

ADITION

HTML5: technische Spezifikationen

Version 2021-09-01

Inhalt:

Einleitung und Limitierungen	2
clicktag-Implementierung	3
Code-Beispiele zum Einsatz von clicktags	4
Weitere Hinweise zu clicktags:	5
HTML5-Parameter	6
Verarbeitung von HTML5-Parametern im HTML-Code	6
HTML5 Responsive Ads: Adserver Anpassungen	7
HTML5 Expandable Ads: Creative Anpassungen	8
HTML5 Expandable Ads: Adserver Anpassungen	9

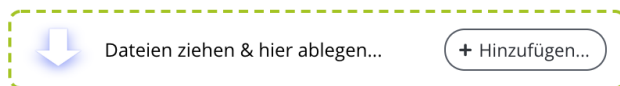
Einleitung und Limitierungen

Zur Verwendung von HTML5 Bannern in ADITION gibt es eine Reihe von Anforderungen, die berücksichtigt werden müssen.

Grundsätzlich müssen HTML5 Banner in Form von ZIP-Dateien (Archivdateityp, Dateiendung: .zip) angeliefert werden. Diese sollten alle für die korrekte Darstellung notwendigen Assets und Dateien beinhalten, mindestens aber eine HTML-Datei (Basis-Datei).

Die Ordnerstruktur, die innerhalb des ZIPs eingesetzt wird, wird dabei von ADITION übernommen und kann nach Hochladen des Archivs in ADITION betrachtet werden. Im Folgenden ein Beispiel dafür:

▼ Werbemittel-Upload



Entpackte Größe: 99,13kB

Name	Größe	Typ	Basis-Datei
▼ 🌐 Assets			
📄 custom_font.woff	21,41kB	WOFF	
📄 basic.css	6,85kB	CSS	
📄 manifest.json	0,37kB	JSON	
📄 custom_font_i.woff	29,30kB	WOFF	
📄 basic.js	10,79kB	JS	
📄 index.html	14,50kB	HTML	☒
📄 Fallback.gif	14,68kB	GIF	
📄 BannerProperties.xml	1,21kB	XML	

Weiterhin gelten folgende Limitierungen:

Maximale Dateigröße des ZIPs:	10 MB
Maximale Anzahl an Dateien und Ordnern im ZIP:	50
Erlaubte Dateitypen innerhalb des ZIPs:	*.jpeg, *.gif, *.png, *.swf, *.flv, *.mp4, *.js, *.txt, *.html, *.css, *.xml, *.bin, *.webm, *.ogg, *.ogv, *.svg, *.ico, *.bmp, *.eot, *.ttf, *.woff, *.woff2, *.htm, *.jpg, *.json
Maximale Anzahl an klickbaren Flächen (clicktags):	10 (clicktag, clicktag1, clicktag2 [...], clicktag9)

clicktag-Implementierung

ADITION erwartet bei klickbaren Flächen sogenannte „clicktags“ in Form von Javascript-Variablen. Werden keine clicktags verwendet, sondern die Ziel-URLs fest im HTML5 Banner vorgegeben, kann keine Klickzählung über ADITION erfolgen. Dies lässt sich über einen Klick in der Vorschau des Banners in ADITION prüfen. Bei korrektem clicktag-Einbau erscheint dort nach dem Klick eine zwischengeschaltete Seite mit folgender Meldung:



Code-Beispiele zum Einsatz von clicktags

Einfacher Hyperlink Tag:

```
<html>
<head>
</head>
<body>
  <a href="#clicktag" id="clicktag" target="_blank">
    <div id="example_element" style="width: 300px; height: 250px">
    </div>
  </a>
  <script>
    document.getElementById('clicktag').setAttribute('href', clicktag);
  </script>
</body>
</html>
```

Wichtig: der blau markierte Bereich „example_element“ repräsentiert den Code des eigentlichen Banners bzw. die Elemente des Banners. Dieser Teil sollte nicht mit kopiert werden falls Code aus dem Beispiel kopiert wird.

In diesem Beispiel wird das Banner von einem HTML-Anker-Tag eingeschlossen welches die Verlinkung vornimmt welche wiederum in einem nachfolgenden Script über die clicktag-Variable gesetzt wird.

