Api used : [YouTube Data API  |  Google Developers](https://developers.google.com/youtube/v3)

Library used : axios

Step 1: Generate api key. This needs to be private because someone may use a public api key leading to an increase in traffic in the same. This may cause financial losses to the api key owner.

One such api key is "AIzaSyDh4lSGdVGOnOg9oEvLTeLeCvl1eH5UMb0"

Step 2 : Install library axios, node. Host a local server and embed these urls in the same.

Step 3: Create route to get the search stream. We also need to capture the response of this route here.

Step 4: Declare const url as <https://www.googleapis.com/youtube/v3>

Explanation why above is needed.

Source where all the google apis is present : [googleapis.com](http://www.googleapis.com)

Shows we are working on which domain of the api, i.e., youtube

Shows the version of the api, i.e., v3

Step 5: Further in this we have search (which determines what are we searching for). It can be videos, playlist, songs, whatever we want.

Written as search?key=${apiKey}

Step 6: We need to provide our api key which we created earlier, without which it wouldn’t work.

Step 7: Further we have query parameters to filter our request. Declare type = video, because we need to filter the spiderman videos.

Step 8: After configuring everything mentioned above our url would look like `${apiUrl}/search?key=${apiKey}&type=video&part=snippet&q=${searchQuery}`.

Step 8 : Now we need to filter the response we got through route in step 3.

Step 9 : The response is actually quite large and is an array of objects. But we are only concerned about the data part of the response. We filter the data on basis of various keywords like spiderman spoiler, spoiler alert, upcoming series story, upcoming spiderman, etc.

Step 10 : We need to watch youtube only through this server.

Through these measures we can avoid spoilers on youtube.