

GWT + HTML5

A web developers dream!

John LaBanca
May 11, 2011

Agenda

- Overview
- *Local Storage* for performance
- Visualizations via *Canvas*
- Enrich your app with *Audio*
- Embedding *Video* in your app
- *Drag and Drop* support
- Handling unsupported browser versions
- **CSS3** makes its way into GWT
- Questions

Hash Tags

Google I/O:
#io2011

This session:
#DevTools

Feedback?

<http://goo.gl/3i6vS>

Feedback?

<http://goo.gl/3i6vS>
(positive feedback only)

For negative feedback:
<http://127.0.0.1/feedback>

Overview

- Hi, I'm jlabanca@google.com
- What to expect
 - Lots of demos
 - Lots of code examples
- What you will learn
 - How to use HTML5 in your GWT app

HTML



Overview

What is HTML5?

- HTML5 includes lots of stuff
 - Media rich DOM elements
 - Useful new APIs
 - CSS3 enhancements
- Support is increasing very rapidly
- Embrace it to make your apps stand out

HTML



Agenda

- Overview
- ***Local Storage*** for performance
- Visualizations via *Canvas*
- Enrich your app with *Audio*
- Embedding *Video* in your app
- *Drag and Drop* support
- Handling unsupported browser versions
- CSS3 makes its way into GWT
- Questions

Chrome

Safari

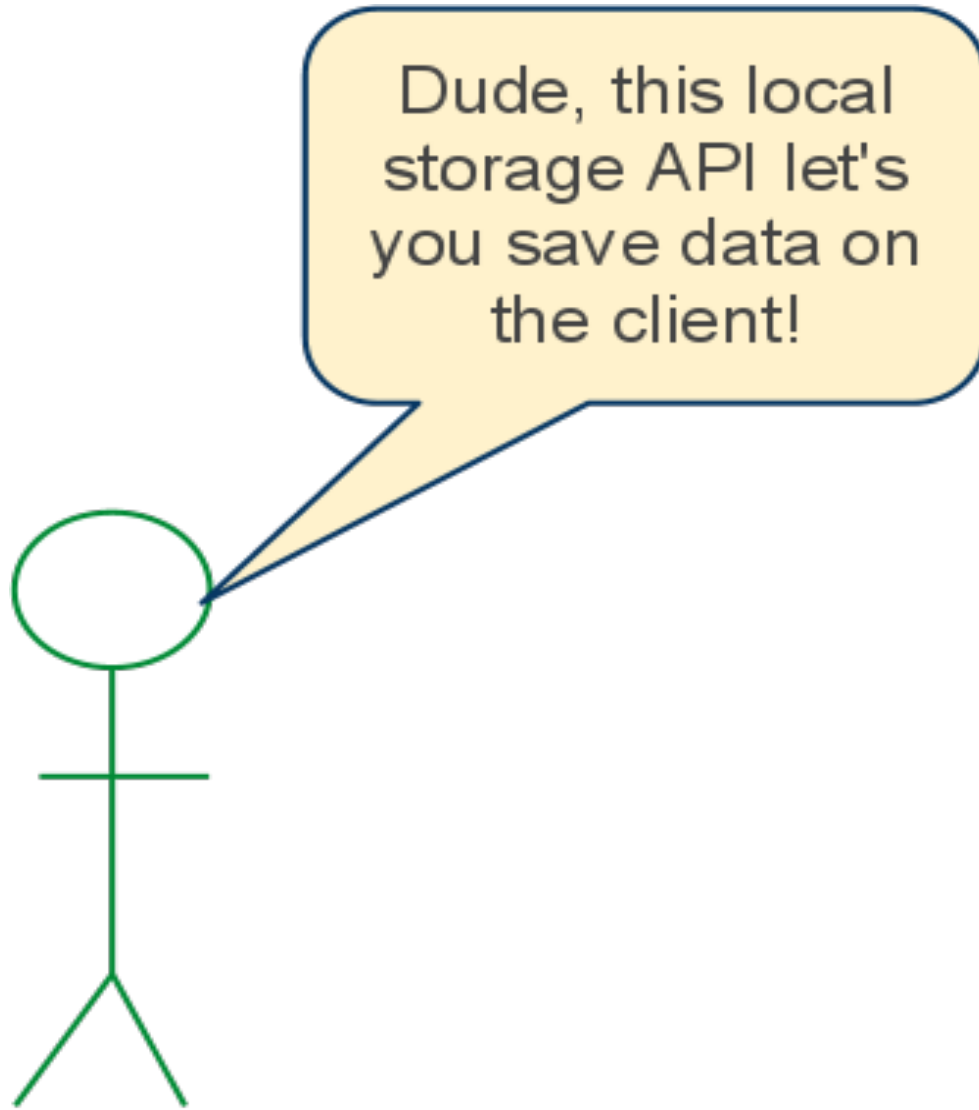
Firefox

IE 8+

Opera

Local Storage

Isn't it just a cookie?



