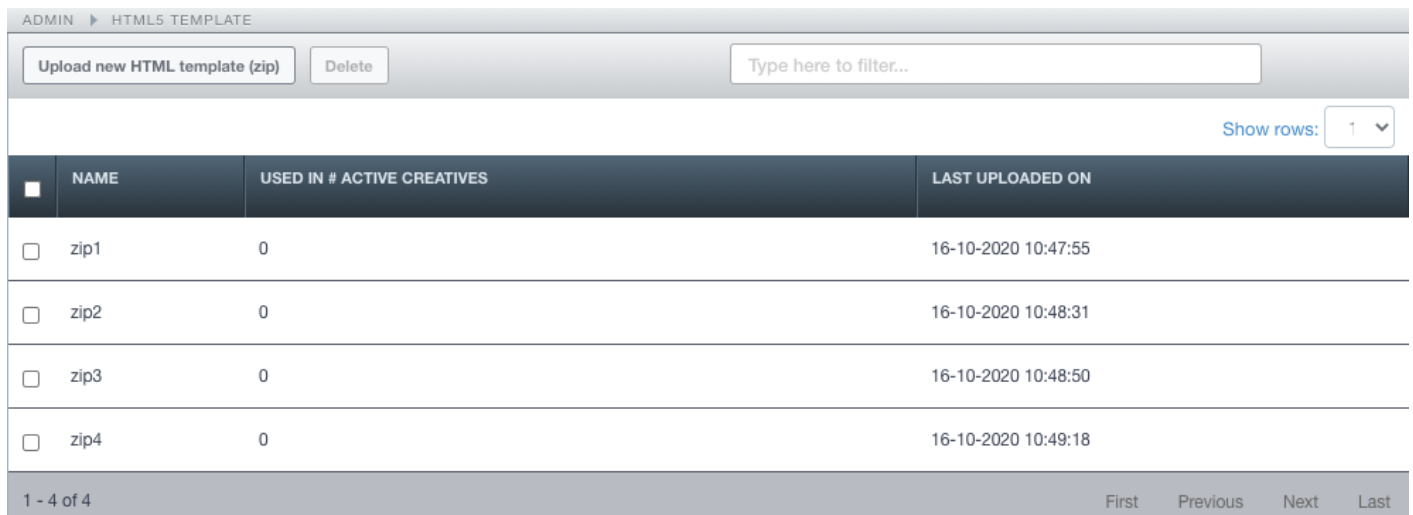


# HTML5 Templates

It is not possible to use Adhese macros such as `[adheseReplace:xx]` or `<ADHESE_X>` from within an HTML5 template. Unlike in regular (advar) templates, these macros will not work.

An HTML5 template is a combination of a regular [advar template](#) and a folder with HTML, CSS and JS files. It allows you to use the form functionality of an advar template while storing code and optional media in a folder structure.

In *Admin > Formats and templates > HTML5 templates*, you can find all previously created templates in zip format. When you click on the link, the following screen will appear:



The screenshot shows the 'ADMIN > HTML5 TEMPLATE' interface. At the top, there are buttons for 'Upload new HTML template (zip)' and 'Delete', and a search filter box labeled 'Type here to filter...'. On the right, there is a 'Show rows:' dropdown menu set to '1'. Below this is a table with the following columns: 'NAME', 'USED IN # ACTIVE CREATIVES', and 'LAST UPLOADED ON'. The table contains four rows of data, each with a checkbox in the first column.

<input type="checkbox"/>	NAME	USED IN # ACTIVE CREATIVES	LAST UPLOADED ON
<input type="checkbox"/>	zip1	0	16-10-2020 10:47:55
<input type="checkbox"/>	zip2	0	16-10-2020 10:48:31
<input type="checkbox"/>	zip3	0	16-10-2020 10:48:50
<input type="checkbox"/>	zip4	0	16-10-2020 10:49:18

At the bottom left of the table, it says '1 - 4 of 4'. At the bottom right, there are navigation links: 'First', 'Previous', 'Next', and 'Last'.

The following actions are available:

- Upload a new file: *click the Upload a new HTML5 template file (ZIP) and choose a file to upload.*
- Delete a template: *select the file and click the delete button.*
- Filter the list by entering a search term in the filter box.

## Using HTML5 templates

Previously created HTML5 templates can be used by clicking the 'Add advar' option in the creative tab and selecting the correct template in the 'Add Advar Template' dropdown list.

