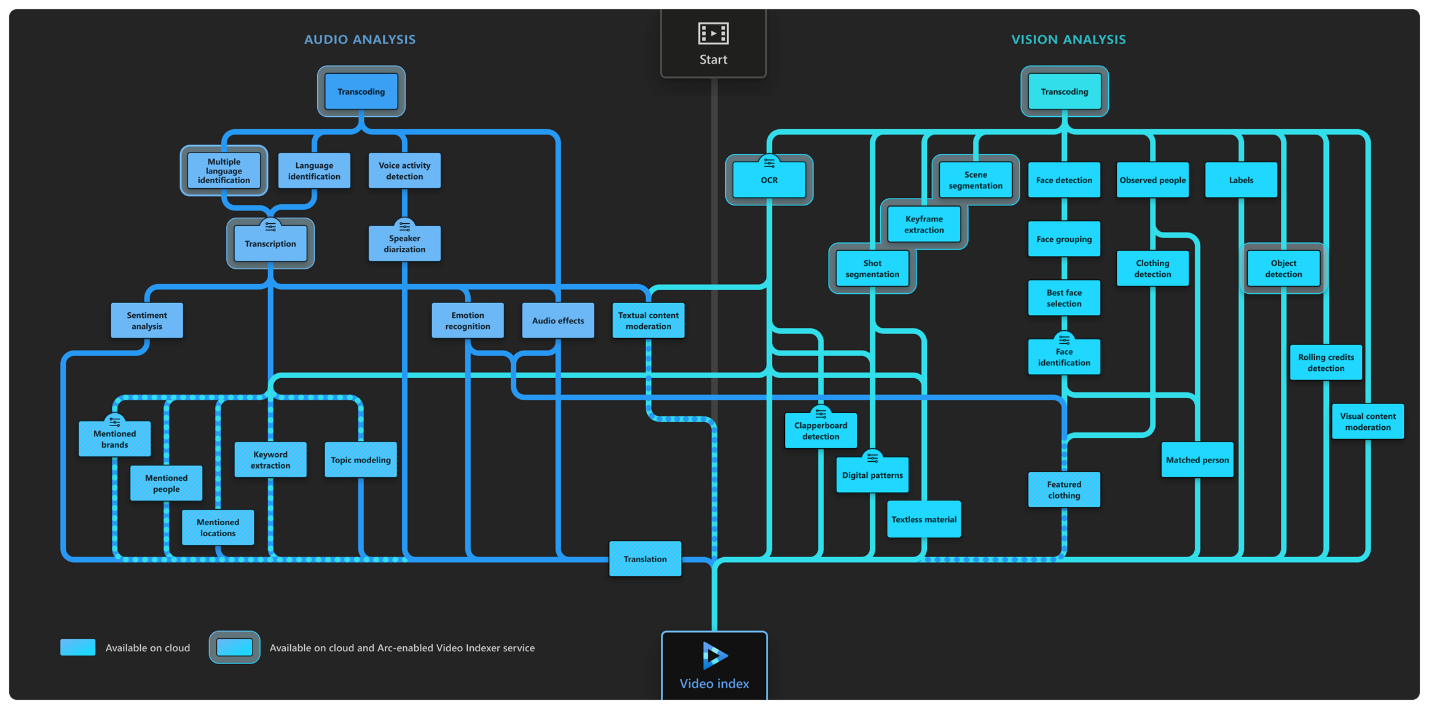
Reference Link: <https://learn.microsoft.com/en-us/azure/azure-video-indexer/video-indexer-overview>

Azure AI Video Indexer is a cloud application, part of Azure AI services, built on Azure AI services (such as the Face, Translator, Azure AI Vision, and Speech). It enables you to extract the insights from your videos using Azure AI Video Indexer video and audio models.



