

# Användardata

*Cookies, LocalStorage, Geolocation och mediahantering*

# Agenda

- *Lagring*
  - *Cookies*
  - *LocalStorage*
- *Platsinformation*
  - *Geolocation*
- *Mediahantering*
  - *Bild, ljud och video*

# Användardata

- *När finns det anledning att spara data om användaren?*

# Exempel: Användardata

- *Inloggning*
- *Kundvagn med varor*
- *Inställningar (språk, valuta, m.m.)*
- *Analysera beteende*
- *Spara tidigare hämtad data (prestanda)*
- *Med mera*

# Användardata

- *Hur görs detta idag?*

# Användardata

- *Hur görs detta idag?*
  - *Cookies*
  - *LocalStorage*
  - *Geolocation*

# Cookies



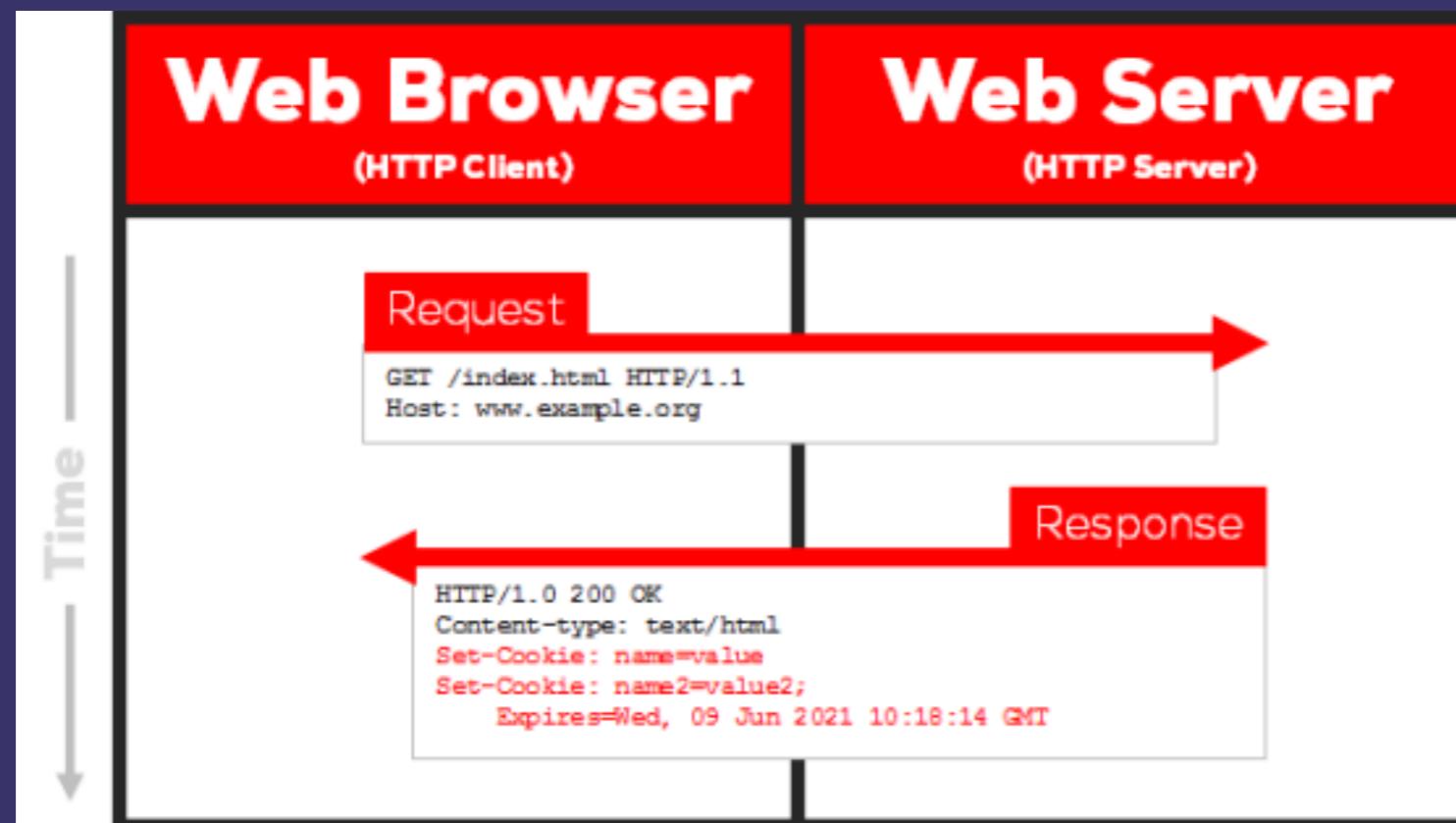
# Cookie

- *Skickas med i HTTP-anropet till servern*
- *Kan användas för delar av eller för en hel domän*
- *Kan kräva en krypterad anslutning (HTTPS)*
- *Kan kräva åtkomst via HTTP*
- *Bestämt utgångsdatum*

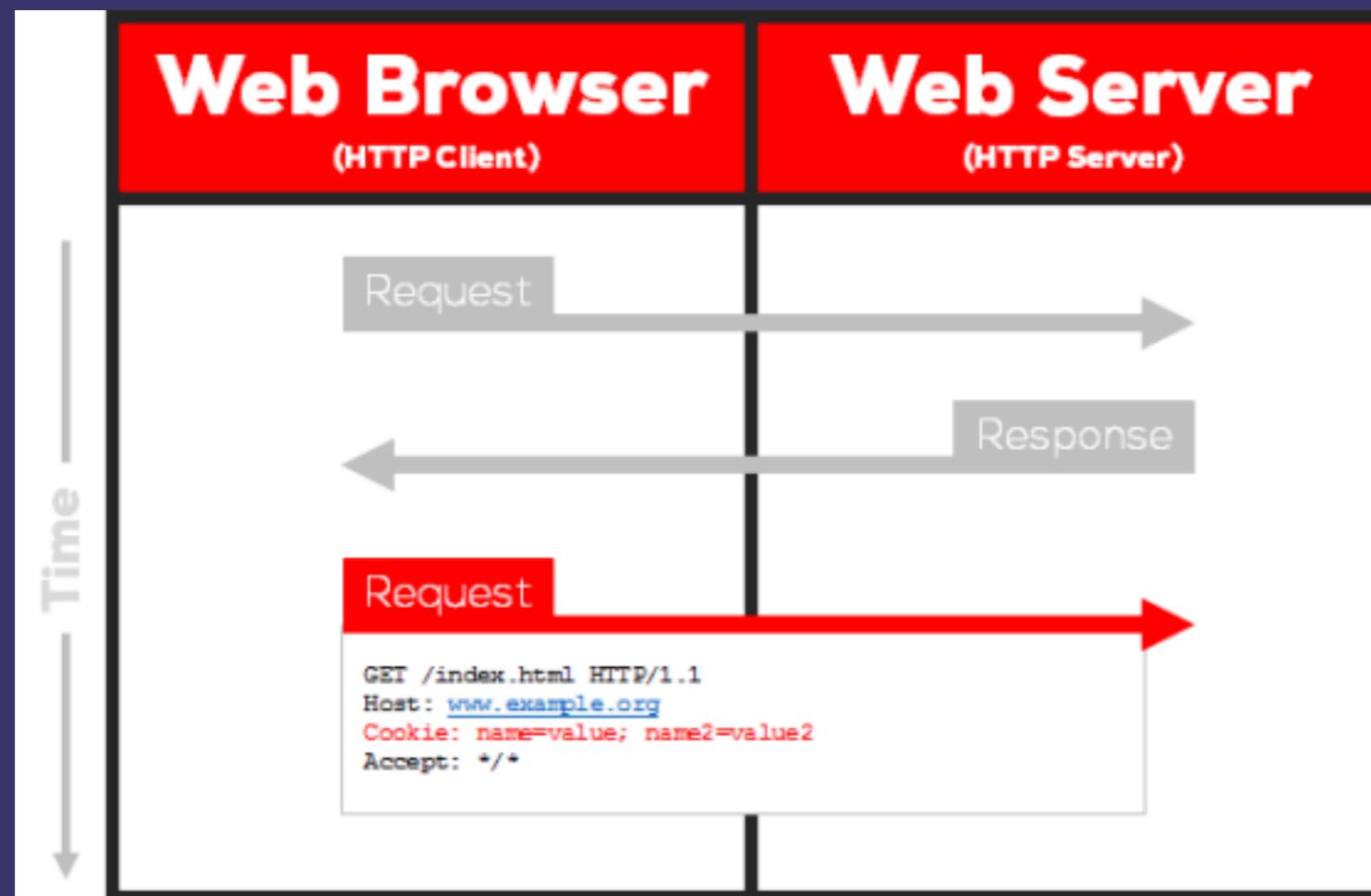
# Cookie

- *Begränsad storlek (4kb)*
- *Skickas med vid varje anrop till servern*
- *Sparas som en sträng (text)*
- *Dålig integration med JavaScript*