Für unterschiedliche klickbare Flächen:

```
<html>
<head>
</head>
<body>
  <a href="#clicktag" id="clicktag" target="_blank">
    <div id="example_element1" style="width: 300px; height: 250px">
    </div>
  </a>
  <a href="#clicktag1" id="clicktag1" target="_blank">
    <div id="example_element2" style="width: 300px; height: 250px">
    </div>
  </a>
  <script>
    document.getElementById('clicktag').setAttribute('href', clicktag);
    document.getElementById('clicktag1').setAttribute('href', clicktag1);
  </script>
</body>
</html>
```

Wichtig: die blau markierten Bereiche „example_element1“ und „example_element2“ repräsentieren Flächen des Banners die klickbar sein sollen. Dieser Teil sollte nicht mit kopiert werden falls Code aus dem Beispiel kopiert wird.

Weitere Hinweise zu clicktags:

Die clicktags können nur vom System gefüllt werden, wenn diese in der Basis-Datei (HTML-Datei) eingesetzt werden. Werden clicktags z.B. in externen Javascript-Dateien eingesetzt, können diese nicht von ADITION mit Werten befüllt werden.

Die clicktags können nicht erkannt werden wenn sie in abweichender Schreibweise geschrieben werden, z.B. clickTag, Clicktag usw. Daher bitte clicktags immer in Kleinschreibung verwenden wie im Beispiel auf Seite 3 ersichtlich.

Es ist nicht notwendig, Javascript-Variablen zu deklarieren. Daher sollte auf Code-Snippets wie z.B. folgenden verzichtet werden:

```
<script>  
clicktag = "";  
</script>
```

Weiterhin dürfen clicktags nicht mit eigenen Werten befüllt werden:

```
<script>  
clicktag = "https://www.adition.com";  
</script>
```

Beide Fälle führen dazu, dass die von ADITION gesetzten Werte überschrieben werden und daher keine Klickzählung möglich ist.

HTML5-Parameter

Über HTML5-Parameter können Informationen in ADITION gepflegt werden welche in HTML5-Bannern verarbeitet werden. Dadurch lassen sich beispielsweise Textinhalte in ADITION vorgeben und ändern, ohne dass dafür die HTML-Dateien nochmals angepasst werden müssen.

Im Folgenden ein Beispiel für die Anlage von HTML5-Parametern in ADITION welche auch im weiteren Verlauf als Beispiele genutzt werden:

▼ HTML5-Parameter

+ HTML5-Parameter hinzufügen

Name	Wert	Akti...
headline	Dies ist eine Überschrift	 
textblock	Dies ist ein Beispieltext	 

<< < Seite von 1 > >> 

Anzeige Eintrag 1 - 2 von 2

Verarbeitung von HTML5-Parametern im HTML-Code

Im eigentlichen Bannercode (index.html des HTML5-Banners) stehen die HTML5-Parameter in Form von Javascript-Variablen zur Verfügung welche vom Adserver initial bei Auslieferung gesetzt werden. Der Variablenname ist dabei identisch zum Namen des HTML5-Parameters in ADITION.

Im Folgenden ein Beispiel zur Verwendung dieser Variablen:

```

<p id="textblock_element"></p>
...
<p id="headline_element"></p>
...
<script>
document.getElementById('headline_element').innerHTML = headline;
document.getElementById('textblock_element').innerHTML = textblock;
</script>
  
```

Ähnlich wie zuvor bei den clicktags angemerkt dürfen auch hier die Werte der Variablen nicht überschrieben werden.

HTML5 Responsive Ads: Adserver Anpassungen

Banner mit dynamischer Größe (responsive Ads) benötigen in ADITION zusätzlichen Code um sicherzustellen, dass das HTML5 iFrame von ADITION ebenfalls eine dynamische Größe erhält.

Hierfür kann nachfolgender Code genutzt werden.

```
<script type="text/javascript">
(function() {
  try {
    var iframe_id = 'html5_iframe_%timestamp%';
    var target = document.getElementById('ad_trgt_%timestamp%');
    var bannerIframe = target.getElementsByTagName('iframe')[0];
    bannerIframe.width='100%';
    bannerIframe.height='100%';
    bannerIframe.id=iframe_id;
    bannerIframe.style.position='absolute';
  } catch (exception) {
    console.log('Failed to make banner in div id=ad_trgt_%timestamp% responsive. Err:' + exception);
  }
})();
</script>
<script>var ad_vt_%timestamp% = {nodeId: "html5_iframe_%timestamp%"};</script>
```

Dabei sind folgende Punkte zu beachten:

- Dieser Code muss in ADITION im Feld „**Zusätzlicher HTML-Code dahinter**“ hinterlegt werden.
- Die Kreativagentur muss die Assets responsive / mit dynamischen Eigenschaften entwickeln, hierfür kann ADITION keinen Beispielcode liefern.
- Der Publisher muss Werbeplätze einsetzen welche entsprechende Flächen dynamisch zur Verfügung stellen.
- Je nach Layout und Setup der Publisher kann es notwendig sein folgende Zeile zu entfernen:

```
bannerIframe.style.position='absolute';
```

HTML5 Expandable Ads: Creative Anpassungen

Für expandierende Banner ist es notwendig eine Expansion oder Kontraktion des Creatives an den Adserver mitzuteilen, z.B. mithilfe von `postMessage`.