### Video models

* **Face detection**: Detects and groups faces appearing in the video.
* **Celebrity identification**: Identifies over 1 million celebrities—like world leaders, actors, artists, athletes, researchers, business, and tech leaders across the globe. The data about these celebrities can also be found on various websites (IMDB, Wikipedia, and so on).
* **Account-based face identification**: Trains a model for a specific account. It then recognizes faces in the video based on the trained model. For more information, see [Customize a Person model from the Azure AI Video Indexer website](https://learn.microsoft.com/en-us/azure/azure-video-indexer/customize-person-model-with-website) and [Customize a Person model with the Azure AI Video Indexer API](https://learn.microsoft.com/en-us/azure/azure-video-indexer/customize-person-model-with-api).
* **Thumbnail extraction for faces**: Identifies the best captured face in each group of faces (based on quality, size, and frontal position) and extracts it as an image asset.
* **Optical character recognition (OCR)**: Extracts text from images like pictures, street signs and products in media files to create insights.
* **Visual content moderation**: Detects adult and/or racy visuals.
* **Labels identification**: Identifies visual objects and actions displayed.
* **Scene segmentation**: Determines when a scene changes in video based on visual cues. A scene depicts a single event and it's composed by a series of consecutive shots, which are semantically related.
* **Shot detection**: Determines when a shot changes in video based on visual cues. A shot is a series of frames taken from the same motion-picture camera. For more information, see [Scenes, shots, and keyframes](https://learn.microsoft.com/en-us/azure/azure-video-indexer/scenes-shots-keyframes).
* **Black frame detection**: Identifies black frames presented in the video.
* **Keyframe extraction**: Detects stable keyframes in a video.
* **Rolling credits**: Identifies the beginning and end of the rolling credits in the end of TV shows and movies.
* **Editorial shot type detection**: Tags shots based on their type (like wide shot, medium shot, close up, extreme close up, two shot, multiple people, outdoor and indoor, and so on). For more information, see [Editorial shot type detection](https://learn.microsoft.com/en-us/azure/azure-video-indexer/scenes-shots-keyframes#editorial-shot-type-detection).
* **Observed people tracking** (preview): Detects observed people in videos and provides information such as the location of the person in the video frame (using bounding boxes) and the exact timestamp (start, end) and confidence when a person appears. For more information, see [Trace observed people in a video](https://learn.microsoft.com/en-us/azure/azure-video-indexer/observed-people-tracking).
  + **People's detected clothing** (preview): Detects the clothing types of people appearing in the video and provides information such as long or short sleeves, long or short pants and skirt or dress. The detected clothing is associated with the people wearing it and the exact timestamp (start, end) along with a confidence level for the detection are provided. For more information, see [detected clothing](https://learn.microsoft.com/en-us/azure/azure-video-indexer/detected-clothing).
  + **Featured clothing** (preview): Captures featured clothing images appearing in a video. You can improve your targeted ads by using the featured clothing insight. For information on how the featured clothing images are ranked and how to get the insights, see [featured clothing](https://learn.microsoft.com/en-us/azure/azure-video-indexer/observed-people-featured-clothing).
* **Matched person** (preview): Matches people that were observed in the video with the corresponding faces detected. The matching between the observed people and the faces contain a confidence level.
* **Object detection** Detects unique objects that are also tracked so that if they return to the frame they are recognized. See [Azure AI Video Indexer object detection](https://learn.microsoft.com/en-us/azure/azure-video-indexer/object-detection)
* **Slate detection** (preview): Identifies the following movie post-production insights when indexing a video using the advanced indexing option:
  + Clapperboard detection with metadata extraction.
  + Digital patterns detection, including color bars.
  + Textless slate detection, including scene matching.

For details, see [Slate detection](https://learn.microsoft.com/en-us/azure/azure-video-indexer/slate-detection-insight).

* **Textual logo detection** (preview): Matches a specific predefined text using Azure AI Video Indexer OCR. For example, if a user created a textual logo: "Microsoft", different appearances of the word Microsoft will be detected as the "Microsoft" logo. For more information, see [Detect textual logo](https://learn.microsoft.com/en-us/azure/azure-video-indexer/detect-textual-logo).