Local Storage

Available in GWT

- GWT provides a Storage API

```
Storage storage = Storage.getLocalStorageIfSupported();  
if (storage != null) {  
    storage.setItem("foo", "bar");  
}
```

```
Storage storage = Storage.getLocalStorageIfSupported();  
if (storage != null) {  
    String value = storage.getItem("foo");  
}
```

Local Storage

Available in GWT

- GWT provides a Storage API

```
Storage storage = Storage.getLocalStorageIfSupported();
if (storage != null) {
    storage.setItem("foo", "bar");
}
```

```
Storage storage = Storage.getLocalStorageIfSupported();
if (storage != null) {
    String value = storage.getItem("foo");
}
```

Local Storage

Available in GWT

- GWT provides a Storage API

```
Storage storage = Storage.getLocalStorageIfSupported();  
if (storage != null) {  
    storage.setItem("foo", "bar");  
}
```

```
Storage storage = Storage.getLocalStorageIfSupported();  
if (storage != null) {  
    String value = storage.getItem("foo");  
}
```

Local Storage

Available in GWT

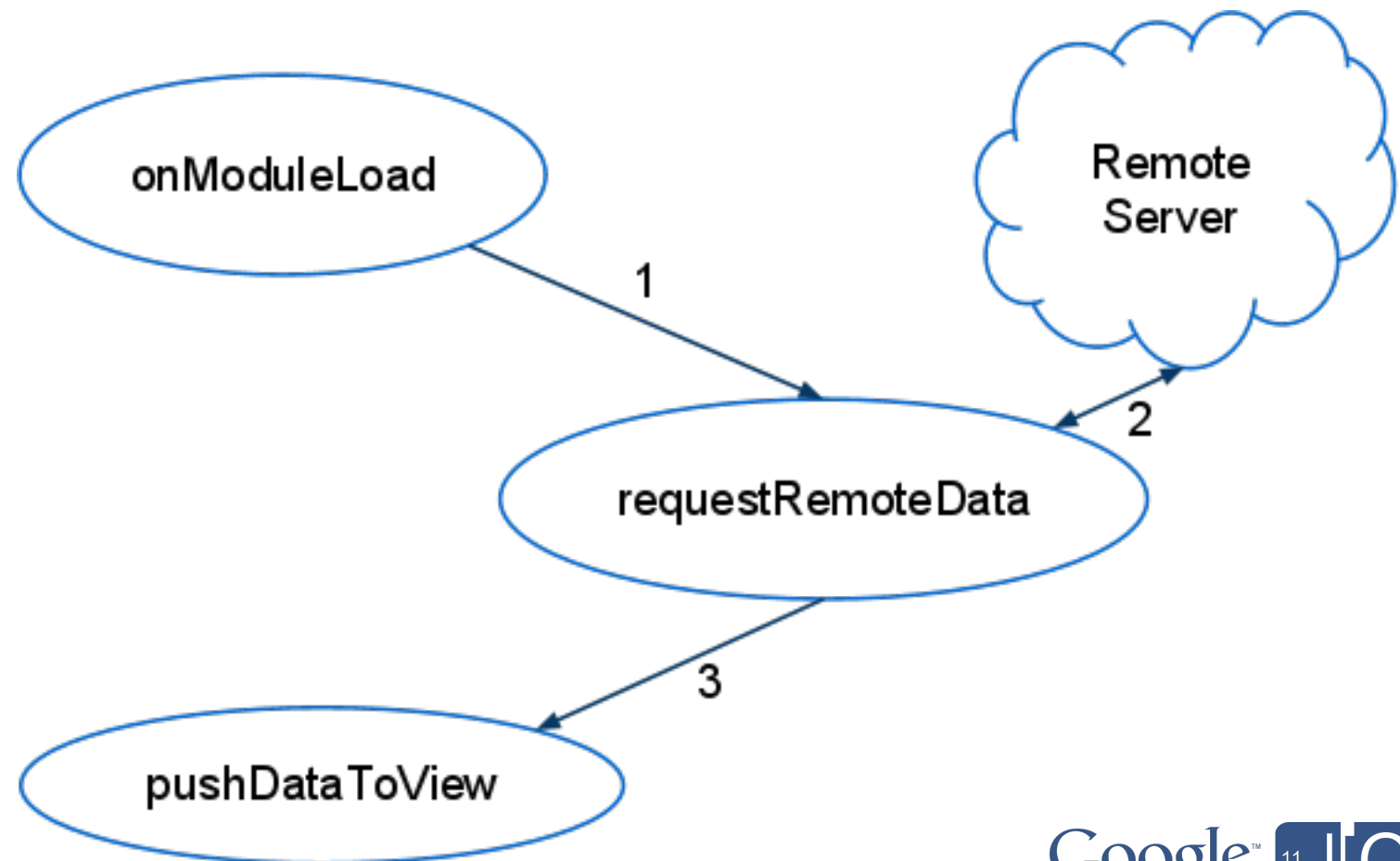
- StorageMap lets you access storage using a java.util.Map
 - Contributions by Bart Guijt

```
Storage storage = Storage.getLocalStorageIfSupported();
if (storage != null) {
    StorageMap map = new StorageMap(storage);
    storage.put("foo", "bar");
}
```