# Creating a new HTML5 template

Ensure that the filenames only contain alphanumeric characters, dots, and underscores. Other characters may cause issues when Adhese processes the file!

Steps for creating and uploading a new template:

1. Locally, create a new folder in which to store all files
2. Choose a name for the template: **[name].zip**
3. Create a **descr.json** file and store it in the main directory of your folder. The content has to be constructed in the same way you would create a regular advar template. (See step 6 of [creating a new advar](#))
4. Create an **index.html** file - which will be the starting point for the HTML5 ad - and store it in the main directory of your folder
5. Make sure to **incorporate clickthrough logic**. Adhese relies on it to track clicks correctly.

You can do this by adding the following line of code to your **index.html** file. Adhese will add its click tracking URL to this variable, after which you can use it in the rest of your clickthrough logic.

```
<script type="text/javascript">
  var clickTag = '';
</script>
```

6. Optionally you can store CSS, JS and images in separate files & folders within the main directory.

All links in the HTML5 creative, such as the link to an image within the ad, need to use a relative path, for example: `/graphics/ad-image.png` or ``. This enables the ad to be self-contained and, therefore, to run independently or to render without a network connection. External libraries and web fonts can be an exception to this guideline.

7. Zip the content of the folder (not the folder itself) and **change the name of the zip** to the name of the template
8. Upload the template in the Adhese UI under *Admin > Formats and templates > HTML5 templates*,
9. Test the template by creating a creative and checking the preview of the banner

## Retrieving text data from advar macro's

An advar template can contain fields that can be filled in while setting up a creative in the UI. Each of these fields has a macro that can be implemented in the index.html file. When the banner is served the macro's will be replaced by the text value provided through the creative.

The screenshot shows the advar creation interface. On the left is the 'Advar form' with fields for 'Product Category Filter', 'Product Brand Filter', and 'Product SKU Filter'. On the right is the 'Index content' code editor. A blue arrow points from the 'Product Category Filter' field to the macro `<CATEGORY>` in the code. A red arrow points from the 'Product Brand Filter' field to the macro `<BRAND>`. Another red arrow points from the 'Product SKU Filter' field to the macro `<SKU>`. A red text box at the bottom right states: 'Fields define the form through which the creative is set up and are used to capture custom campaign data'. A blue text box at the top right states: 'The macro's can be used in the index.html file and will be replaced by the inserted text'.

## Retrieving uploaded files

An advar creative can contain up to five extra files (count starts at 2) that will only be relevant for this specific creative. Templates that plan on using such files need to define the relevant fields in the description file:

```

"files": [
  {
    "default": "",
    "doc": "",
    "label": "Desktop fallback image",
    "key": "2nd"
  },
  {
    "default": "",
    "doc": "",
    "label": "Mobile fallback image",
    "key": "3rd"
  }
]

```

When served, each HTML5 creative will contain a **helpers** directory in which you can find **metadata.json**. This JSON can be used to determine if any uploaded files are available and what

their extensions are.

Example response:

```
{
  "ADHESE_2ND": "2nd.png",
  "ADHESE_3RD": "3rd.png"
}
```

The URL for metadata.json can be put together as follows:

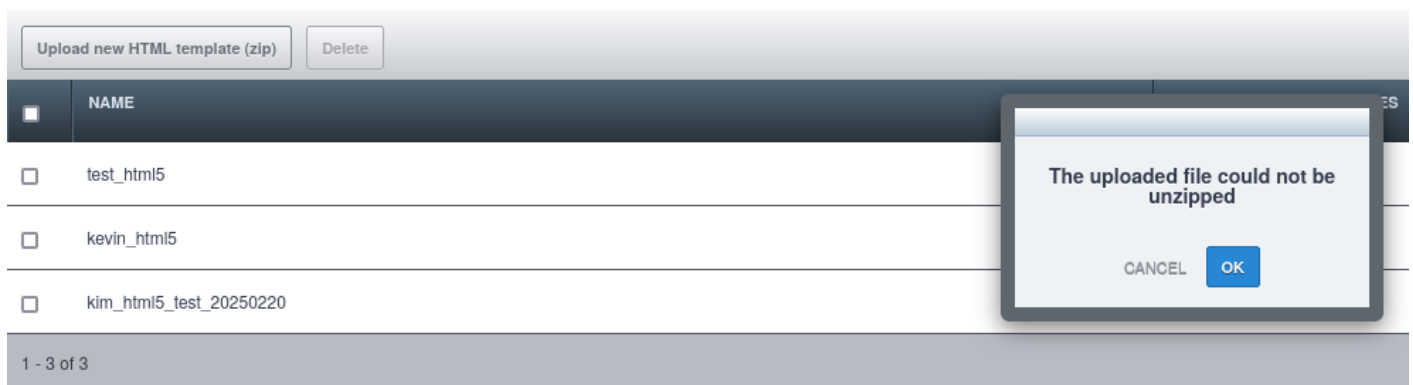
```
var metadata_url = location.protocol + '//' + location.host +
location.pathname.replace('/index.html', '') + '/helpers/metadata.json';
```