### Audio models

* **Audio transcription**: Converts speech to text over 50 languages and allows extensions. For more information, see [Azure AI Video Indexer language support](https://learn.microsoft.com/en-us/azure/azure-video-indexer/language-support).
* **Automatic language detection**: Identifies the dominant spoken language. For more information, see [Azure AI Video Indexer language support](https://learn.microsoft.com/en-us/azure/azure-video-indexer/language-support). If the language can't be identified with confidence, Azure AI Video Indexer assumes the spoken language is English. For more information, see [Language identification model](https://learn.microsoft.com/en-us/azure/azure-video-indexer/language-identification-model).
* **Multi-language speech identification and transcription**: Identifies the spoken language in different segments from audio. It sends each segment of the media file to be transcribed and then combines the transcription back to one unified transcription. For more information, see [Automatically identify and transcribe multi-language content](https://learn.microsoft.com/en-us/azure/azure-video-indexer/multi-language-identification-transcription).
* **Closed captioning**: Creates closed captioning in three formats: VTT, TTML, SRT.
* **Two channel processing**: Auto detects separate transcript and merges to single timeline.
* **Noise reduction**: Clears up telephony audio or noisy recordings (based on Skype filters).
* **Transcript customization** (CRIS): Trains custom speech to text models to create industry-specific transcripts. For more information, see [Customize a Language model from the Azure AI Video Indexer website](https://learn.microsoft.com/en-us/azure/azure-video-indexer/customize-language-model-with-website) and [Customize a Language model with the Azure AI Video Indexer APIs](https://learn.microsoft.com/en-us/azure/azure-video-indexer/customize-language-model-with-api).
* **Speaker enumeration**: Maps and understands which speaker spoke which words and when. Sixteen speakers can be detected in a single audio-file.
* **Speaker statistics**: Provides statistics for speakers' speech ratios.
* **Textual content moderation**: Detects explicit text in the audio transcript.
* **Text-based emotion detection**: Emotions such as joy, sadness, anger, and fear that were detected via transcript analysis.
* **Translation**: Creates translations of the audio transcript to many different languages. For more information, see [Azure AI Video Indexer language support](https://learn.microsoft.com/en-us/azure/azure-video-indexer/language-support).
* **Audio effects detection** (preview): Detects the following audio effects in the non-speech segments of the content: alarm or siren, dog barking, crowd reactions (cheering, clapping, and booing), gunshot or explosion, laughter, breaking glass, and silence.

The detected acoustic events are in the closed captions file. The file can be downloaded from the Azure AI Video Indexer website. For more information, see [Audio effects detection](https://learn.microsoft.com/en-us/azure/azure-video-indexer/audio-effects-detection).

**Note**

The full set of events is available only when you choose **Advanced Audio Analysis** when uploading a file, in upload preset. By default, only silence is detected.

### Audio and video models (multi-channels)

When indexing by one channel, partial results for those models are available.

* **Keywords extraction**: Extracts keywords from speech and visual text.
* **Named entities extraction**: Extracts brands, locations, and people from speech and visual text via natural language processing (NLP).
* **Topic inference**: Extracts topics based on various keywords (that is, keywords 'Stock Exchange', 'Wall Street' produces the topic 'Economics'). The model uses three different ontologies ([IPTC](https://iptc.org/standards/media-topics/), [Wikipedia](https://www.wikipedia.org/) and the Video Indexer hierarchical topic ontology). The model uses transcription (spoken words), OCR content (visual text), and celebrities recognized in the video using the Video Indexer facial recognition model.
* **Artifacts**: Extracts rich set of "next level of details" artifacts for each of the models.
* **Sentiment analysis**: Identifies positive, negative, and neutral sentiments from speech and visual text.

Azure Video Analyzer capabilities include:

* **Deep search—**improves the search experience of a video library using insights gained from videos
* **Content creation**—develop trailers, social media content, highlight reels, or news clips using the insights Video Analyzer Media extracts from your content
* **Monetization**—insights can improve a video’s value by driving content and advertising optimization
* **Recommendations**—insights from the video may be used to enhance user engagement by featuring relevant or compelling video moments

Practical Reference Link: <https://learn.microsoft.com/en-us/training/modules/analyze-video/>

Azure Video Indexer is a service to extract insights from video, including face identification, text recognition, object labels, scene segmentations, and more.

