What do we learn in this project?

Youtube api link:
https://developers.google.com/youtube/v3

Add YouTube functionality to your site

With the YouTube Data API, you can add a variety of YouTube features to your application. Use the API to upload videos, manage playlists and subscriptions, update channel settings, and more.

Get started   Implementation guide

Search for content

Use the API to search for videos matching specific search terms, topics, locations, publication dates, and much more. The APIs `search.list` method also supports searches for playlists and channels.

Search for content

Other Resources

Tools

The APIs Explorer lets you test unauthorized and authorized requests. The Quota Calculator shows how different requests impact your quota usage.

Code Samples

Use our code samples to jump-start your project. Samples are available for Apps Script, Go, Java, JavaScript, .NET, PHP, Python, and Ruby.

Java   Python   More

https://developers.google.com/youtube/v3/getting-started

API supports.

Before you start

1. You need a Google Account to access the Google API Console, request an API key, and register your application.

2. Create a project in the Google Developers Console and obtain authorization credentials so your application can submit API requests.

3. After creating your project, make sure the YouTube Data API is one of the services that your application is registered to use:

   a. Go to the API Console and select the project that you just registered.

   b. Visit the Enabled APIs page. In the list of APIs, make sure the status is **ON** for the **YouTube Data API v3**.

4. If your application will use any API methods that require user authorization, read the authentication guide to learn how to implement OAuth 2.0 authorization.

5. Select a client library to simplify your API implementation.

6. Familiarize yourself with the core concepts of the JSON (JavaScript Object Notation) data format. JSON is a common, language-independent data format that provides a simple text representation of arbitrary data structures. For more information, see json.org.

Google Cloud   My Project 81218 ▾   Search (/) for resources, docs, products, and more   🔍 Search

API APIs & Services

APIs & Services   ＋ ENABLE APIS AND SERVICES

- Enabled APIs & services
- Library
- Credentials
- OAuth consent screen
- Page usage agreements

1 hour   6 hours   12 hours   ✓ 1 day   2 days   4 days

Traffic   ≈ ⩗ ⋮

⚠ No data is available for the selected time frame.

Errors

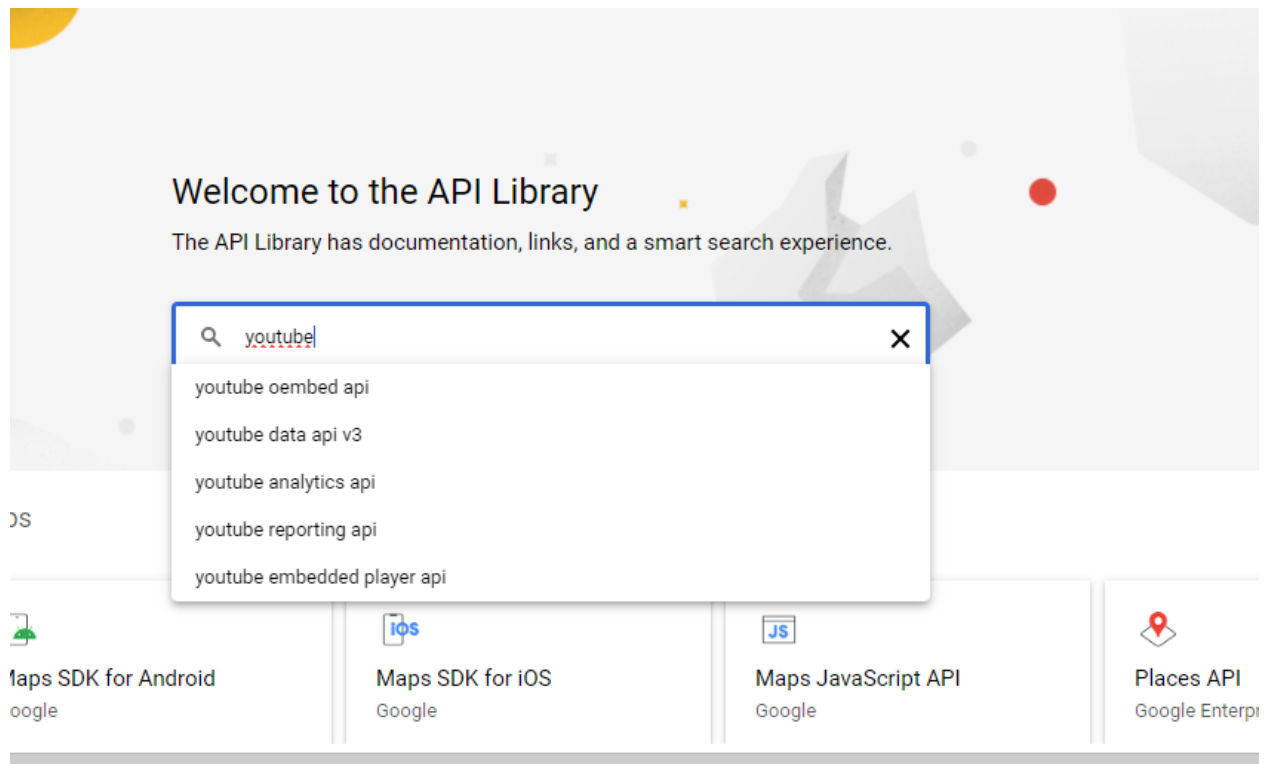⚠ No data is available for the selected tim