Each file will be stored in the same helpers directory. Once you know the extension of the file, you can put together the full URL:

```
var secondImage_url = location.protocol + '//' + location.host +
location.pathname.replace('/index.html', '') + '/helpers/' + '2nd.png';
```

## Troubleshooting HTML5 Templates

If an HTML5 template can't be uploaded for some reason, you'll get an error message:

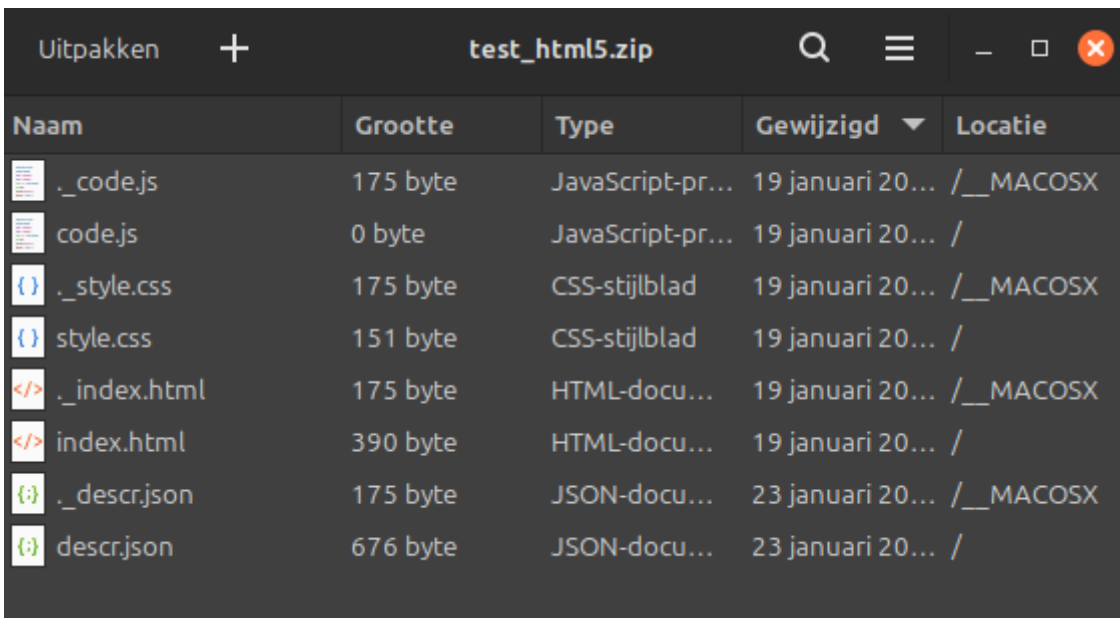
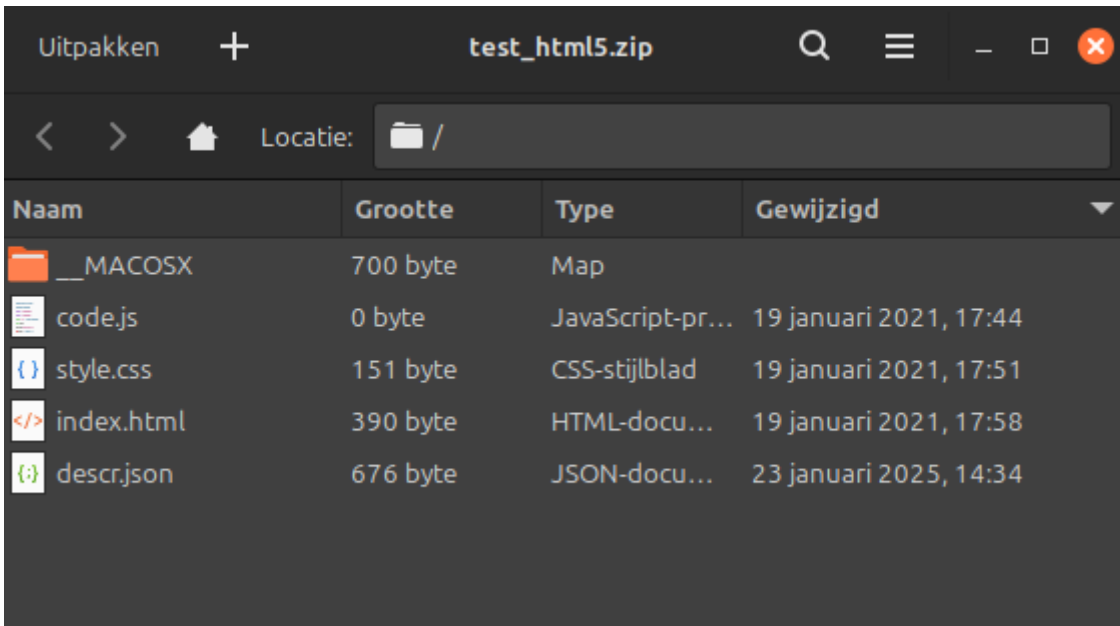


