

Video in canvas

```
[{
  index: 0,
  sentence: "This is a simple Javascript test",
  media: "https://miro.medium.com/max/1024/1*OK8xc3Ic6EGYg2k6BeGabg.jpeg",
  duration: 3
}, {
  index: 1,
  sentence: "Here comes the video!",
  media: "https://media.gettyimages.com/videos/goodlooking-young-woman-in-casual-clothing-is-painting-in-workroom-video-id1069900546",
  duration: 5
}]
```

Assume the response above is the 2 scenes data for an 8 seconds long 16:9 video. You are required to draw an HTML canvas to play those scenes in order.

Expected output: the video will **play, pause and stop** when user click on the canvas.

Extra efforts:

- Write in **React hook** to handle the playing state
- Play the video **from 5th second** when the second scene starts
- Text animations. Eg. typing text
- Scene transitions
- Play music in background

Question 2

You are tasked with creating a simple API following the requirements handed to you. This task is not meant to be strict, if you find yourself having to make decision, just choose whatever you think benefits you the most.

As a final note, remember that this is more about showcasing your knowledge than about solving a problem.

Requirements

This task is about building an image search API based on user's input. You'll have to get images from a few 3rd party services, compile the results and then return to the user. Here's the details for you App:

- You'll need to create a route(it can be Post or Get) for the API call, and there'll have an input for the user to key in the **search term (String)** to search images.
- Assume that we plan to have **3** 3rd party image libraries([Unsplash](#), [Pixabay](#), [Storyblocks](#)) as our image source, you'll have to register a free **account** for each image library, get the **api key**, and then integrate the library service in the app, make sure we'll be able to get images from all the image library.
- Assume that you're able to get images from all the images library that you've integrated in the app based on the user input, then we're expecting you to compile the results together before you return to the user. The results will be **object in array**, here's the details of the object as below:

```
{  
  image_ID: String, the ID of the image  
  thumbnails: String, thumbnails url of the image  
  preview: String, preview url of the image  
  title: String, preview url from the image  
  source: String, which image library you get this image from? [Unsplash, Storyblocks, Pixabay]  
  tags: Array the tag/keywords of the images (if any)  
}
```

here's a sample result of **Cat** :

```
[  
  {  
    "image_ID": "NodtnCsl4TE",  
    "thumbnails": "https://images.unsplash.com/photo-1494256997694-768df6688cc7?ixlib=rb-1.2.1&q=80&fm=jpg&crop=entropy&cs=tinysrgb&w=200&fit=max&ixid=eyJhcHBfM010JkYMDY5fQ",  
    "preview": "https://images.unsplash.com/photo-1494256997694-768df6688cc7?ixlib=rb-1.2.1&q=80&fm=jpg&crop=entropy&cs=tinysrgb&w=1680&fit=max&ixid=eyJhcHBfM010JkYMDY5fQ",  
    "title": "Crouching Tiger, Hidden Dragon.\nPhoto of a kitten named William hiding under a plaid",  
    "source": "Unsplash",  
    "tags": [ "cat", "animal", "pet" ]  
  },  
  {  
    "image_ID": "2083492",  
    "thumbnails": "https://cdn.pixabay.com/photo/2017/02/20/18/03/cat-2083492_150.jpg",  
    "preview": "https://pixabay.com/get/54e8d4404e5bae14e089857ec029337f163cc3e45656724f73287dd196_1280.jpg",  
    "title": null,  
    "source": "Pixabay",  
    "tags": [ "cat, young animal, curious" ]  
  },  
  {  
    "image_ID": "559529",  
    "thumbnails": "https://d1ynikh78jjrr.cloudfront.net/image/thumbnail/rDthN98Qoishumwih/vvv-0216-img_0053-3-981_thumb.jpg",  
    "preview": "https://d1ynikh78jjrr.cloudfront.net/image/preview/rDthN98Qoishumwih/vvv-0216-img_0053-3-981_PM.jpg",  
    "title": "Portrait of cute thai cat",  
    "source": "Storyblocks",  
    "tags": [  
      "animals",  
      "cat",  
      "color",  
      "feline",  
      "image",  
      "kitten",  
      "kitty",  
      "look",  
      "mammal",  
      "pet",  
      "point",  
      "portrait",  
      "pussy",  
      "studio",  
      "thai",  
      "watch",  
      "white"  
    ]  
  }  
]
```

What we expect from you

- You know how to synchronize your asynchronous code. In this case, we would like to see all the **image api** in your application is triggered **simultaneously**, instead of trigger one followed by one (e.g. You get images from Unsplash first, and then trigger the next image API call from StoryBlocks.)
- You're able to handle error from 3rd party API calls, whether it's a **response error** or **null** result. Eg: Pixabay returns null results, and Unsplash return response error, but the user still able to get images from StoryBlocks.
- The whole application is built on **Node Js**.

Optional task: *[completing this will get special considerations]:*

- You're able to build an **authentication** feature, which includes Registration, Login/Logout, and only allow registered users to access your image API (eg. only registered user with **valid token** can access your API) **Level: intermediate**
- For the web services part, you use **GraphQL** instead of **REST**. **Level: Hard**
- You're able to build a proper **infrastructure** for you application, and deploy to cloud server(**AWS/Google Cloud/Azure...**). **Level: Challenging**
- Your application can handle **> 1000s req/s** without breaking a sweat(**Please provide us your load test method.**) **Level: Insane**

Programming language: JavaScript is preferred. If you need to use another language, please document well in your code on your approach.