Local Storage

For improved performance

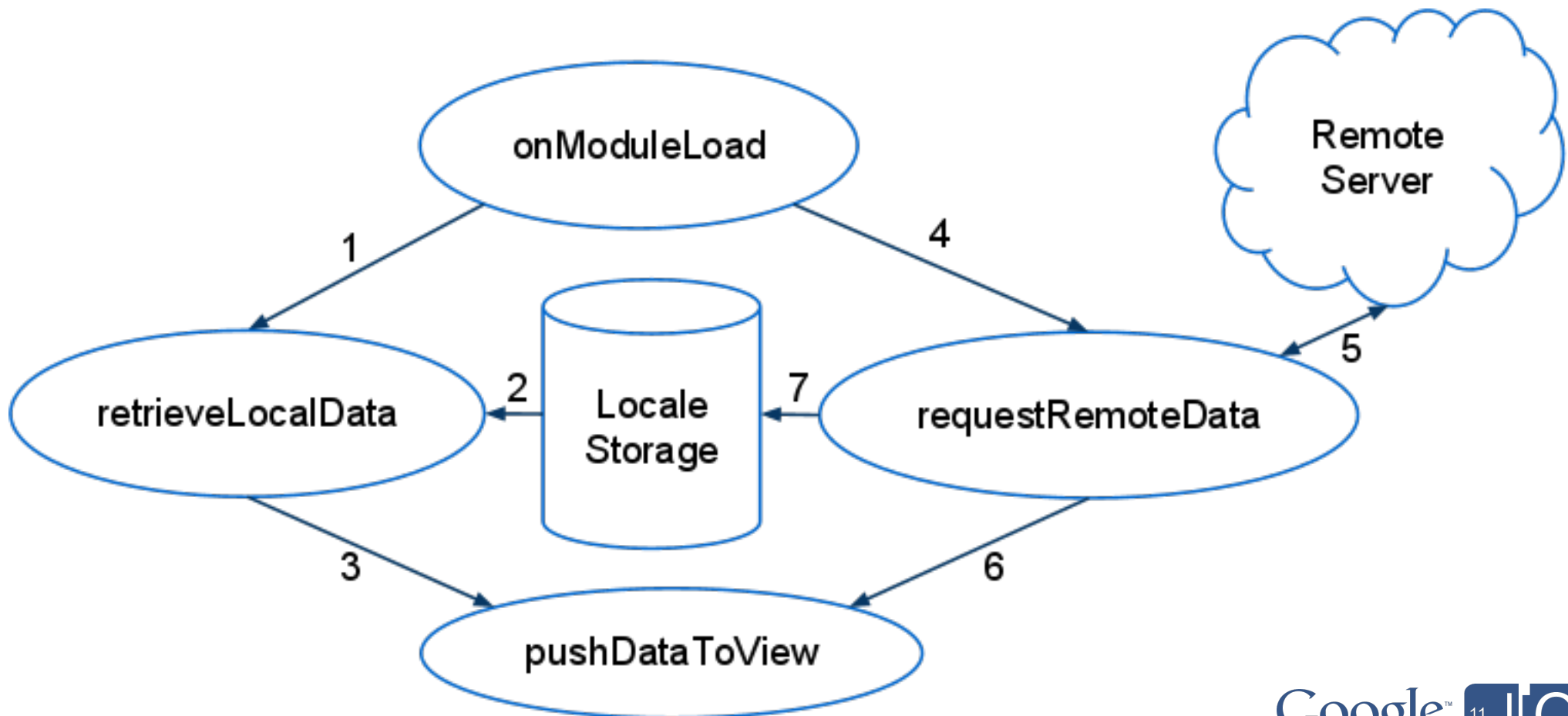
- Cache data from RPC calls
- Load cached data on startup for faster startup times



Local Storage

For improved performance

- Cache data from RPC calls
- Load cached data on startup for faster startup times



Local Storage

For improved performance

Demo

Local Storage

For improved user experience

- Save temporary state to local storage
- Restore state when user re-enters app
- Prevent lost work on network disconnect

Agenda

- Overview
- *Local Storage* for performance
- **Visualizations via *Canvas***
- Enrich your app with *Audio*
- Embedding *Video* in your app
- *Drag and Drop* support
- Handling unsupported browser versions
- CSS3 makes its way into GWT
- Questions