The **Azure Video Indexer** service is designed to help you extract information from videos. It provides functionality that you can use for:

* Facial recognition - detecting the presence of individual people in the image. This requires [Limited Access](https://aka.ms/cog-services-limited-access) approval.
* Optical character recognition - reading text in the video.
* Speech transcription - creating a text transcript of spoken dialog in the video.
* Topics - identification of key topics discussed in the video.
* Sentiment - analysis of how positive or negative segments within the video are.
* Labels - label tags that identify key objects or themes throughout the video.
* Content moderation - detection of adult or violent themes in the video.
* Scene segmentation - a breakdown of the video into its constituent scenes.

<https://www.videoindexer.ai/account/login> 🡪Video indexer to be connected with Azure AI Video Indexer

## Deploy with ARM template

Azure Resource Manager (ARM) templates are available to create the Azure AI Video Indexer resource in your subscription, based on the parameters specified in the template file.

Practical:

1. Start Visual Studio Code.
2. Open the palette (SHIFT+CTRL+P) and run a **Git: Clone** command to clone the https://github.com/MicrosoftLearning/mslearn-ai-vision repository to a local folder (it doesn't matter which folder).
3. In your browser, open the Video Indexer portal at https://www.videoindexer.ai.
4. If you have an existing Video Indexer account, sign in. Otherwise, sign up for a free account and sign in using your Microsoft account (or any other valid account type). If you have difficulty signing in, try opening a private browser session.
5. In a new tab, download the Responsible AI video by visiting https://aka.ms/responsible-ai-video. Save the file.
6. In Video Indexer, select the **Upload** option. Then select the option to **Browse for files**, select the downloaded video, and click **Add**. Change the default name to **Responsible AI**, review the default settings, select the checkbox to verify compliance with Microsoft's policies for facial recognition, and upload the file.
7. After the file has uploaded, wait a few minutes while Video Indexer automatically indexes it.

**Note**: In this exercise, we're using this video to explore Video Indexer functionality; but you should take the time to watch it in full when you've finished the exercise as it contains useful information and guidance for developing AI-enabled applications responsibly!

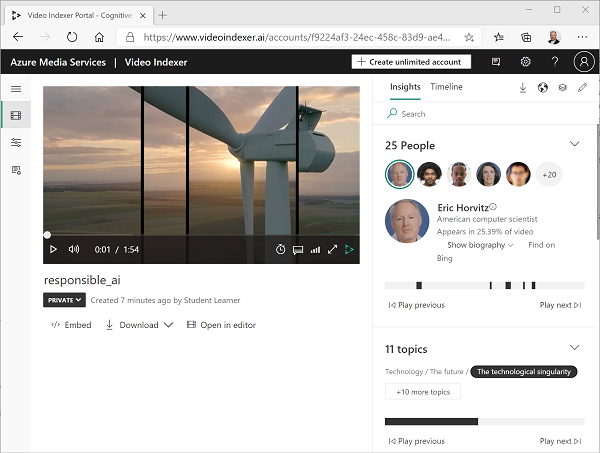
## **A screenshot of a computer Description automatically generated**

## **Review video insights**

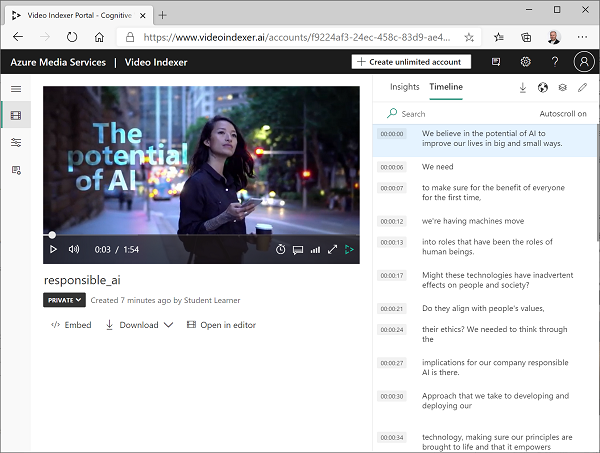
The indexing process extracts insights from the video, which you can view in the portal.

