

Technical Specifications HTML5

Overview

Technical Specification	Specifications
Delivery format	Single HTML file + additional images/videos
3rd-Party-Tracking	allowed
Videoformats	H264/mp4 und VP8/WebM
Video filesize	max. 5 MB
Videostart & Audiostart	Only after explicit useraction
Leadtime	5 days or 10 days for mobile HTML5 formats

Guidelines

General Guidelines

- Deliver the HTML5 media ad as index.html, with full HTML syntax (Doctype definitions, , , etc.)
- As much content (scripts, CSS, etc.) as possible should be included directly in the index.html instead of being outsourced to separate files.
- When reloading external resources, make sure that they are enabled for cross-domain access on the server side (CORS).
- Use an alternative graphic to replace the ad media for browsers that do not fully support HTML5.
- The orientation of the displaying device should be taken into account and the advertising medium should be optimized for "Landscape" and "Portrait" displays.
- The tracking/counting of user actions (except clicks) by a separate server must be done by the ad creator.

JavaScript Guidelines

- Use small, compact JS frameworks. Larger libraries like jQuery are usually unnecessary and increase the data load of the ad media exponentially.
- Reduce your final JS code as much as possible. Use minifiers or similar tools so that your ad can be transferred to the viewer as quickly as possible.
- Avoid logs (e.g. console.log or etc.) and possible errors to optimize the performance of your ad and not overload the browser console.

HTML 5 Redirect

- **HTML5** advertising material must be delivered as an **iFrame redirect**. When delivering, please note that 3rd party counting pixels and click commands must already be integrated into the HTML by the creator.
- The redirect must be **100% dynamic** so that the advert can be displayed as a multiscreen product.

Click Measurement

- In order to measure clicks on links to external sites via the Ströer Digital AdServer, it is necessary for the HTML5 advert to allow for the dynamic assignment of link URLs.
- When the ad medium is played out, the complete URLs of Ströer are inserted into the ad medium and measured when the user clicks.
- During development of the media ad, the link URL-encoded (encodeURIComponent()) can be transferred to the ad media using GET parameters.
- `yourAdvertisingMaterial.html?clicktag1=http%3A%2F%2Fwww.youreUrl.de&clicktag2=http%3A%2F%2Fwww.yourOtherUrl.de`

- Fill all links to be measured with the name "#clicktag" as placeholder and an ID attribute of the same name. If you want to use different links with different targets, you have to number the placeholders sequentially ("clicktag", "clicktag2", etc.)

```
<a href="#clicktag" id="clicktag" target="_blank">Linktext 1</a>
<a href="#clicktag2" id="clicktag2" target="_blank">Linktext 2</a>
```

- Insert the following script in the <head> of your ad media:

```
<script>
var getUriParams = function ()
{
    var query_string = {};
    var query = window.location.search.substring(1);
    var parmsArray = query.split('&');
    if (parmsArray.length <= 0) return query_string;
    for (var i = 0; i < parmsArray.length; i++)
    {
        var pair = parmsArray[i].split('=');
        var val = decodeURIComponent(pair[1]);
        if (val != '' && pair[0] != '') query_string[pair[0]] = val;
    }
    return query_string;
}();
</script>
```

- Either at the end of the HTML file or when initiating the ad media, the dynamic assignment of the URLs must be triggered with the following code

```
<script> document.getElementById('clicktag').setAttribute('href', getUriParams.clicktag); document.getElementById('clicktag2').setAttribute('href', getUriParams.clicktag2); </script>
```

Pre-test click measurement

- The advertising material should be tested by the creative agency for functioning clicktag transfer so that unnecessary feedback loops can be excluded.
- Test: `html5mediaad.html?clicktag=%LANDINGPAGE%`
- ***%LANDINGPAGE%** is to be replaced with a test target page and must be passed URL - encoded (either via the `encodeURIComponent` function or e.g. with the help of this page: <https://meyerweb.com/eric/tools/dencoder/>)

Guidelines for animations

- Animations should be created as resource-saving as possible. Avoid the excessive use of transparency effects (opacity) or the triggering of browser "repaints". You can find more information about paint/repaint [here](#).
- Remove unneeded GIFs (e.g. a loading animation) completely from the HTML tree instead of just hiding the image. Hidden GIF animations continue to burden the browser.
- Use the animation possibilities of CSS3 (Animation, Keyframe, Translate, etc.)
- Never use animations with JavaScript's `setInterval()` or `setTimeout()`. JavaScript animations should always use the `requestAnimationFrame` API of the browser. Further information about `requestAnimationFrame()` can be found [here](#).

Guidelines for pictures & videos

- For an optimal image display on all devices, all images should be adapted for HDPI, if reasonable.
- In order to keep the data load of the advertising medium low despite HDPI, the formats Progressive JPEG / Progressive PNG, SVG or WebP (limited browser support) are suitable. However, the choice of format is not restricted and is at the developer's discretion.
- To deliver individual image versions to the correct device or pixel density, it is recommended to use CSS3 `image-set()` or 's `srcset` element (note browser support).
- Use server requests for smaller images sparingly. If you want to use many smaller image files (icons, logos), they should be loaded either as CSS sprites or directly in HTML as DataURI.
- Videos must be provided as H264/mp4, as well as VP8/WebM and must not exceed 5 MB. When integrating videos, use the Browser Video API.
- For videos in the advertising material, the sound may only be started after explicit action by the user (click on a "Sound on" button, etc.).

HTML5 on mobile devices

- For in-app advertising formats we support the standardized API MRAID (Mobile Rich Media Ad Interface Definitions) of the Interactive Advertising Bureau (IAB). In order to access device functions, such as the creation of a calendar entry, the ad media must be MRAID-compatible.

- Data from motion and alignment sensors of the mobile device may be read and used for the interaction of the advertising medium, but may not be used in a blocking manner. Functions such as closing or reducing the size of advertising media may not depend exclusively on the movement of the device.

Tracking when delivering physical HTML5

- The following must be observed when delivering physical HTML5:
- No 3rd party tracking and click commands may be integrated into the HTML file; these must be delivered separately and contain the necessary GDPR macros for the vendors used.

In-banner video playback on iOS devices

- Technical requirement All HTML5 creatives with in-banner videos must contain the playsinline attribute. This attribute ensures that in-banner video creatives do not switch to full-screen mode without user action. Important: The outdated webkit-playsinline attribute is not supported. Only the standardized playsinline attribute must be used.
- Implementation HTML video element The playsinline attribute must be added to every video element in the HTML code. It is a Boolean attribute that is activated simply by specifying it:

```
<video id="video1" controls="controls" poster="image.jpg" playsinline>
  <source type="video/mp4" src="video.mp4" />
  <source type="video/ogg" src="video.ogg" />
  <source type="video/webm" src="video.webm" />
</video>
```

Translated with DeepL.com (free version)

- YouTube Player When using the YouTube Player API, the playsinline parameter must be added:

```
<iframe id="ytplayer" type="text/html" width="720" height="405"
  src="https://www.youtube.com/embed/M7lc1UVf-VE?playsinline=1"
  frameborder="0" allowfullscreen>
</iframe>
```

- Additional notes for autoplay For autoplay functionality on iOS devices, the video must also be muted:

```
<video autoplay muted playsinline>
  <source src="video.mp4" type="video/mp4">
</video>
```