Chrome

Safari

Firefox

IE 9+

Opera

Canvas

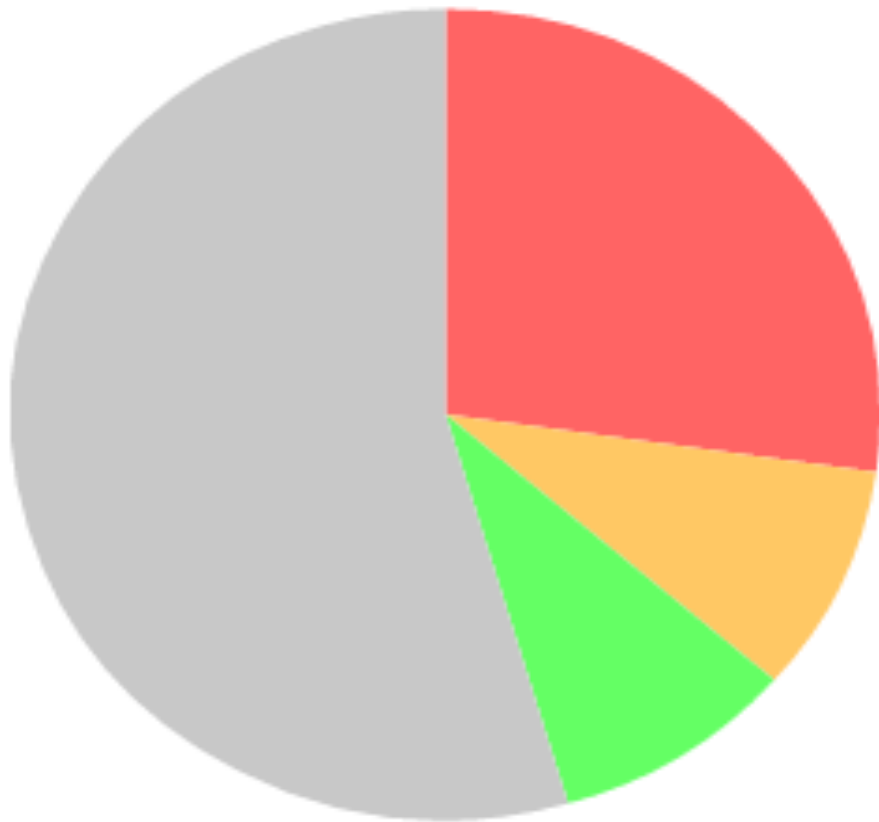
Adding visualizations to your app

- This slide is boring

Past Due	3
Due Tomorrow	1
On Time	1
No Due Date	6

Canvas

Adding visualizations to your app



- Past Due
- Due Tomorrow
- On Time
- No Due Date

- Charts bring data to life
- Canvas brings charts to the web

Canvas

Adding visualizations to your app

Demo

Canvas

Adding visualizations to your app

1. Initialize the canvas
2. Calculate values for new shapes
3. Clear the current canvas
4. Draw the new shapes



```
// Initialize the canvas.  
final Canvas canvas = Canvas.createIfSupported();  
canvas.setCoordinateSpaceHeight(300);  
canvas.setCoordinateSpaceWidth(300);  
canvas.setPixelSize(300, 300);  
  
// Get the dimensions of the canvas.  
int width = canvas.getCoordinateSpaceWidth();  
int height = canvas.getCoordinateSpaceHeight();  
double radius = Math.min(width, height) / 2.0;  
double cx = width / 2.0;  
double cy = height / 2.0;  
  
// Clear the context.  
Context2d context = canvas.getContext2d();  
context.clearRect(0, 0, width, height);  
  
// Draw a filled arc.  
context.setFillStyle(CssColor.make(255, 211, 25));  
context.beginPath();  
context.moveTo(cx, cy);  
context.arc(cx, cy, radius, 0, Math.PI);  
context.fill();
```



```
// Initialize the canvas.
final Canvas canvas = Canvas.createIfSupported();
canvas.setCoordinateSpaceHeight(300);
canvas.setCoordinateSpaceWidth(300);
canvas.setPixelSize(300, 300);
// Get the dimensions of the canvas.
int width = canvas.getCoordinateSpaceWidth();
int height = canvas.getCoordinateSpaceHeight();
double radius = Math.min(width, height) / 2.0;
double cx = width / 2.0;
double cy = height / 2.0;
// Clear the context.
Context2d context = canvas.getContext2d();
context.clearRect(0, 0, width, height);
// Draw a filled arc.
context.setFillStyle(CssColor.make(255, 211, 25));
context.beginPath();
context.moveTo(cx, cy);
context.arc(cx, cy, radius, 0, Math.PI);
context.fill();
```

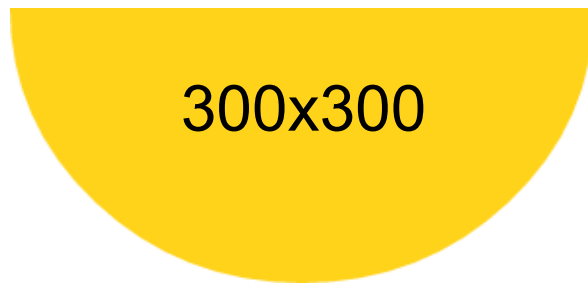
```
// Initialize the canvas.
final Canvas canvas = Canvas.createIfSupported();
canvas.setCoordinateSpaceHeight(300);
canvas.setCoordinateSpaceWidth(300);
canvas.setPixelSize(300, 300);
// Get the dimensions of the canvas.
int width = canvas.getCoordinateSpaceWidth();
int height = canvas.getCoordinateSpaceHeight();
double radius = Math.min(width, height) / 2.0;
double cx = width / 2.0;
double cy = height / 2.0;
// Clear the context.
Context2d context = canvas.getContext2d();
context.clearRect(0, 0, width, height);
// Draw a filled arc.
context.setFillStyle(CssColor.make(255, 211, 25));
context.beginPath();
context.moveTo(cx, cy);
context.arc(cx, cy, radius, 0, Math.PI);
context.fill();
```

```
// Initialize the canvas.
final Canvas canvas = Canvas.createIfSupported();
canvas.setCoordinateSpaceHeight(300);
canvas.setCoordinateSpaceWidth(300);
canvas.setPixelSize(300, 300);
// Get the dimensions of the canvas.
int width = canvas.getCoordinateSpaceWidth();
int height = canvas.getCoordinateSpaceHeight();
double radius = Math.min(width, height) / 2.0;
double cx = width / 2.0;
double cy = height / 2.0;
// Clear the context.
Context2d context = canvas.getContext2d();
context.clearRect(0, 0, width, height);
// Draw a filled arc.
context.setFillStyle(CssColor.make(255, 211, 25));
context.beginPath();
context.moveTo(cx, cy);
context.arc(cx, cy, radius, 0, Math.PI);
context.fill();
```

Canvas

Adding visualizations to your app

- Canvas has a coordinate space separate from DOM size
- Coordinate space is scaled to the DOM size
 - Small coordinate space = faster performance
 - Large coordinate space = higher quality
- Match coordinate space to DOM size
 - Assign a fixed size to the DOM element, or
 - Catch resize events and adjust the coordinate space



Agenda

- Overview
- *Local Storage* for performance
- Visualizations via *Canvas*
- **Enrich your app with Audio**
- Embedding *Video* in your app
- *Drag and Drop* support
- Handling unsupported browser versions
- CSS3 makes its way into GWT
- Questions

Chrome

Safari

Firefox

IE 8+

Opera

Audio

- Embed audio using the Audio widget
- Play a sound or stream



Audio

Demo

Audio

- Format support varies across browsers
- Choose *at least two* of the three main formats

