



Technical Specification for Preparing HTML5 Creatives

www.gemius.com

HTML5 creative

Example of HTML5 creative

```
<html>
  <head>
  </head>
  <body>
    <a id="creativelink" target="_blank">
      
    </a>
  </body>
</html>
```

Click tags

In order for the creative parameters for measuring clicks to be sent, the advertiser should include the following JS function in the HTML5 creative just before </body>:

```
<script type="text/javascript">
  var parsed =
  (document.location.href.split('#')[1]||'').split('&');
  var params = parsed.reduce(function (params, param) {
    var param = param.split('=');
    params[param[0]] =
  decodeURIComponent(param.slice(1).join('='));
    return params;
  }, {});
</script>
```

The clickable place in a banner should be surrounded by <a> tags:

```
<a id="creativelink" target="_blank">
  
</a>
```

The ID of the element is important. If modified, it also needs to be changed in the succeeding functions.

Use the clickTag parameter to set href on the creative. The ID of the element (here: 'creativelink') should be the same as in the <a> tags surrounding the clickable place. This code should be placed after the JS function mentioned above but before </body>:

```
// change link href
document.getElementById('creativelink').href = params.clickTag;
```

Example after applying all of the above:

```
<html>
  <head>
  </head>
  <body>
    <a id="creativelink" target="_blank">
      
    </a>
    <script type="text/javascript">
      var parsed =
(document.location.href.split('#')[1]||'').split('&');
      var params = parsed.reduce(function (params, param) {
        var param = param.split('=');
        params[param[0]] =
decodeURIComponent(param.slice(1).join('='));
        return params;
      }, {});

      // change link href
      document.getElementById('creativelink').href =
params.clickTag;
    </script>
  </body>
</html>
```

IMPORTANT: This fragment depends on **document.location.hash**. It means that your creative must not depend on this property, at least not on document load.

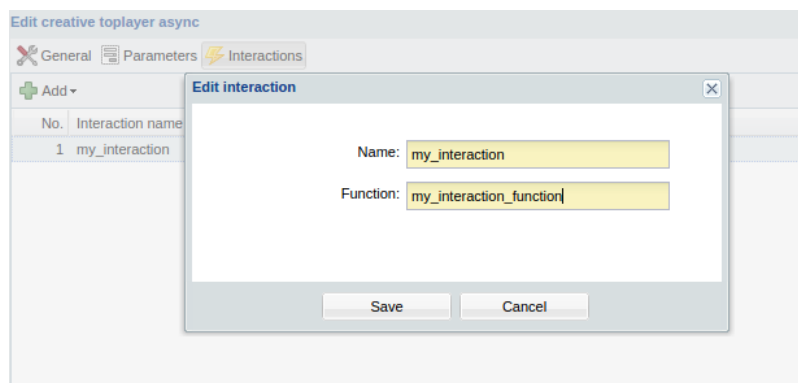
Multiple clicktags

In order to measure multiple clickTags you need to surround each clickable element with <a> tags, as described above, and add appropriate JS functions remembering that the IDs should be the same in the JS functions and <a> elements.

```
document.getElementById('creativelink1').href = params.clickTag1;  
document.getElementById('creativelink2').href = params.clickTag2;
```

Interactions

With the following interaction definition in the gDE interface:



use the following code to trigger an interaction in your HTML file. Remember to add an appropriate ID to the element you want to use as your interaction trigger. If you want to add more than one interaction, you should do it for each one of them.

```
document.getElementById('interaction-  
trigger').addEventListener('mouseenter', function() {  
    window.parent.postMessage('my_interaction_function', '*');  
});
```

Example code with interactions:

```

<html>
  <head>
  </head>
  <body>
    
    <a id="creativelink" target="_blank">
      
    </a>
    <script type="text/javascript">
      var parsed =
(document.location.hash.split('#')[1]||'').split('&');
      var params = parsed.reduce(function (params, param) {
        var param = param.split('=');
        params[param[0]] =
decodeURIComponent(param.slice(1).join('='));
        return params;
      }, {});

      // change link href
      var creativeLink = document.getElementById('creativelink');
      creativeLink.href = params.clickTag;
      // add interaction
      document.getElementById('interaction-
trigger').addEventListener("mouseenter", function() {
        window.parent.postMessage('my_interaction_function',
'*');
      });
    </script>
  </body>
</html>

```

IMPORTANT: The corresponding fragment is built into the creative code:

```

if (typeof window.addEventListener == "function") {
  window.addEventListener("message", function(message) {
    if (message.origin == "http://my.adocean.pl") {
      if (typeof window[message.data] == "function") {
        window[message.data] ();
      }
    }
  });
}

```

This is provided by the template of the creative.

Working templates

Currently, there 3 templates available that can deliver HTML5 file. These are:

- Toplayer [N] [ASYNC]
- Expand [N] [ASYNC]
- Billboard/Banner/Box [N] [ASYNC]

Creative assets

IMPORTANT: This limitation is temporary. It is valid until .ZIP file support is implemented.

There are several ways of attaching assets and additional JS files to a creative, e.g.:

- Base64 encoded images,
- Inline CSS, JS,
- Content Delivery Network,
- Use inliner (<https://github.com/remy/inliner>): It is a tool that inlines all assets of creatives; if you use Windows, please install NodeJS first (<https://nodejs.org/download/>).

Please pay attention to the number of requests required to show your creative. The lower the number, the sooner it will be visible to the end user.



Gemius S.A.

18 B Postepu Street

02-676 Warsaw, Poland

Phone: + 48 22 390 90 90

+ 48 22 378 30 50

Fax: + 48 22 874 41 01

contact@gemius.com

www.gemius.com