1. In the Video Indexer portal, when the video is indexed, select it to view it. You'll see the video player alongside a pane that shows insights extracted from the video.

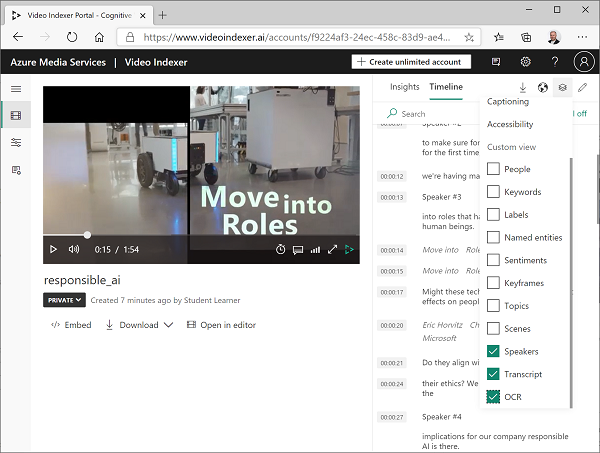
**Note**: Due to the limited access policy to protect individuals identities, you may not see names when you index the video.



1. As the video plays, select the **Timeline** tab to view a transcript of the video audio.



1. At the top right of the portal, select the **View** symbol (which looks similar to 🗇), and in the list of insights, in addition to **Transcript**, select **OCR** and **Speakers**.



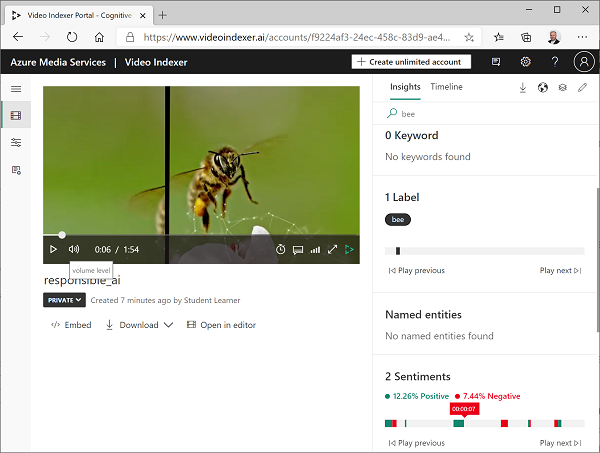
1. Observe that the **Timeline** pane now includes:
   * Transcript of audio narration.
   * Text visible in the video.
   * Indications of speakers who appear in the video. Some well-known people are automatically recognized by name, others are indicated by number (for example Speaker #1).
2. Switch back to the **Insights** pane and view the insights show there. They include:
   * Individual people who appear in the video.
   * Topics discussed in the video.
   * Labels for objects that appear in the video.
   * Named entities, such as people and brands that appear in the video.
   * Key scenes.
3. With the **Insights** pane visible, select the **View** symbol again, and in the list of insights, add **Keywords** and **Sentiments** to the pane.

The insights found can help you determine the main themes in the video. For example, the **topics** for this video show that it is clearly about technology, social responsibility, and ethics.

## **Search for insights**

You can use Video Indexer to search the video for insights.

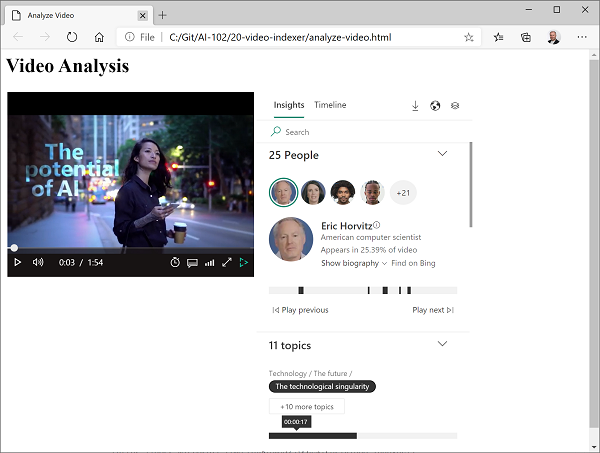
1. In the **Insights** pane, in the **Search** box, enter Bee. You may need to scroll down in the Insights pane to see results for all types of insight.
2. Observe that one matching label is found, with its location in the video indicated beneath.
3. Select the beginning of the section where the presence of a bee is indicated, and view the video at that point (you may need to pause the video and select carefully - the bee only appears briefly!)
4. Clear the **Search** box to show all insights for the video.