	Ogg Vorbis	MP3	WAV	
Chrome	✓	✓		
Firefox	✓		✓	
Internet Explorer		✓	✓	
Opera	✓		✓	
Safari		✓	✓	

Source: <http://html5doctor.com/native-audio-in-the-browser/>

Audio

Specify multiple sources

- You can specify multiple sources in HTML5
- Don't forget the "type" attribute
 - Browsers will load all source files
 - Must load meta data to detect mime-type
 - Cannot assume mime-type from file extension

```
<audio controls>  
  <source src="path/file.ogg" type="audio/ogg" />  
  <source src="path/file.mp3" type="audio/mpeg" />  
  <source src="path/file.wav" type="audio/wav" />  
</audio>
```

Audio

Specify multiple sources

```
public Audio createAudio() {
    Audio audio = Audio.createIfSupported();
    if (audio == null) {
        return;
    }

    audio.addSource("path/file.ogg", AudioElement.TYPE_OGG);
    audio.addSource("path/file.mp3", AudioElement.TYPE_MP3);
    audio.addSource("path/file.wav", AudioElement.TYPE_WAV);

    // Show audio controls.
    audio.setControls(true);
    return audio;
}
```

Audio

Specify multiple sources

```
public Audio createAudio() {  
    Audio audio = Audio.createIfSupported();  
    if (audio == null) {  
        return;  
    }  
  
    audio.addSource("path/file.ogg", AudioElement.TYPE_OGG);  
    audio.addSource("path/file.mp3", AudioElement.TYPE_MP3);  
    audio.addSource("path/file.wav", AudioElement.TYPE_WAV);  
  
    // Show audio controls.  
    audio.setControls(true);  
    return audio;  
}
```

Audio

Specify multiple sources

```
public Audio createAudio() {
    Audio audio = Audio.createIfSupported();
    if (audio == null) {
        return;
    }

    audio.addSource("path/file.ogg", AudioElement.TYPE_OGG);
    audio.addSource("path/file.mp3", AudioElement.TYPE_MP3);
    audio.addSource("path/file.wav", AudioElement.TYPE_WAV);

    // Show audio controls.
    audio.setControls(true);
    return audio;
}
```

Audio

Specify multiple sources

```
public Audio createAudio() {
    Audio audio = Audio.createIfSupported();
    if (audio == null) {
        return;
    }

    audio.addSource("path/file.ogg", AudioElement.TYPE_OGG);
    audio.addSource("path/file.mp3", AudioElement.TYPE_MP3);
    audio.addSource("path/file.wav", AudioElement.TYPE_WAV);

    // Show audio controls.
    audio.setControls(true);
    return audio;
}
```

Audio

Manually choose the source

- You can ask the browser which files it *might* support
 - Tells you "probably", "maybe", or ""
- Ensures that you only load that source

```
public Audio createAudio() {
    Audio audio = Audio.createIfSupported();
    if (audio == null) {
        return;
    }
    if (MediaElement.CAN_PLAY_PROBABLY.equals(
        audio.canPlayType(AudioElement.TYPE_OGG)) {
        audio.setSrc("path/file.ogg");
    } else if (MediaElement.CAN_PLAY_PROBABLY.equals(
        audio.canPlayType(AudioElement.TYPE_MP3)) {
        audio.setSrc("path/file.mp3");
    } else if (MediaElement.CAN_PLAY_MAYBE.equals(
        audio.canPlayType(AudioElement.TYPE_OGG)) {
        audio.setSrc("path/file.mp3");
    } else if (MediaElement.CAN_PLAY_MAYBE.equals(
        audio.canPlayType(AudioElement.TYPE_MP3)) {
        audio.setSrc("path/file.mp3");
    }
    return audio;
}
```

```
public Audio createAudio() {
    Audio audio = Audio.createIfSupported();
    if (audio == null) {
        return;
    }
    if (MediaElement.CAN_PLAY_PROBABLY.equals(
        audio.canPlayType(AudioElement.TYPE_OGG)) {
        audio.setSrc("path/file.ogg");
    } else if (MediaElement.CAN_PLAY_PROBABLY.equals(
        audio.canPlayType(AudioElement.TYPE_MP3)) {
        audio.setSrc("path/file.mp3");
    } else if (MediaElement.CAN_PLAY_MAYBE.equals(
        audio.canPlayType(AudioElement.TYPE_OGG)) {
        audio.setSrc("path/file.mp3");
    } else if (MediaElement.CAN_PLAY_MAYBE.equals(
        audio.canPlayType(AudioElement.TYPE_MP3)) {
        audio.setSrc("path/file.mp3");
    }
    return audio;
}
```


Audio

Preloading a file

- Let the browser decide when to preload

```
Audio audio = Audio.createIfSupported();  
audio.setSrc("path/file.ogg");  
audio.setPreload(MediaElement.PRELOAD_AUTO);
```

- Or force the browser to preload

```
Audio audio = Audio.createIfSupported();  
audio.setSrc("path/file.ogg");  
audio.load();
```

Audio

Preloading a file

- Let the browser decide when to preload

```
Audio audio = Audio.createIfSupported();  
audio.setSrc("path/file.ogg");  
audio.setPreload(MediaElement.PRELOAD_AUTO);
```

- Or force the browser to preload

```
Audio audio = Audio.createIfSupported();  
audio.setSrc("path/file.ogg");  
audio.load();
```

Audio

Preloading a file

- Let the browser decide when to preload

```
Audio audio = Audio.createIfSupported();  
audio.setSrc("path/file.ogg");  
audio.setPreload(MediaElement.PRELOAD_AUTO);
```