The screenshot shows a web interface for uploading HTML5 templates. At the top, there are two buttons: "Upload new HTML template (zip)" and "Delete". Below these is a table with a header "NAME" and three rows of templates:

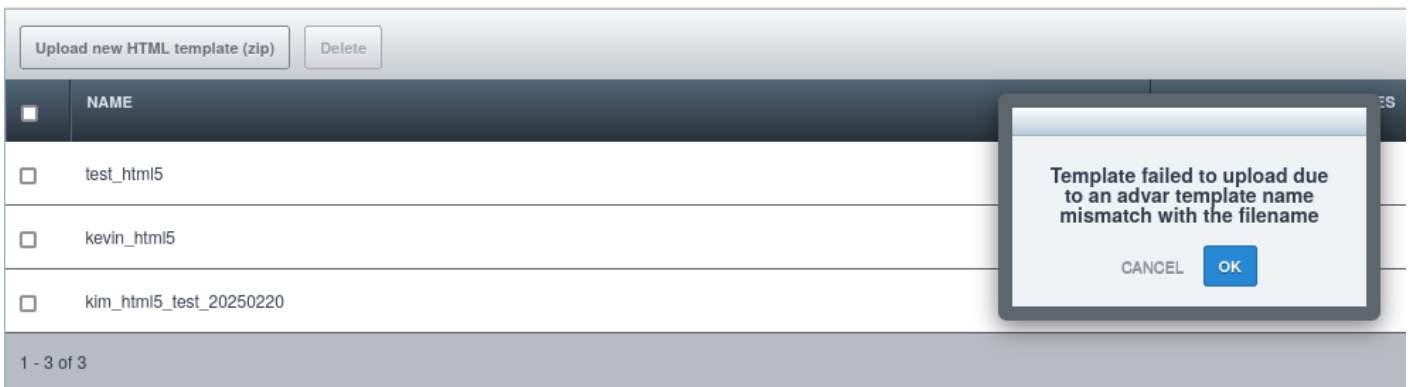
	NAME
<input type="checkbox"/>	test_html5
<input type="checkbox"/>	kevin_html5
<input type="checkbox"/>	kim_html5_test_20250220

At the bottom left of the table, it says "1 - 3 of 3". Overlaid on the right side of the table is a modal dialog box with the text "The uploaded file could not be unzipped" and two buttons: "CANCEL" and "OK".

In the case above, hidden files might block Adheses from properly unzipping the archive:



When creating an archive on Mac OS, a hidden `__MACOSX` folder might be created. These files must be removed for the upload to succeed (you may need to unzip, delete and re-zip the files).



In the case above, the zip file has been renamed. If the zip file does not match the advar name in the descr.json file, rename the zip file to match the advar name in the descr.json file, and the archive can be uploaded.

---

Revision #13

Created 26 February 2025 13:46:13 by Casper Steuperaert

Updated 14 November 2025 15:30:51 by Kim Van Crombrugge