Welcome to the API Library

The API Library has documentation, links, and a smart search experience.
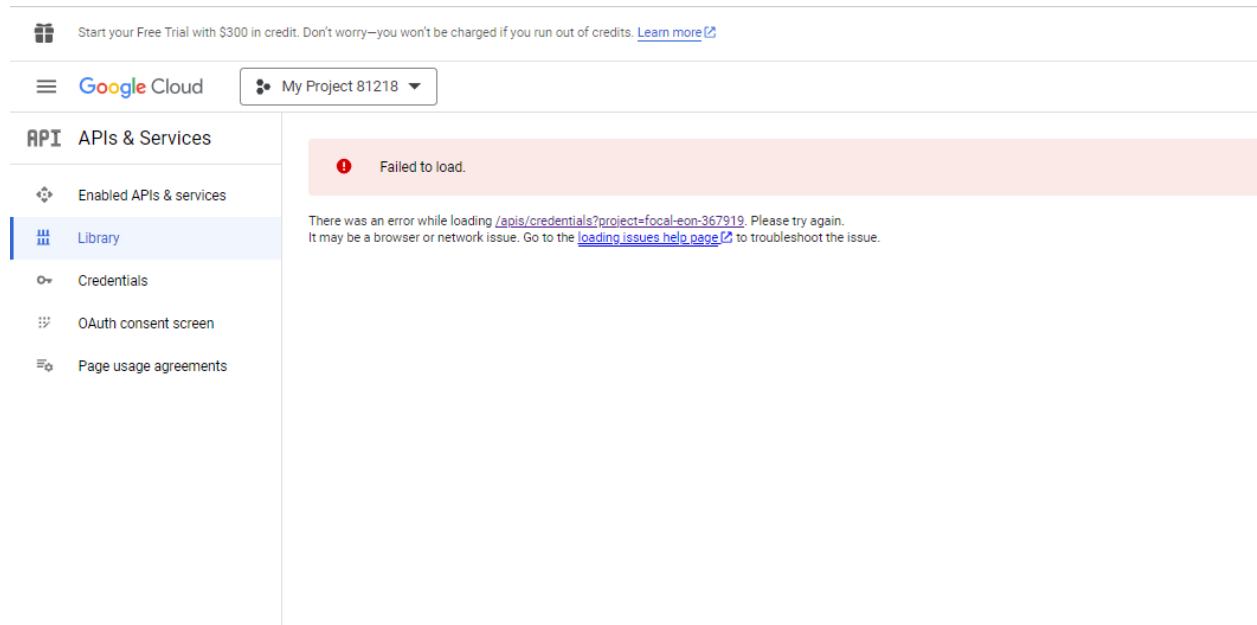