- Or force the browser to preload

```
Audio audio = Audio.createIfSupported();  
audio.setSrc("path/file.ogg");  
audio.load();
```

Agenda

- Overview
- *Local Storage* for performance
- Visualizations via *Canvas*
- Enrich your app with *Audio*
- **Embedding *Video* in your app**
- *Drag and Drop* support
- Handling unsupported browser versions
- CSS3 makes its way into GWT
- Questions

Chrome

Safari

Firefox

IE 8+

Opera

Video

- Embed videos using the Video widget
- Play a static video or a stream



Video

Demo

Video

- Format support varies across browsers
- Choose *at least two* of the three main formats
 - Safari only supports mp4

	Ogg Vorbis	MP4	WebM	
Chrome	✓		✓	
Firefox	✓		✓	
Internet Explorer		✓		
Opera	✓		✓	
Safari		✓		

Video

```
public Video createVideo() {  
    Video video = Video.createIfSupported();  
    if (video == null) {  
        return null;  
    }  
  
    video.addSource("path/file.ogg", VideoElement.TYPE_OGG);  
    video.addSource("path/file.mp4", VideoElement.TYPE_MP4);  
    video.addSource("path/file.webm", VideoElement.TYPE_WEBM);  
  
    // Show audio controls.  
    video.setControls(true);  
    return video;  
}
```


Video

```
public Video createVideo() {  
    Video video = Video.createIfSupported();  
    if (video == null) {  
        return null;  
    }  
  
    video.addSource("path/file.ogg", VideoElement.TYPE_OGG);  
    video.addSource("path/file.mp4", VideoElement.TYPE_MP4);  
    video.addSource("path/file.webm", VideoElement.TYPE_WEBM);  
  
    // Show audio controls.  
    video.setControls(true);  
    return video;  
}
```

Video

```
public Video createVideo() {  
    Video video = Video.createIfSupported();  
    if (video == null) {  
        return null;  
    }  
  
    video.addSource("path/file.ogg", VideoElement.TYPE_OGG);  
    video.addSource("path/file.mp4", VideoElement.TYPE_MP4);  
    video.addSource("path/file.webm", VideoElement.TYPE_WEBM);  
  
    // Show audio controls.  
    video.setControls(true);  
    return video;  
}
```

Video

```
public Video createVideo() {
    Video video = Video.createIfSupported();
    if (video == null) {
        return null;
    }

    video.addSource("path/file.ogg", VideoElement.TYPE_OGG);
    video.addSource("path/file.mp4", VideoElement.TYPE_MP4);
    video.addSource("path/file.webm", VideoElement.TYPE_WEBM);

    // Show audio controls.
    video.setControls(true);
    return video;
}
```

Agenda

- Overview
- *Local Storage* for performance
- Visualizations via *Canvas*
- Enrich your app with *Audio*
- Embedding *Video* in your app
- ***Drag and Drop*** support
- Handling unsupported browser versions
- CSS3 makes its way into GWT
- Questions

Chrome

Safari

Firefox

IE 8+

Opera

Drag and Drop

- Move stuff around... with style
- Associate text or ID with the event
- Many types of drag events

Task Details

Task Name:

Register to vote

Notes:

Where is my polling location again?

Due date:

Set due date

Add to my calendar

Done

Cancel

Templates:

Drag a template over the form to the left use it.

Call mom

Register to vote
Where is my polling location again?

Register to vote
Where is my polling location again?

Take out the trash

Drag and Drop

Demo

Drag and Drop

Make a widget draggable

- Set the "draggable" attribute to "true"
- Add a DragStart handler
- Set data in the DragStart handler

Drag and Drop

Make a widget draggable

```
// Make the widget draggable.  
Label w = new Label("Drag Me");  
w.getElement().setDraggable(Element.DRAGGABLE_TRUE);  
  
// Add a DragStartHandler.  
w.addDragStartHandler(new DragStartHandler() {  
    public void onDragStart(DragStartEvent event) {  
        // Required: set data for the event.  
        event.setData("text", "Hello World");  
  
        // Optional: show a copy of the widget under cursor.  
        event.getDataTransfer().setDragImage(w.getElement(),  
            10, 10);  
    }  
});
```


Drag and Drop

Make a widget draggable

```
// Make the widget draggable.  
Label w = new Label("Drag Me");  
w.getElement().setDraggable(Element.DRAGGABLE_TRUE);  
  
// Add a DragStartHandler.  
w.addDragStartHandler(new DragStartHandler() {  
    public void onDragStart(DragStartEvent event) {  
        // Required: set data for the event.  
        event.setData("text", "Hello World");  
  
        // Optional: show a copy of the widget under cursor.  
        event.getDataTransfer().setDragImage(w.getElement(),  
            10, 10);  
    }  
});
```

Drag and Drop

Make a widget draggable

```
// Make the widget draggable.  
Label w = new Label("Drag Me");  
w.getElement().setDraggable(Element.DRAGGABLE_TRUE);  
  
// Add a DragStartHandler.  
w.addDragStartHandler(new DragStartHandler() {  
    public void onDragStart(DragStartEvent event) {  
        // Required: set data for the event.  
        event.setData("text", "Hello World");  
  
        // Optional: show a copy of the widget under cursor.  
        event.getDataTransfer().setDragImage(w.getElement(),  
            10, 10);  
    }  
});
```