# Cookie



# Cookie



# Hur?

```
1
2 // Hämta sparad data
3 //   - Sparas som nyckel:värde-par
4 console.log(document.cookie);
5
6 // Spara data
7 document.cookie = "username=Sebastian";
8 document.cookie = "role=admin";
9
10 // Verifiera
11 console.log(document.cookie);
12 // => "username=Sebastian; role=admin"
13
```

# Hur?

```
1  
2 document.cookie = "username=Sebastian; expires=Thu, 18 Dec 2020 12:00:00 UTC; path=/";  
3
```

- *Attribut:*
  - expires=<datum>; (UTC)
  - path=/;
  - secure;
  - httpOnly;
  - sameSite=strict;

# Att tänka på

- *Cookies kräver en webbserver*
- *EU-direktiv om upplysning*
  - <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32009L0136>
- *Säkerhetsaspekter*
  - *CSRF, XSS, Session hijacking*
  - [https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies#Secure\\_and\\_HttpOnly\\_cookies](https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies#Secure_and_HttpOnly_cookies)

# Bibliotek

<https://github.com/js-cookie/js-cookie>

The screenshot shows the GitHub repository page for `js-cookie / js-cookie`. The page includes a search bar, navigation links for Pull requests, Issues, Marketplace, and Explore, and a header with metrics: 98.1k used by, 358 watchers, 15.4k stars, 1.9k forks, and a blue 'Clone' button. Below the header are tabs for Code, Issues (1), Pull requests (0), Actions, Projects (0), Wiki, Security (0), and Insights. A description states: "A simple, lightweight JavaScript API for handling browser cookies". The repository has 745 commits, 4 branches, 0 packages, 29 releases, 47 contributors, and an MIT license. A pull request from `carhartl` is shown, along with a list of recent commits:

Commit	Description	Date
<code>.github/ISSUE_TEMPLATE</code>	Add predefined labels to issue templates	7 months ago
<code>examples</code>	Update references for release candidate	2 months ago
<code>src</code>	Rename variables for clarity	2 months ago
<code>test</code>	Tidy up converters	2 months ago
<code>.eslintignore</code>	Add examples, starting with webpack	7 months ago
<code>.eslintrc.json</code>	Ensure consistent formatting: html, json, md files	8 months ago
<code>.gitignore</code>	Set up package for module/nomodule distributions	8 months ago
<code>.prettierignore</code>	Configure prettier to ignore dist	7 months ago
<code>.release-it.json</code>	Fix conditional in release hook	5 months ago

# Demo

# LocalStorage

# LocalStorage

- *Kopplat till webbläsaren (inbyggt API)*
- *Inget utgångsdatum*
- *Cirka 5 MB data*
- *Sparas som nyckel:värde-par*
- *Per domän*

# LocalStorage

- *Osäkert (spara aldrig känslig data i LocalStorage)*
- *Svårare att radera datan (utan utgångsdatum)*
- *Sparas som en sträng (text)*

# Hur?

```
1
2 // Spara data
3 localStorage.setItem("username", "Sebastian");
4
5 // Hämta data
6 localStorage.getItem("username");
7
8 // Radera data
9 localStorage.removeItem("username");
10
11 // Töm allt data
12 localStorage.clear();
13
```

# Hur?

```
1
2 // Spara data som JSON
3 let user = {
4     username: "Sebastian"
5 };
6
7 localStorage.setItem("user", JSON.stringify(user));
8
9 // Hämta data i JSON
10 let user = JSON.parse(localStorage.getItem("user"));
11
```

# Demo

# Geolocation

# Geolocation

- *När utnyttjas en användares platsdata?*

# Geolocation

- *JavaScript ger oss möjlighet att hämta enhetens position*
- *Kräver användarens godkännande*
- *Kräver en säker anslutning (HTTPS)*