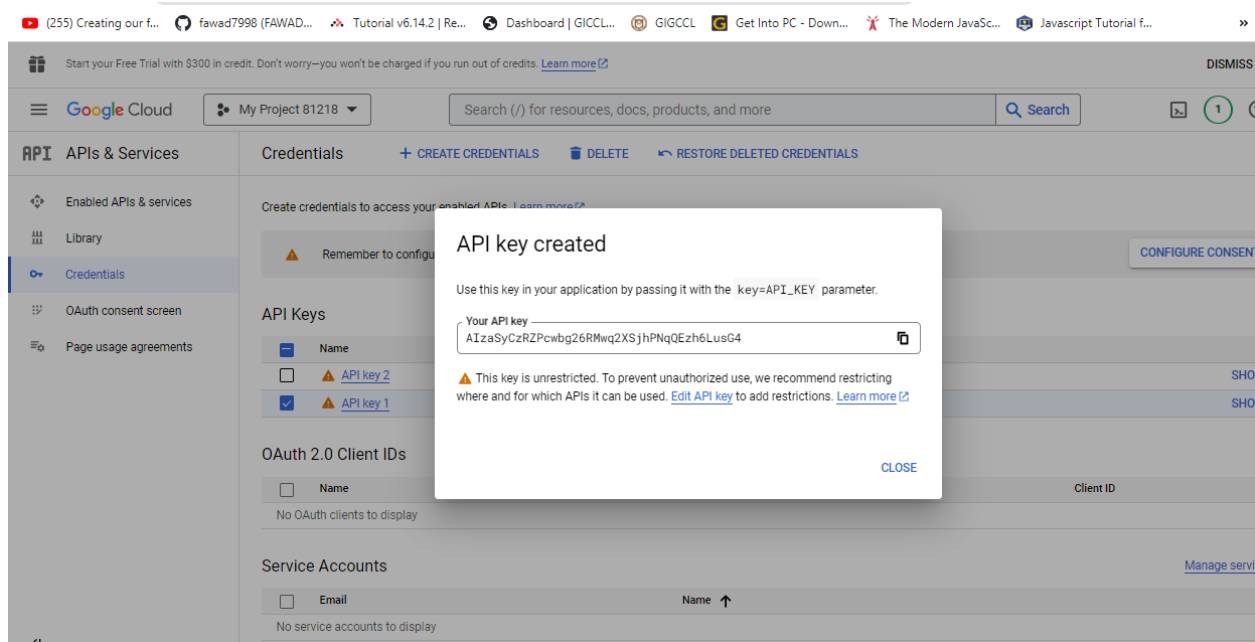🔍 Search for APIs & Services
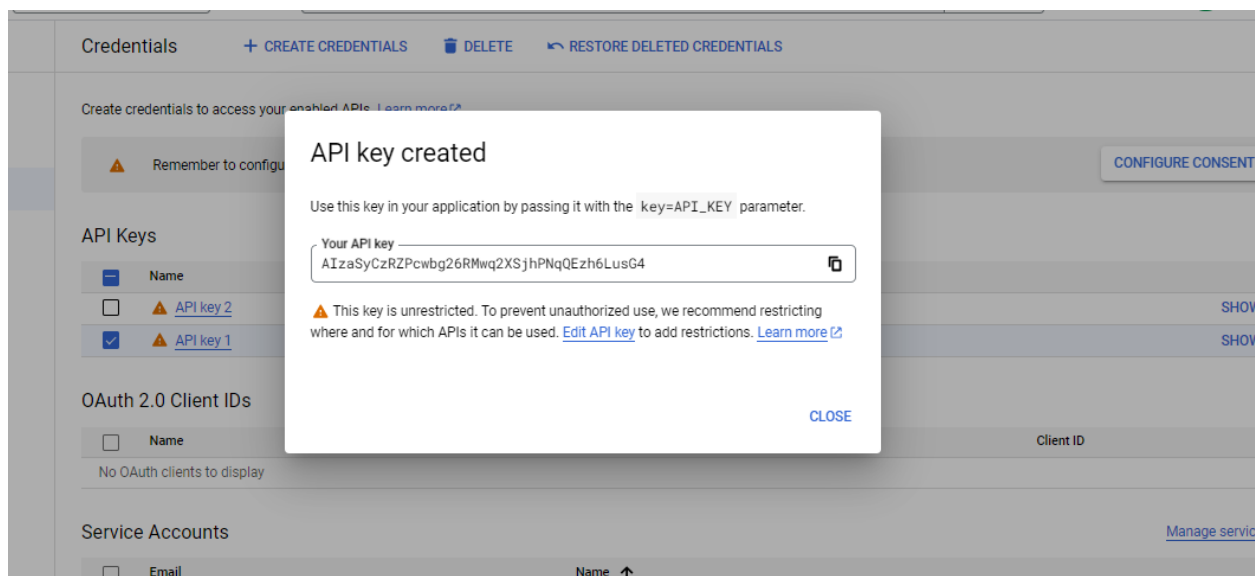
Maps

