Explainer: MP4 Muxer for MediaRecorder.

Summary

Support MP4 container to the MediaRecorder API.

1. **MediaRecorder API Overview**:
    - The **MediaRecorder API** is a powerful tool that allows web developers to capture media streams (such as video and audio) directly from the user's device.
    - It encodes the media stream and writes it to a **blob**, which is essentially a binary data container.
2. **Encoding and Container Format**:
3. When using the **MediaRecorder**, developers can specify the encoding format they want for the recorded media. This is done by passing relevant options during the API constructor.
    - Here is an example.

```
navigator.mediaDevices
  .getUserMedia({video: true, audio: true})
  .then ((stream) => {
   const recorder = new MediaRecorder(stream,
{mimeType:"video/webm;codecs=vp9,opus"});
   recorder.start();
});
```

    - In the above example, it is capturing both video and audio streams using `getUserMedia`. The `MediaRecorder` instance is created with a **mimeType** of `"video/webm;codecs=vp8,opus that record stream, which is captured by getUserMedai API.`
    - **WebM** container format is used, which supports multiple Codecs: **VP9** for video and **Opus** for audio.
4. **Chromium and Safari Support**:

    Chromium (the open-source project be

```
navigator.mediaDevices
      .getUserMedia({video: true, audio: true})
```

```
      .then ((stream) => {
  const recorder = new MediaRecorder(stream,
{mimeType:"video/webm;codecs=vp9,opus"});
  recorder.start();
});
```

    -
    - hind Google Chrome) currently supports **WebM** containers with codecs like **VP8**, **VP9**, **H.264 (also known as avc1)**, **Opus**, and **PCM**.
    - On the other hand, **Safari** only supports the **MP4** container format.
5. **Project Objective**:
    - This project aims to enhance Chromium's capabilities by adding support for the **MP4** container format.
    - Specifically, a user wants to enable **H.264 (avc1)** video codec and **AAC (mp4a.40.2)** audio codec within the **MP4** container.
    - Since both H.264 and AAC codecs are already supported in Chromium, the project involves configuring the `Media-Recorder` to use these codecs and package the resulting media into an **MP4** container.
    - This would also bring more compatibility among **MediaRecorder** and **MediaSource**. Currently they can only work together in the direction of **MediaRecorder -> MediaSource** with **WebM** container and limited to **VP8**, **VP9** and **Opus** codecs. No **AAC (mp4a.40.2) support..**
    - If the OS does not have proprietary codecs such as H.264 and AAC, users can use VP9 and Opus for creating MP4 as well.

Here's how you'd achieve this in your code:

```
const recorder = new MediaRecorder(stream, { mimeType:
"video/mp4;codecs=avc1,mp4a.40.2" });

const recorder = new MediaRecorder(stream, { mimeType:
"video/mp4;codecs=vp9,Opus" });
```

How to Detect a Feature ("video/mp4;codecs=avc1,mp4a.40.2").

Developers can see if the feature is enabled by using the API of MediaRecorder.IsTypeSupported (MediaRecorder: isTypeSupported() static method - Web APIs : MDN)

```
if (MediaRecorder.isTypeSupported("video/mp4;codecs=avc1,mp4a.40.2")) {
  console.log(`mp4 mimetype for mediaRecorder is supported` ).
}
```

Developer engagement

Partners were notified with this new feature as it was a pain point for them because crbug.com/45811744