# Geolocation

- *Tre funktioner:*
  - navigator.geolocation.getCurrentPosition
  - navigator.geolocation.watchCurrentPosition
  - navigator.geolocation.clearWatch
- *Båda tar emot följande parametrar:*
  - *Funktion som anropas när vi lyckas hämta positionen*
  - *Funktion som anropas när vi misslyckas hämta positionen*
  - *Inställningar (cache-tid, timeout och precision)*
- *Tänk på att allting sker asynkront*

# Kodexempel

```
1
2 // Anropas när vi lyckas hämta en användares position
3 function success(position) {
4     console.log("Received your position: ", position);
5 }
6
7 // Anropas när vi inte lyckas hämta en användares position
8 function fail(error) {
9     console.log("Unable to fetch your position");
10}
11
12 // Kontrollera att vi har stöd för funktionerna
13 if (navigator.geolocation) {
14     navigator.geolocation.getCurrentPosition(
15         success,
16         fail,
17         { enableHighAccuracy: true, timeout: 10000 }
18     );
19 }
20
```

# GeolocationPosition

```
▼ GeolocationPosition
  ▼ coords: GeolocationCoordinates
    accuracy: 65
    altitude: 8.0011625289917
    altitudeAccuracy: 10
    heading: null
    latitude: 55.58978927450167
    longitude: 13.010123732058908
    speed: null
    ▶ <prototype>: GeolocationCoordinatesPrototype { latitude: Getter, longitude: Getter, altitude: Getter, ... }
    timestamp: 1587971122205
  ▶ <prototype>: GeolocationPositionPrototype { coords: Getter, timestamp: Getter, ... }
```

Property	Returns
coords.latitude	The latitude as a decimal number (always returned)
coords.longitude	The longitude as a decimal number (always returned)
coords.accuracy	The accuracy of position (always returned)
coords.altitude	The altitude in meters above the mean sea level (returned if available)
coords.altitudeAccuracy	The altitude accuracy of position (returned if available)
coords.heading	The heading as degrees clockwise from North (returned if available)
coords.speed	The speed in meters per second (returned if available)
timestamp	The date/time of the response (returned if available)