∧

Click youtuve data api v3

Then go back to page of

Then go to credentials

Click the create credentials

Then click the edit api key.
Then click the website and click the restrict key and then click the youtube api v3 and then save the key

Then go to your project and paste the key .

```
import React from 'react'


const App = () => {
  const apikey="AIzaSyCzRZPcwbg26RMwq2XSjhPNqQEzh6LusG4"
  return (
```

```
    <div>
      hello

    </div>
  )
}

export default App
```

After that same work

```
import React, { useEffect, useState } from 'react'
import axios from 'axios'


const App = () => {
  // const apikey=`AIzaSyCzRZPcwbg26RMwq2XSjhPNqQEzh6LusG4`
  // const url = `https://www.googleapis.com/youtube/v3/`
  const [data, setData] = useState([])
  const [search,setSearch] = useState("")
  const
url2=`https://www.googleapis.com/youtube/v3/search?part=snippet&key=AIzaSy
CzRZPcwbg26RMwq2XSjhPNqQEzh6LusG4&maxResults=500`
  useEffect(()=>{
   async function fetchvideos(){
      const res = await fetch(url2)
      const data2 = await res.json()
      console.log(data2)
      setData(data2.items)
   }
   fetchvideos();

  },[])
  const clickMe =async()=>{
    const
api2=`https://www.googleapis.com/youtube/v3/search?q=${search}&part=snippe
t&key=AIzaSyCzRZPcwbg26RMwq2XSjhPNqQEzh6LusG4&maxResults=500`
    const response = await fetch(api2)
```

```
    const data2 = await response.json()
    console.log(data2)
    setData(data2.items)


}
// const fetchvideo =()=>{
//    axios.get(url),{
//      param:{
//        key:apikey,
//        q:"",
//        type:"video",
//        part:"snippet",
//        maxresults:10

//      }
//    }.then((response)=>{
//      console.log(response)

//    }).catch((error)=>{
//      console.log(error)

//    })


// }
return (
  <>
  <input type="text" onChange={(e)=>{
    setSearch(e.target.value)
  }} />
  <button onClick={clickMe}>search</button>
  <div>
     {
    data.map((pack)=>{
      const {snippet}=pack
      return(
        <>
        <img src={snippet.thumbnails.medium.url} alt="" />
        <h2>{snippet.title}</h2>
```

```
            <h2>{snippet.channelTitle
}</h2>



            </>
        )


    })
    }


    </div>




    </>
  )
}

export default App
```