Drag and Drop

Make a widget draggable

```
// Make the widget draggable.  
Label w = new Label("Drag Me");  
w.getElement().setDraggable(Element.DRAGGABLE_TRUE);  
  
// Add a DragStartHandler.  
w.addDragStartHandler(new DragStartHandler() {  
    public void onDragStart(DragStartEvent event) {  
        // Required: set data for the event.  
        event.setData("text", "Hello World");  
  
        // Optional: show a copy of the widget under cursor.  
        event.getDataTransfer().setDragImage(w.getElement(),  
            10, 10);  
    }  
});
```

Drag and Drop

Make a widget draggable

```
// Make the widget draggable.  
Label w = new Label("Drag Me");  
w.getElement().setDraggable(Element.DRAGGABLE_TRUE);  
  
// Add a DragStartHandler.  
w.addDragStartHandler(new DragStartHandler() {  
    public void onDragStart(DragStartEvent event) {  
        // Required: set data for the event.  
        event.setData("text", "Hello World");  
  
        // Optional: show a copy of the widget under cursor.  
        event.getDataTransfer().setDragImage(w.getElement(),  
            10, 10);  
    }  
});
```

Drag and Drop

Create a target

- Add a DragOver handler
- Add a Drop handler
 - Prevent default action
 - Get the data from the transfer object

```
Label target = new Label("Drag onto me");
```

```
// Required: You must add a DragOverHandler to  
// create a target.  
target.addDragOverHandler(new DragOverHandler() {  
    public void onDragOver(DragOverEvent event) {  
        target.getElement().setBackgroundColor("#ffa");  
    }  
});
```

```
// Add a DropHandler.  
target.addDropHandler(new DropHandler() {  
    public void onDrop(DropEvent event) {  
        // Prevent the native text drop.  
        event.preventDefault();  
  
        // Get the data out of the event.  
        String data = event.getData("text");  
        target.setText(data);  
    }  
});
```

```
Label target = new Label("Drag onto me");

// Required: You must add a DragOverHandler to
// create a target.
target.addDragOverHandler(new DragOverHandler() {
    public void onDragOver(DragOverEvent event) {
        target.getElement().setBackgroundColor("#ffa");
    }
});
```

```
// Add a DropHandler.
target.addDropHandler(new DropHandler() {
    public void onDrop(DropEvent event) {
        // Prevent the native text drop.
        event.preventDefault();
```

```
        // Get the data out of the event.
        String data = event.getData("text");
        target.setText(data);
    }
});
```

```
Label target = new Label("Drag onto me");

// Required: You must add a DragOverHandler to
// create a target.
target.addDragOverHandler(new DragOverHandler() {
    public void onDragOver(DragOverEvent event) {
        target.getElement().setBackgroundColor("#ffa");
    }
});

// Add a DropHandler.
target.addDropHandler(new DropHandler() {
    public void onDrop(DropEvent event) {
        // Prevent the native text drop.
        event.preventDefault();

        // Get the data out of the event.
        String data = event.getData("text");
        target.setText(data);
    }
});
```



```
Label target = new Label("Drag onto me");

// Required: You must add a DragOverHandler to
// create a target.
target.addDragOverHandler(new DragOverHandler() {
    public void onDragOver(DragOverEvent event) {
        target.getElement().setBackgroundColor("#ffa");
    }
});

// Add a DropHandler.
target.addDropHandler(new DropHandler() {
    public void onDrop(DropEvent event) {
        // Prevent the native text drop.
        event.preventDefault();

        // Get the data out of the event.
        String data = event.getData("text");
        target.setText(data);
    }
});
```

```
Label target = new Label("Drag onto me");

// Required: You must add a DragOverHandler to
// create a target.
target.addDragOverHandler(new DragOverHandler() {
    public void onDragOver(DragOverEvent event) {
        target.getElement().setBackgroundColor("#ffa");
    }
});

// Add a DropHandler.
target.addDropHandler(new DropHandler() {
    public void onDrop(DropEvent event) {
        // Prevent the native text drop.
        event.preventDefault();

        // Get the data out of the event.
        String data = event.getData("text");
        target.setText(data);
    }
});
```

Drag and Drop

On any widget

- Some widgets do not implement drag handlers
- You can always use `addDomHandler()` to add a handler

```
// Add a DragStartHandler.  
w.addDomHandler(new DragStartHandler() {  
    public void onDragStart(DragStartEvent event) {  
        // Required: set data for the event.  
        event.setData("text", "Hello World");  
  
        // Optional: show a copy of the widget under cursor.  
        event.getDataTransfer().setDragImage(w.getElement,  
            10, 10);  
    }  
}, DragStartEvent.getType());
```

Agenda

- Overview
- *Local Storage* for performance
- Visualizations via *Canvas*
- Enrich your app with *Audio*
- Embedding *Video* in your app
- *Drag and Drop* support
- **Handling unsupported browser versions**
- CSS3 makes its way into GWT
- Questions

Partial Support

Because if all browsers supported it, it would be too easy

- GWT provides APIs that are not supported in all browsers
 - Modern browsers gaining market share rapidly
- Implementors of PartialSupport provide two static methods:
 - `isSupported()`
 - `createIfSupported()`
- Three recipes for handling legacy browsers:
 1. Fallback to another implementation
 2. Disable the feature
 3. Suggest that the user upgrade to a newer browser

Partial Support

1. Fallback to another implementation

```
public void showChart(AcceptsOneWidget panel,  
    ChartData data) {  
    Canvas canvas = Canvas.createIfSupported();  
    if (canvas == null) {  
        // Fallback to an image if Canvas isn't supported.  
        String imageUrl = encodeChartData(data);  
        panel.setWidget(new Image(imageUrl));  
        return;  
    }  
  
    // Create a canvas chart.  
    ...  
  
    // Display the canvas.  
    panel.setWidget(canvas);  
}
```