Hierfür kann beispielsweise folgender Code im Creative selbst eingesetzt werden:

```
var adname = 'myAD';

function expandAd() {
    window.top.postMessage('expandAd:::' + adname + ':::' + expandedwidth + ':::' +
        expandedHeight, '*');
};

function collapseAd() {
    window.top.postMessage(
        'contractAd:::' + adname + ':::' + collapsedwidth + ':::' + collapsedHeight, '*');
};
```

Wird dieses Beispiel genutzt sind folgende Punkte zu beachten:

- „myAD“ kann durch einen eigenen, aussagekräftigen Namen ersetzt werden, es ist hierbei lediglich wichtig, dass derselbe Name nicht von mehreren Bannern gleichzeitig auf der Zielseite genutzt wird.
- Die Methoden „expandAd“ and „contractAd“ können umbenannt und erweitert werden, die `postMessage` muss aber in ihrer Form beibehalten werden um mit dem nachfolgend vorgeschlagenen Adserver-Code zu kooperieren.

In ADITION selbst können die festgelegten Methoden dann aufgefangen und verarbeitet werden um Elemente wie das HTML5 iFrame anzupassen.

HTML5 Expandable Ads: Adserver Anpassungen

Sofern der zuvor vorgeschlagenen Code auf Creative Seite eingesetzt wird kann nachfolgender Code auf Adserver Seite genutzt werden:

```
<script>
window.top.ovk = window.top.ovk || {
  ads2Handle: {}
};
var ovk = window.top.ovk;
ovk.listenMessage = function(msg) {
  if (msg.data && msg.data.match(':::')) {
    var call = msg.data.split(':::');
    if (!ovk.ads2Handle[call[1]]) {
      ovk.walkFrames(call[1], window.top, msg);
    }
    ovk[call[0]](msg);
  }
};

ovk.walkFrames = function(adName, w, event) {
  var i, j, currentFrame;
  for (i = 0; i < w.frames.length; i++) {
    currentFrame = w.frames[i];
    if (event.source.window === currentFrame) {
      try {
        for (j = 0; j < w.frames.length; j++) {
          if (w.document.getElementsByTagName('iframe')[j].contentWindow ===
currentFrame) {
            this.ads2Handle[adName] =
w.document.getElementsByTagName('iframe')[j];
          }
        }
      } catch (e) {throw e;}
    }
    if (currentFrame.frames.length > 0) {
      this.walkFrames(adName, currentFrame, event);
    }
  }
};

ovk.expandAd = function(msg) {
  var call = msg.data.split(':::');
  this.origwidth = this.ads2Handle[call[1]].width;
  this.origHeight = this.ads2Handle[call[1]].height;
  this.ads2Handle[call[1]].style.width = call[2] + 'px';
  this.ads2Handle[call[1]].style.height = call[3] + 'px';
  if (this.ads2Handle[call[1]].contentWindow.parent !== window.top) {
    try {
      this.ads2Handle[call[1]].contentWindow.frameElement.style.width = call[2] +
'px';
      this.ads2Handle[call[1]].contentWindow.frameElement.style.height = call[3] +
'px';
    } catch (e) {}
  }
};

ovk.contractAd = function(msg) {
  var call = msg.data.split(':::');
  this.ads2Handle[call[1]].style.width = this.origwidth + 'px';
  this.ads2Handle[call[1]].style.height = this.origHeight + 'px';
};

(window.attachEvent) ? window.attachEvent('onmessage', ovk.listenMessage) :
window.addEventListener('message',
  ovk.listenMessage, false);
</script>
```

Wird dieses Beispiel genutzt sind folgende Punkte zu beachten:

- Dieser Code muss in ADITION im Feld „**Zusätzlicher HTML-Code davor**“ hinterlegt werden.
- Dies verändert nur die Größe der Adserver-Elemente von ADITION bei Expansion bzw. Kontraktion. Je nachdem, wie das Werbemittel auf Publisherseite ausgespielt wird und welchen Bezugspunkt es im Layout bekommt expandiert es nach links oder rechts. Hierauf hat ADITION keinen Einfluss.