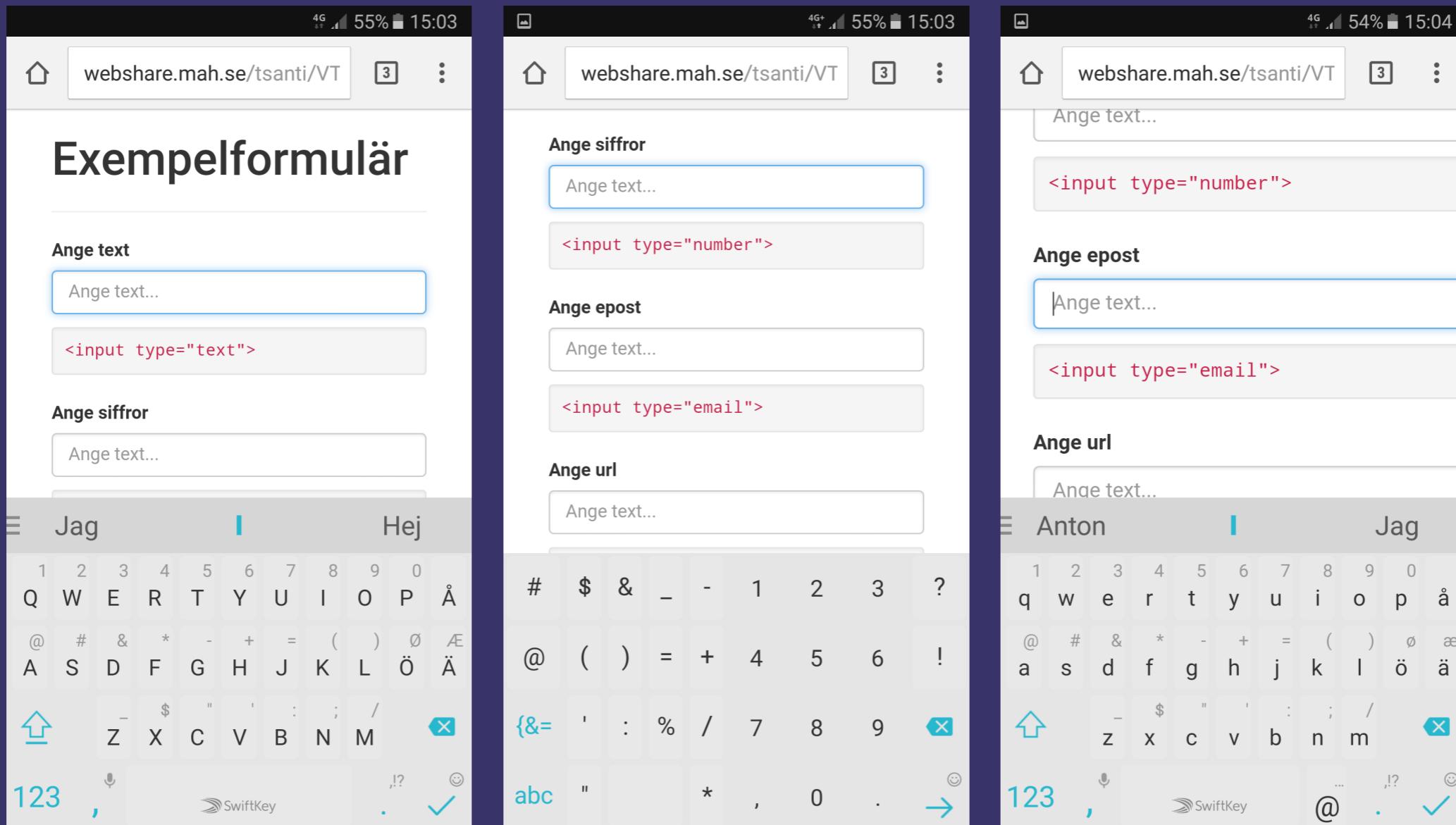
# Demo

# Mediahantering

- *Ljud, bild och video*

# Input

- Det finns olika typer av formulärsfältet **input**
  - *text, date, password, email, number, etc.*



# Filuppladdning

## Syntax

```
<input accept="file_extension|audio/*|video/*|image/*|media_type">
```

**Tip:** To specify more than one value, separate the values with a comma (e.g. <input accept="audio/\*,video/\*,image/\*" />).

## Attribute Values

Value	Description
<i>file_extension</i>	A file extension starting with the STOP character, e.g: .gif, .jpg, .png, .doc
audio/*	All sound files are accepted
video/*	All video files are accepted
image/*	All image files are accepted
<i>media_type</i>	A valid media type, with no parameters. Look at <a href="#">IANA Media Types</a> for a complete list of standard media types

## Example

Specify that the server accepts only image files in the file upload:

```
<form action="/action_page.php">
  <input type="file" name="pic" accept="image/*">
  <input type="submit">
</form>
```

# Filuppladdning

- *Öppna kamera eller ljudinspelare automatiskt med "capture"*

- To take a picture using the device's local still image capture device, such as a camera, and upload the picture taken using an HTML form:

## EXAMPLE 1

```
<form action="server.cgi" method="post" enctype="multipart/form-data">
  <input type="file" name="image" accept="image/*" capture>
  <input type="submit" value="Upload">
</form>
```

- Or alternatively, to capture video using the device's local video camera:

## EXAMPLE 2

```
<form action="server.cgi" method="post" enctype="multipart/form-data">
  <input type="file" name="video" accept="video/*" capture>
  <input type="submit" value="Upload">
</form>
```

