3rd-Party Creatives & Troubleshooting

Information on various third party creatives and troubleshooting tips

- Gemius Creatives Troubleshooting
- Troubleshooting Guide for VAST Campaigns
- Google Web Designer (GWD) Banner Creation
- DoubleClick Creatives

Gemius Creatives Troubleshooting

How to change "Gemius" creatives click_url macro

If the client sends Gemius creatives, we need to modify them, deviating from Gemius' instructions.

N/B: The {CLICK_URL} Macro in Gemius creative JS tags should be automatically added/updated after uploading/saving, but it's good to always re-check and confirm.

Original Gemius creative example:

```
<script language="javascript1.2" type="text/javascript">
//<![CDATA[
document.write('<scr'+'ipt src="https://hrgde.adocean.pl/_'+(new
Date()).getTime()+'/ad.js?id=6O.qlp3G489VMXJZaa4nI1Lk7kAq6ZtA16tbRx8dxt3.m7/nc=0/gdpr=0/gdpr_consent
=/redir=" language="javascript"></scr'+'ipt>');
//]]>
</script>
```

How to change it:

1. Include **{click_url}** as a new variable in a separate script tag:

```
<script>
var click = '{CLICK_URL}';
</script>
```

2. Then add **click** variable in the end of Gemius tag as shown:

```
<script language="javascript1.2" type="text/javascript">
//<![CDATA[
document.write('<scr'+'ipt src="https://hrgde.adocean.pl/_'+(new
Date()).getTime()+'/ad.js?id=6O.qlp3G489VMXJZaa4nI1Lk7kAq6ZtA16tbRx8dxt3.m7/nc=0/gdpr=0/gdpr_consent
```

```
=/redir=' + click + '" language="javascript"></scr'+'ipt>');
//]]>
</script>
```

The full JS tag should appear in the following manner:

```
<script>
var click = '{CLICK_URL}';
</script><script language="javascript1.2" type="text/javascript">
//<![CDATA[
document.write('<scr'+'ipt src="https://hrgde.adocean.pl/_'+(new
Date()).getTime()+'/ad.js?id=6O.qlp3G489VMXJZaa4nI1Lk7kAq6ZtA16tbRx8dxt3.m7/nc=0/gdpr=0/gdpr_consent
=/redir=' + click + '" language="javascript"></scr'+'ipt>');
//]]>
</script>
```

Troubleshooting Guide for VAST Campaigns

How VAST works:

We create VAST tag with one of the viewability measurement option (VPAID or OMID) and we share with MOAT(or any 3rd party) which then wraps it into VAST tag with either VPAID or OMID or both measurement options in one tag. When we add creative into DSP when we receive from 3rd party, we start campaigns and we then:

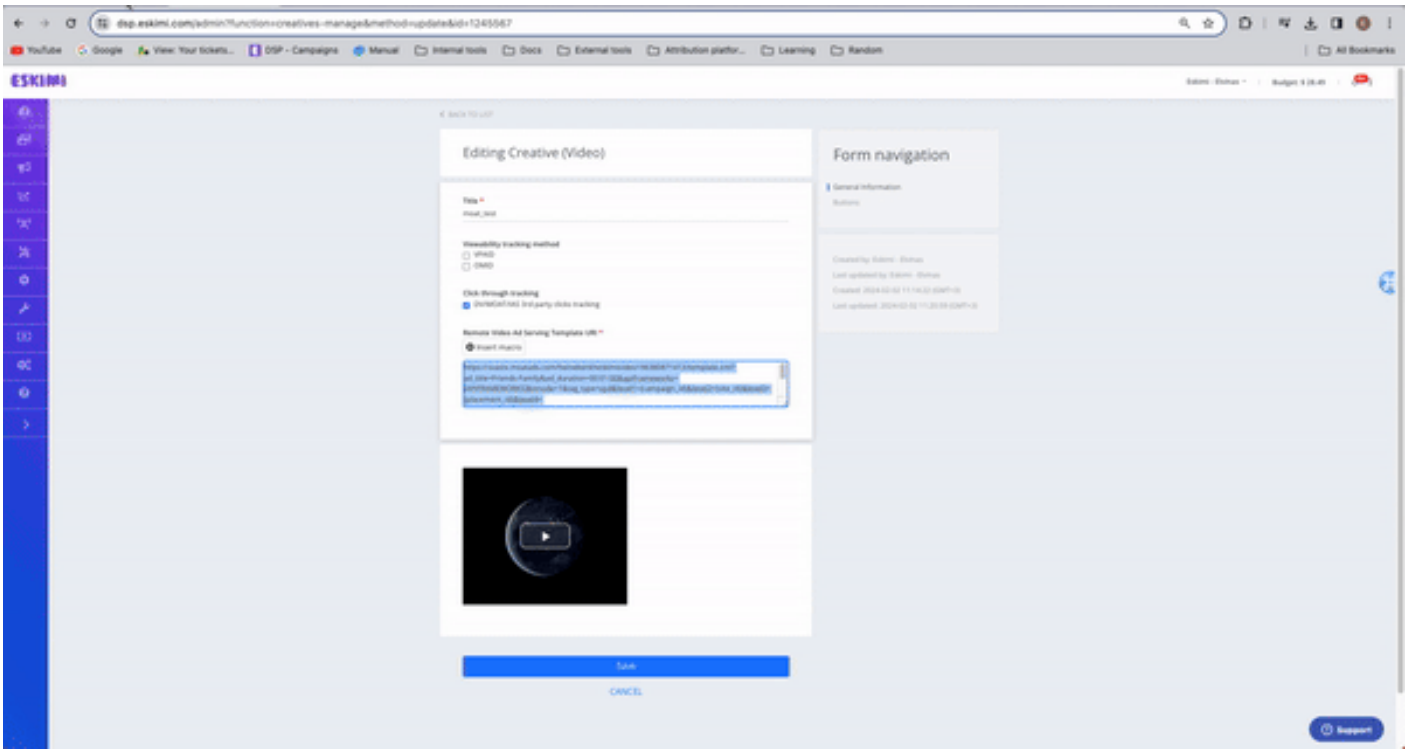
1. Try to create different version (vast2, vast3, vast4) of the creative, to cover more bid requests (if creative version is vast2 all will be created, if version is vast3 we will create vast3 and vast4, if creative is vast4 only vast 4 will be used)

2. Check each bid request, what vast version and measurement option (VPAID/OMID) this bid supports our bidders then select which creative to take to match the bid request information

3. We bid only on "matching" bid request and then get info why we didn't won or we get an impression What to look in creatives: - When sending creative for 3rd party, I suggest not to add any viewability measurement option - this will be less tricky, to get two different measurement (OMID and VPAID). We can add it later, when we get wrapped creatives. - What viewability measurement is used in "received wrapped creative" - search VPAID or OMID - and you will know which is used - If we want more reach - we need creative with OMID instead VPAID (best two separate creatives(in most cases, but could be both in one creative)).

How to check VAST XML:

VAST XML contains all in the information regarding the VAST video you're running, to open it you just need to take the VAST URI and paste it your browser:



Vast Versions:

If you notice that some of VAST campaigns are struggling with spending, it might be that it's because the VAST is running on certain version, most likely **VAST 4**, most of vast traffic comes from **VAST 2** and **VAST 3**

To check witch VAST version is being used, you need to open pre-wrapped VAST XML, and in the top you will see the vast version :

```
<VAST version="2.0">
```