## **Use Video Indexer widgets**

The Video Indexer portal is a useful interface to manage video indexing projects. However, there may be occasions when you want to make the video and its insights available to people who don't have access to your Video Indexer account. Video Indexer provides widgets that you can embed in a web page for this purpose.

1. In Visual Studio Code, in the **06-video-indexer** folder, open **analyze-video.html**. This is a basic HTML page to which you will add the Video Indexer **Player** and **Insights** widgets. Note the reference to the **vb.widgets.mediator.js** script in the header - this script enables multiple Video Indexer widgets on the page to interact with one another.
2. In the Video Indexer portal, return to the **Media files** page and open your **Responsible AI** video.
3. Under the video player, select **</> Embed** to view the HTML iframe code to embed the widgets.
4. In the **Share and Embed** dialog box, select the **Player** widget, set the video size to 560 x 315, and then copy the embed code to the clipboard.
5. In Visual Studio Code, in the **analyze-video.html** file, paste the copied code under the comment **<-- Player widget goes here -- >**.
6. Back in the **Share and Embed** dialog box, select the **Insights** widget and then copy the embed code to the clipboard. Then close the **Share and Embed** dialog box, switch back to Visual Studio Code, and paste the copied code under the comment **<-- Insights widget goes here -- >**.
7. Save the file. Then in the **Explorer** pane, right-click **analyze-video.html** and select **Reveal in File Explorer**.
8. In File Explorer, open **analyze-video.html** in your browser to see the web page.
9. Experiment with the widgets, using the **Insights** widget to search for insights and jump to them in the video.



## **Use the Video Indexer REST API**

Video Indexer provides a REST API that you can use to upload and manage videos in your account.

### **Get your API details**

To use the Video Indexer API, you need some information to authenticate requests:

1. In the Video Indexer portal, expand the left pane and select the **Account settings** page.
2. Note the **Account ID** on this page - you will need it later.
3. Open a new browser tab and go to the Video Indexer developer portal at https://api-portal.videoindexer.ai, signing in using the credentials for your Video Indexer account.
4. On the **Profile** page, view the **Subscriptions** associated with your profile.
5. On the page with your subscription(s), observe that you have been assigned two keys (primary and secondary) for each subscription. Then select **Show** for any of the keys to see it. You will need this key shortly.

### **Use the REST API**

Now that you have the account ID and an API key, you can use the REST API to work with videos in your account. In this procedure, you'll use a PowerShell script to make REST calls; but the same principles apply with HTTP utilities such as cURL or Postman, or any programming language capable of sending and receiving JSON over HTTP.

All interactions with the Video Indexer REST API follow the same pattern:

* An initial request to the **AccessToken** method with the API key in the header is used to obtain an access token.
* Subsequent requests use the access token to authenticate when calling REST methods to work with videos.

1. In Visual Studio Code, in the **06-video-indexer** folder, open **get-videos.ps1**.
2. In the PowerShell script, replace the **YOUR\_ACCOUNT\_ID** and **YOUR\_API\_KEY** placeholders with the account ID and API key values you identified previously.
3. Observe that the location for a free account is "trial". If you have created an unrestricted Video Indexer account (with an associated Azure resource), you can change this to the location where your Azure resource is provisioned (for example "eastus").
4. Review the code in the script, noting that invokes two REST methods: one to get an access token, and another to list the videos in your account.
5. Save your changes, and then at the top-right of the script pane, use the **▷** button to run the script.
6. View the JSON response from the REST service, which should contain details of the **Responsible AI** video you indexed previously.

