Video Manager (Angular 2 + Cloudinary, Test Single Page Application)

Description:

Test Scope: Interactive web application that allows the user to play videos from a predefined list.

Examples:

http://veeca.me/fetchvideo/
http://veeca.me/fetchvideo/dashboard
http://veeca.me/fetchvideo/detail/1
http://veeca.me/fetchvideo/detail/2
http://veeca.me/fetchvideo/detail/3
http://veeca.me/fetchvideo/detail/4
http://veeca.me/fetchvideo/detail/5

!!! **http://res.cloudinary.com/sdk-test/raw/upload/video-list-home-task.json, contains invalid JSON format, so I couldn't use this URL.**

!!! Car Attached with cute cat video: https://res.cloudinary.com/sdk-test/raw/upload/car-pan-up.json

Code on Git

https://github.com/Kasha/VideoManager.git

Technology:

1. Angular 2 (version 5), Typescript, JavaScript, Cloudinary for Angular 2 (SDK)
2. Dev environment:
    a. Nodejs and npm for installation
    b. Angular cli – compiler and server (`ng serve –aot`)
    - Aot, Angular 2 version 5, quick and improved Angular syntax and Typescript compilation

Liad Kashanovsky
Liadky@gmail.com
25/2/2018

Content

1. Video Manager Single Page Responsive Application:
    a. View Video List, view expanded details for each video, view video
    b. 3 Views with Message Console:
        i. **Detail View:** Explicit video link, http://[uri]/detail/[video id]
        ii. **Video List View:** Video list (Default landing http://[uri]), http://[uri]/VideoManager
        iii. **Dashboard View:** Dashboard link, http://[uri]/dashboard
        iv. **Message Console**: Each View contains Message console reporting of Application Components and Services get and view videos activities
2. Video Manager Angular project
3. Video Manager Dist Publish folder (Copy Paste to any host server)

Functionality:

1. **App Functions:**
    a. Navigate to VideoManager, Dashboard or to Detail Views (Video Details)
    b. Messages used for reporting and displaying components and services getVideos, getVideo, View Details activities.
    c. Fetch Video List, Fetch with late binding detailed video JSON file for each video item in list
        i. Asynchronous Video List loading using Angular Service Observer and Angular Component subscribe – **Video List, getVideos**.
            1. List is loaded when requesting getVideos for the first time explicit from VideoManager or Dashboard or implicit from DetailsView
            2. getVideos is requested from Angular Component ngOnInit
            3. getVideos is called from root component VideoManagerComponent (VideoManager View) or from DashboardComponent (Dashboard View) (depended on panel user navigated to)
            4. getVideos is also being called from VideoDetailComponent when user navigates to video and video list wasn't requested yet.
            5. After loading, video list is cached with VideoService.videos
        ii. Late binding of video Details – **VideoDetails Item, getVideo**:
            1. Each video in list contains a video details URL to JSON file
            2. These items are loaded after video list was finished loading or when user navigates to video and it wasn't loaded or requested yet.
2. **Single Page Application with 3 Views and Message Console:**
    a. Message Console:

Liad Kashanovsky
Liadky@gmail.com
25/2/2018

i. Message Console, shared by all components, services and viewed in Dashboard, Video List, and Video Details panes.
ii. Message Console displays Loading and viewing video user and app activities

b. VideoManager (root view):

i. Video List display
ii. Message display console

c. Detail (Video Details):

i. Display of video name, description and responsive video image with video to play.
ii. Message display console.

d. Dashboard :

i. Top 3 Videos display: 3 clickable buttons each with video name (3 first items)
ii. Click on button, navigate and replace to video details view
iii. Message display console.
iv. Suggested features: Upload, Delete, and Publish URL features could be implemented in Dashboard.

Liad Kashanovsky
Liadky@gmail.com
25/2/2018

Structure:

1. **Video Model data:**
    a. **Class:**

```
export class Video
{
   id: number;
   name: string; // Video Name
   url: string; // URL to Json with Video Details
   image: string; //Video Image
   details: VideoDetails ;
}
```

    b. **JSON example:**

    Predefined Video list JSON file: http://veeca.me/fetchvideo/cloudinary.json

    !!! **http://res.cloudinary.com/sdk-test/raw/upload/video-list-home-task.json, contains invalid JSON format, so I couldn't use this URL.**

2. **VideoDetails Model Data:**
    a. **Class:**

```
export class VideoDetails
{
   name: string; // Video Title
   url: string; //Video URL
   description: string ; //Video Description
}
```

    b. **JSON example:**

    https://res.cloudinary.com/sdk-test/raw/upload/cute-cat.json

3. **Components:**
    c. AppComponent, DashboardComponent, VideoDetailsComponent, VideoManagerComponent, MessagesComponent
4. **Services:**
    a. VideoService – Retrieves JSON Files, Videos and Video Details
    b. MessageService

Liad Kashanovsky
Liadky@gmail.com
25/2/2018

Product suggestion features:

5. Login
6. Import user list of uploaded videos fromDB
7. Upload Video
8. Delete Video
9. Search video with in user's uploaded videos
10. Add Cloudinary features for creating and publish video URL with Quality, Size and more URL settings
11. Clodinary Statistics for users uploaded videos, such as views.

Development Project Setup:

npm install bootstrap@next –save
npm install tether
npm install jquery –save
npm install cloudinary-core @cloudinary/angular-5.x --save
npm update
ng serve –aot (Angular 2, version 5, fast and improved compiler + run web server)
URL: http://localhost:4200

Deployment:

https://coursetro.com/posts/code/112/Angular-5-Deployment---Deploy-your-Angular-App

1. Set environment.ts
   production: true
2. You could take the files in the */dist* folder and upload them to a server. Just note, if you're uploading them to a sub folder, you will need to run the following build command:
   ng build --prod --base-href="myURL"

   else

   ng build –prod

3. After build **dist** folder is created, copy paste it to any web server and start playing

Liad Kashanovsky
Liadky@gmail.com
25/2/2018