Partial Support

2. Disable the feature

```
public void showChart(AcceptsOneWidget panel,  
    ChartData data) {  
    Canvas canvas = Canvas.createIfSupported();  
    if (canvas == null) {  
        return;  
    }  
  
    // Create a canvas chart.  
    ...  
  
    // Display the canvas.  
    panel.setWidget(canvas);  
}
```

Partial Support

3. Upgrade to a newer browser

```
public void showChart(AcceptsOneWidget panel,  
    ChartData data) {  
    Canvas canvas = Canvas.createIfSupported();  
    if (canvas == null) {  
        // Try upgrading to a modern browser.  
        String message = "This feature is not supported. "  
            + "Try upgrading to a newer browser.";  
        panel.setWidget(new Label(message));  
        return;  
    }  
  
    // Create a canvas chart.  
    ...  
  
    // Display the canvas.  
    panel.setWidget(canvas);  
}
```


Agenda

- Overview
- *Local Storage* for performance
- Visualizations via *Canvas*
- Enrich your app with *Audio*
- Embedding *Video* in your app
- *Drag and Drop* support
- Handling unsupported browser versions
- **CSS3 makes its way into GWT**
- Questions

Chrome

Safari

Firefox

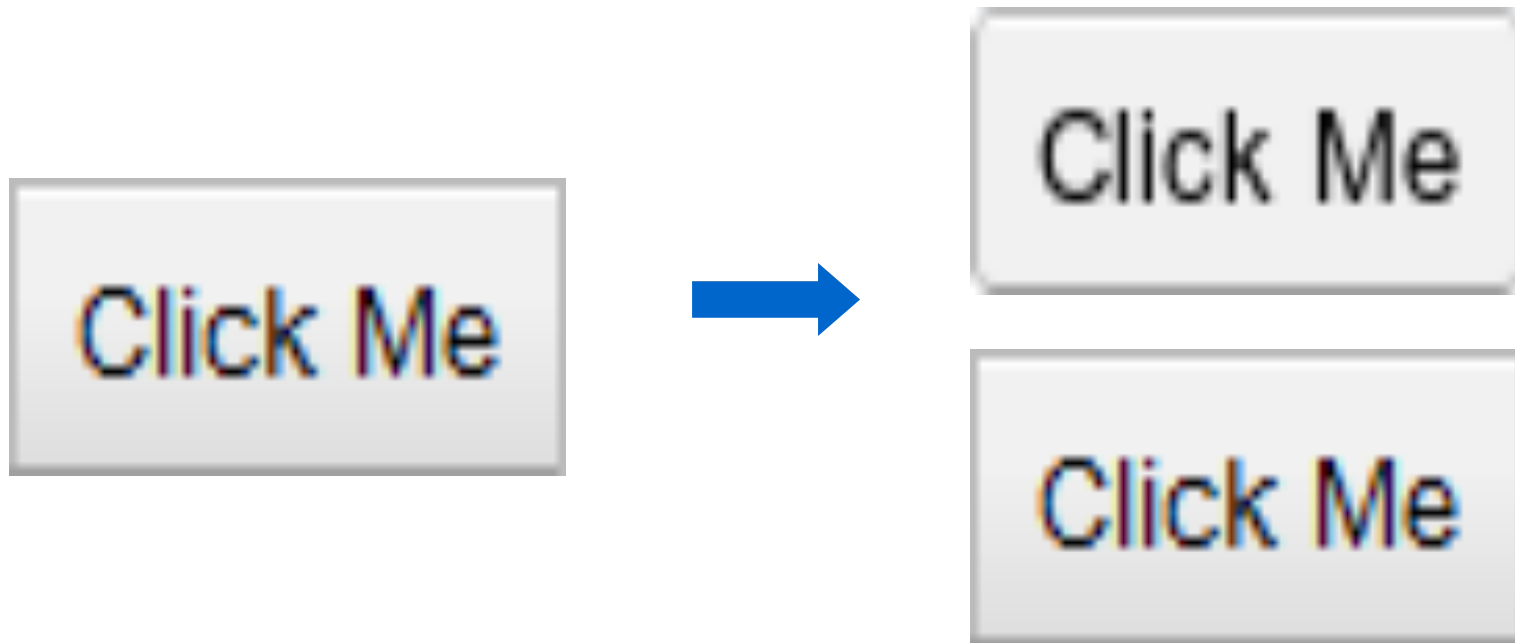
IE 9+

Opera

CSS3 in Widgets

- As of GWT 2.3, widgets now use CSS3
- On legacy browsers, widgets will be "usable"

Example: Rounded Corners



Recap

- *Local Storage* improves performance and experience
- *Canvas* highlights important data
- *Audio* and *video* add dimension
- *Drag and Drop* simplifies interactions
- CSS3 can transform the UI
- Apps must check for feature support

Submit feedback:
<http://goo.gl/3i6vS>

Try GWT

Download GWT from our homepage
<http://code.google.com/webtoolkit/>

GWT 2.4 introduced many of the features described in this presentation.

Stay Tuned...

High-Performance GWT:

best practices for writing smaller, faster apps

"The GWT compiler isn't just a Java to JavaScript transliterator. In this session, we'll show you compiler optimizations to shrink your app and make it compile and run faster. Learn common performance pitfalls, how to use lightweight cell widgets, how to use code splitting with Activities and Places, and compiler options to reduce your app's size and compile time."

Google™

