thursday, September 20, 2018 11:33 AM

Promise based HTTP client for the browser and node.js

- Make XMLHttpRequests from the browser
- Make http requests from node.js
- Supports the Promise API
- Intercept request and response
- Transform request and response data
- Cancel requests
- Automatic transforms for JSON data
- Client side support for protecting against XSRF

From <https://github.com/axios/axios>

Sass (node-sass-chokidar)

Thursday, September 20, 2018 10:59 AM

A thin wrapper around node-sass executable to use chokidar instead of Gaze when watching files.

From <https://www.npmjs.com/package/node-sass-chokidar>

Implemented this to have advance CSS in app

react-jsonschema-form

Thursday, September 20, 2018 11:36 AM

A simple React component capable of building HTML forms out of a JSON schema and using Bootstrap semantics by default.

From <https://www.npmjs.com/package/react-jsonschema-form>

codemirror

Thursday, September 20, 2018 11:37 AM

CodeMirror is a versatile text editor implemented in JavaScript for the browser. It is specialized for editing code, and comes with over 100 language modes and various addons that implement more advanced editing functionality. Every language comes with fully-featured code and syntax highlighting to help with reading and editing complex code.

From <https://www.npmjs.com/package/codemirror>

react-codemirror2

Thursday, September 20, 2018 11:38 AM

`react-codemirror2` ships with the notion of an uncontrolled and controlled component. `UnControlled` consists of a simple wrapper largely powered by the inner workings of `codemirror` itself, while `Controlled` will demand state management from the user, preventing codemirror changes unless properly handled via `value`.

From <https://www.npmjs.com/package/react-codemirror2>

Webpack scripts (react-scripts)

Thursday, September 20, 2018 11:39 AM

DOM manipulation (react-dom)

Thursday, September 20, 2018 11:39 AM

npm-run-all

Thursday, September 20, 2018 11:41 AM

A CLI tool to run multiple npm-scripts in parallel or sequential.

From <https://www.npmjs.com/package/npm-run-all>

npm

Thursday, September 20, 2018 11:42 AM

npm opens up an entire world of JavaScript talent for you and your team. It's the world's largest software registry, with approximately 3 billion downloads per week. The registry contains over 600,000 *packages* (building blocks of code). Open-source developers from every continent use npm to share and borrow packages, and many organizations use npm to manage private development as well.

From <https://docs.npmjs.com/getting-started/what-is-npm>