

MediaStreamTrack.ended

“What happened?”

Reading prep for this presentation

JS Arrow functions are used for briefer examples:

```
track.onended(function(event) { });
```



```
track.onended(event => { });
```

They're still callback functions.

Reading prep for this presentation

Promise - unified primitive for asynchronicity.

```
foo(result => log("foo's callback called w/" + result), failure);
```



```
foo().then(result => log("foo fulfilled w/" + result)).catch(failure);
```

Powerful standardized pattern/error-handling/prose

<http://stackoverflow.com/questions/22539815/arent-promises-just-callbacks>

Standard prose & good guide: <https://w3ctag.github.io/promises-guide>

WebIDL-supported: <http://heycam.github.io/webidl/#idl-promise>

Spec: <http://people.mozilla.org/~jorendorff/es6-draft.html>

- 1. Problem**
- 2. What's proposed**

Happy ending?

```
pc.onaddstream = event => {  
  var track = event.stream.getVideoTracks()[0];  
  track.onended = e => log("Hangup or dropped call");  
}
```

which is it?

```
navigator.mediaDevices.getUserMedia({ video: true })  
  .then(stream => {  
    var track = stream.getVideoTracks()[0];  
    track.onended = e => log("Stop, permission rescind or driver issue");  
  })  
  .catch(failure);
```

Seems important which one it is!

What's proposed - 'ended' promise

`promise.then(success,
failure)`

```
pc.onaddstream = event => {  
  var track = event.stream.getVideoTracks()[0];  
  track.ended.then(() => log("Hangup"),  
    reason => log("Dropped call: " + reason.message));  
}
```

<http://w3ctag.github.io/promises-guide/#one-time-events>

<https://github.com/w3c/mediacapture-main/pull/40>

What's proposed - 'ended' promise

```
nav.mediaDevices.getUserMedia({ video: true })  
  .then(stream => {  
    var track = stream.getVideoTracks()[0];  
    track.ended.catch(reason => { // Success = we stopped it. Don't care  
      switch (reason.name) {  
        case "PermissionDeniedError": log("Permission rescind"); break;  
        case "SourceUnavailableError": log("Driver issue: " +  
                                           reason.message); break;  
        default: log("Something else: " + reason.message); break;  
      }  
    });  
  }).catch(failure);
```

promise.catch(failure)

PermissionDeniedError
SourceUnavailableError
AbortError

<https://github.com/w3c/mediacapture-main/pull/40>

The.end

Before:

```
track.onended = () => log("Ended or dropped");
```

After:

```
track.ended.then(() => log("Ended"),  
  reason => log("Dropped: " + reason.message));
```