**Practical work:**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Login to <https://api-portal.videoindexer.ai>

A screenshot of a computer

Description automatically generated

Try it yourself:

## **Use Video Indexer widgets**

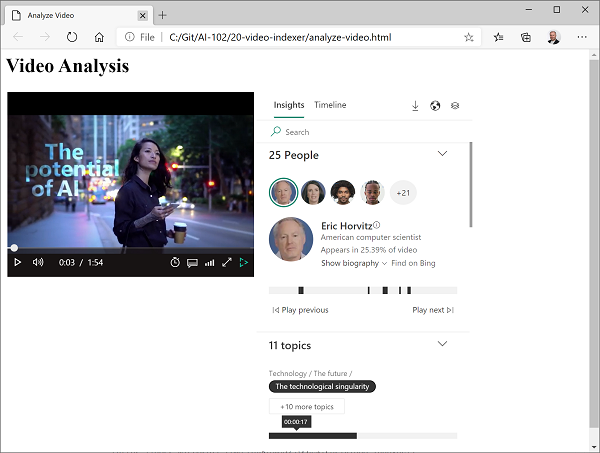
The Video Indexer portal is a useful interface to manage video indexing projects. However, there may be occasions when you want to make the video and its insights available to people who don't have access to your Video Indexer account. Video Indexer provides widgets that you can embed in a web page for this purpose.

1. In Visual Studio Code, in the **06-video-indexer** folder, open **analyze-video.html**. This is a basic HTML page to which you will add the Video Indexer **Player** and **Insights** widgets. Note the reference to the **vb.widgets.mediator.js** script in the header - this script enables multiple Video Indexer widgets on the page to interact with one another.
2. In the Video Indexer portal, return to the **Media files** page and open your **Responsible AI** video.
3. Under the video player, select **</> Embed** to view the HTML iframe code to embed the widgets.
4. In the **Share and Embed** dialog box, select the **Player** widget, set the video size to 560 x 315, and then copy the embed code to the clipboard.
5. In Visual Studio Code, in the **analyze-video.html** file, paste the copied code under the comment **<-- Player widget goes here -- >**.
6. Back in the **Share and Embed** dialog box, select the **Insights** widget and then copy the embed code to the clipboard. Then close the **Share and Embed** dialog box, switch back to Visual Studio Code, and paste the copied code under the comment **<-- Insights widget goes here -- >**.
7. Save the file. Then in the **Explorer** pane, right-click **analyze-video.html** and select **Reveal in File Explorer**.
8. In File Explorer, open **analyze-video.html** in your browser to see the web page.
9. Experiment with the widgets, using the **Insights** widget to search for insights and jump to them in the video.

***Thank you***

A screenshot of a computer

Description automatically generated



Try yourself:

### **Use the REST API**

Now that you have the account ID and an API key, you can use the REST API to work with videos in your account. In this procedure, you'll use a PowerShell script to make REST calls; but the same principles apply with HTTP utilities such as cURL or Postman, or any programming language capable of sending and receiving JSON over HTTP.

All interactions with the Video Indexer REST API follow the same pattern:

* An initial request to the **AccessToken** method with the API key in the header is used to obtain an access token.
* Subsequent requests use the access token to authenticate when calling REST methods to work with videos.

1. In Visual Studio Code, in the **06-video-indexer** folder, open **get-videos.ps1**.
2. In the PowerShell script, replace the **YOUR\_ACCOUNT\_ID** and **YOUR\_API\_KEY** placeholders with the account ID and API key values you identified previously.
3. Observe that the location for a free account is "trial". If you have created an unrestricted Video Indexer account (with an associated Azure resource), you can change this to the location where your Azure resource is provisioned (for example "eastus").
4. Review the code in the script, noting that invokes two REST methods: one to get an access token, and another to list the videos in your account.
5. Save your changes, and then at the top-right of the script pane, use the **▷** button to run the script.
6. View the JSON response from the REST service, which should contain details of the **Responsible AI** video you indexed previously.