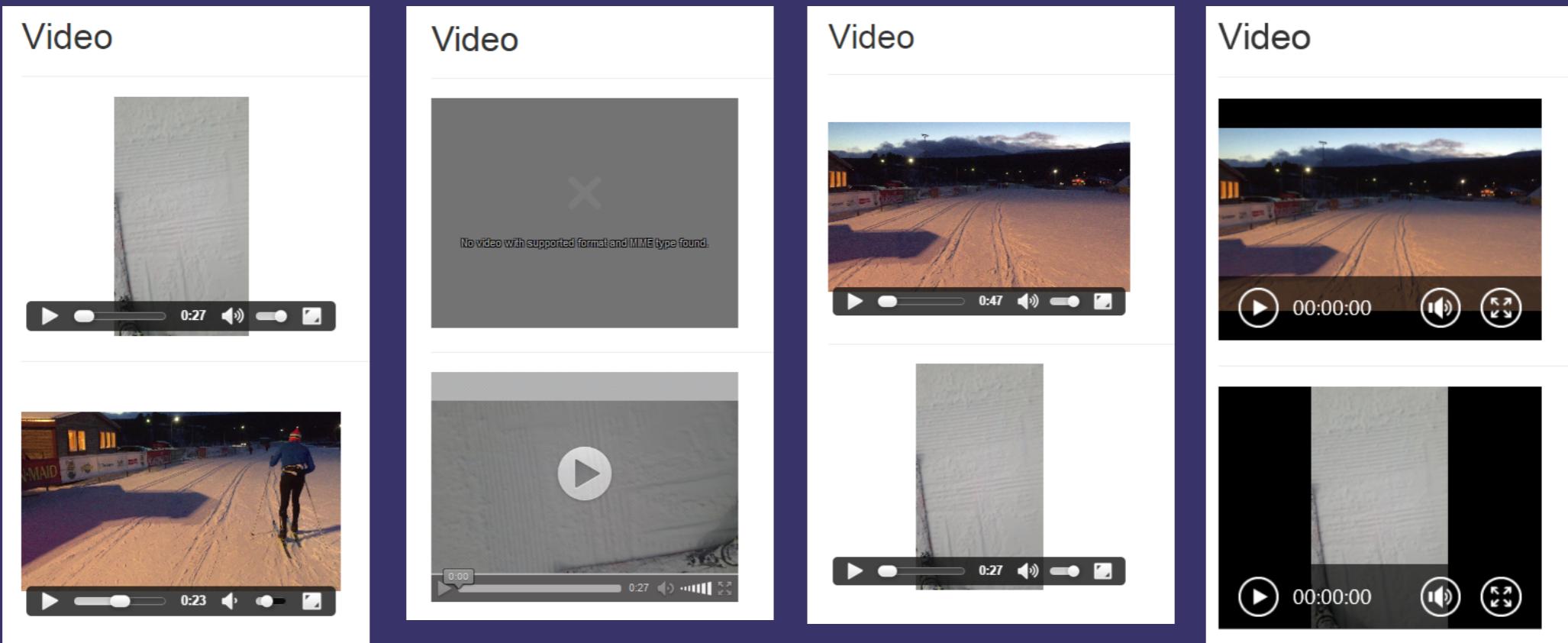
- Or alternatively, to capture audio using the device's local microphone:

## EXAMPLE 3

```
<form action="server.cgi" method="post" enctype="multipart/form-data">
  <input type="file" name="audio" accept="audio/*" capture>
  <input type="submit" value="Upload">
</form>
```

# Video

```
1 <video width="320" height="240" controls>
2   <source src="movie.mp4" type="video/mp4">
3   <source src="movie.ogv" type="video/ogg">
4   Your browser does not support the video tag.
5 </video>
6
7
```



# Video

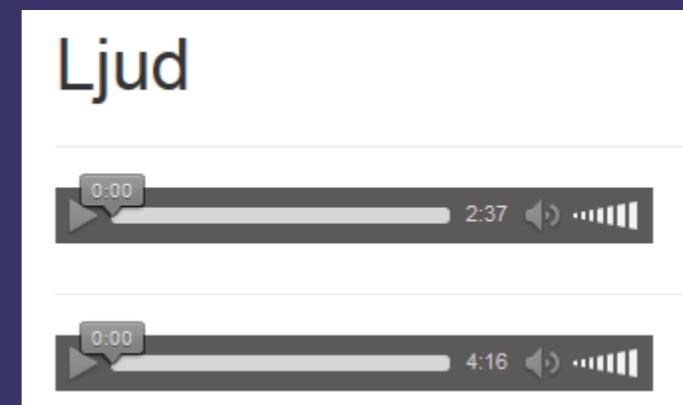
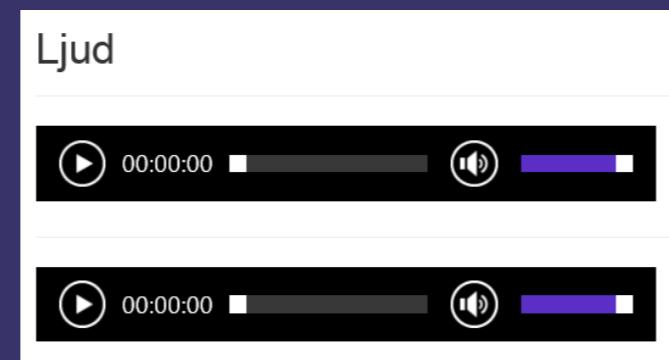
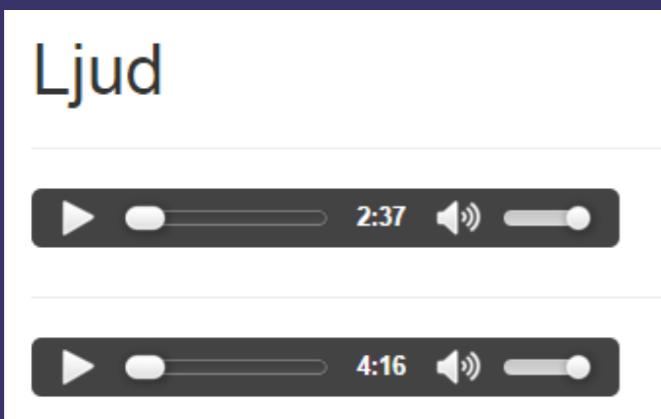
## Optional Attributes

Attribute	Value	Description
<u>autoplay</u>	autoplay	Specifies that the video will start playing as soon as it is ready
<u>controls</u>	controls	Specifies that video controls should be displayed (such as a play/pause button etc).
<u>height</u>	<i>pixels</i>	Sets the height of the video player
<u>loop</u>	loop	Specifies that the video will start over again, every time it is finished
<u>muted</u>	muted	Specifies that the audio output of the video should be muted
<u>poster</u>	<i>URL</i>	Specifies an image to be shown while the video is downloading, or until the user hits the play button
<u>preload</u>	auto metadata none	Specifies if and how the author thinks the video should be loaded when the page loads
<u>src</u>	<i>URL</i>	Specifies the URL of the video file
<u>width</u>	<i>pixels</i>	Sets the width of the video player

Browser	MP4	WebM	Ogg
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES from Firefox 21 from Firefox 30 for Linux	YES	YES
Safari	YES	NO	NO
Opera	YES From Opera 25	YES	YES

# Ljud

```
1 <audio controls>
2   <source src="horse.ogg" type="audio/ogg">
3   <source src="horse.mp3" type="audio/mpeg">
4   Your browser does not support the audio tag.
5 </audio>
6
7
```



# Ljud

## Attributes

Attribute	Value	Description
<u>autoplay</u>	autoplay	Specifies that the audio will start playing as soon as it is ready
<u>controls</u>	controls	Specifies that audio controls should be displayed (such as a play/pause button etc)
<u>loop</u>	loop	Specifies that the audio will start over again, every time it is finished
<u>muted</u>	muted	Specifies that the audio output should be muted
<u>preload</u>	auto metadata none	Specifies if and how the author thinks the audio should be loaded when the page loads
<u>src</u>	URL	Specifies the URL of the audio file

Browser	MP3	WAV	OGG
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES	YES	YES
Safari	YES	YES	NO
Opera	YES	YES	YES

# Ljud och video

- *Både ljud och video elementen kan styras via JavaScript*

## HTML Audio/Video Methods

Method	Description
<a href="#"><u>addTextTrack()</u></a>	Adds a new text track to the audio/video
<a href="#"><u>canPlayType()</u></a>	Checks if the browser can play the specified audio/video type
<a href="#"><u>load()</u></a>	Re-loads the audio/video element
<a href="#"><u>play()</u></a>	Starts playing the audio/video
<a href="#"><u>pause()</u></a>	Pauses the currently playing audio/video

```
1 <audio id="my-audio" controls>
2   <source src="horse.ogg" type="audio/ogg">
3   <source src="horse.mp3" type="audio/mpeg">
4   Your browser does not support the audio tag.
5 </audio>
6
7 <script>
8   let audio = document.getElementById("my-audio");
9   audio.play();
10  audio.pause();
11  // Etc.
12 </script>
13
14
```

# Frågor?