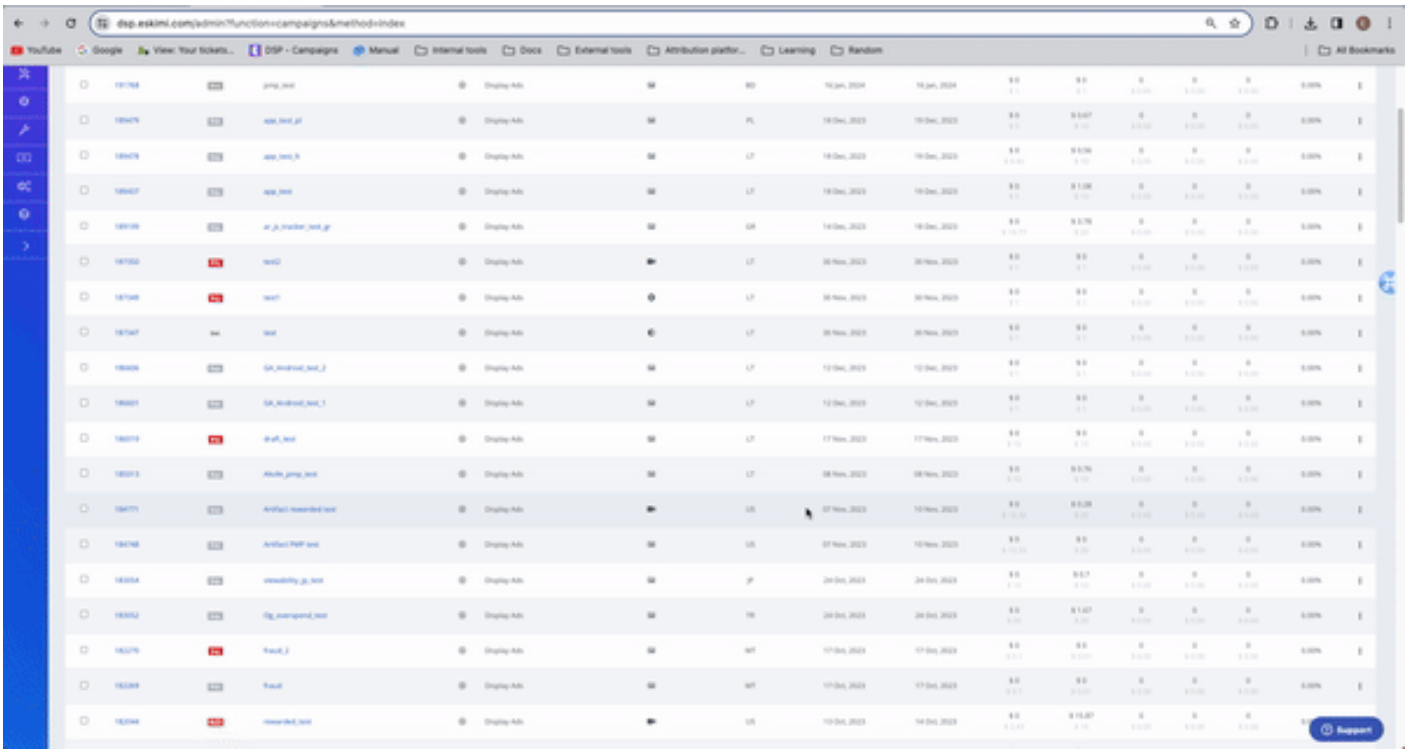
Usually, the best scenario is when the client shares tags for all VAST versions, but if you get only 1 tag from the client, and it's VAST 4, you can manually change the version by adding **&vastVersion=3** or **&vastVersion=2** in the end of the tag, for example:

```
https://unified.adsafeprotected.com/v2/1598445/73100257?omidPartner=[OMIDPARTNER]&apiframeworks=[APIFRAMEWORKS]&bundleId=[BUNDLEID]&xsId=[PLEASE_IMPLEMENT_UNIQUE_ADSERVER_IMPRESSION_ID_HERE]&ias_xappb=[ctv_appid]&originalVast=https://dsp-media.eskimi.com/vast?e=0lmb8oMQVJj0yEnGgCyv80W1t7OatF8Pm7Zn7si7J0JT9a5Egr2JFN7H8Y7ugeyBsppmSqkKmTlCU8lbQm7g83l6Z2hxb2pha2xjbm1ydHU&vastVersion=3
```

Wrapped tags:

1. Create campaign **A** with the original VAST video, set the campaigns date to the actual campaign date that will run with the wrapped tags.

2. Go to the campaign approval page, copy the VAST URI and send it to the client to wrap (Do not approve the actual campaign, leave it as Pending).

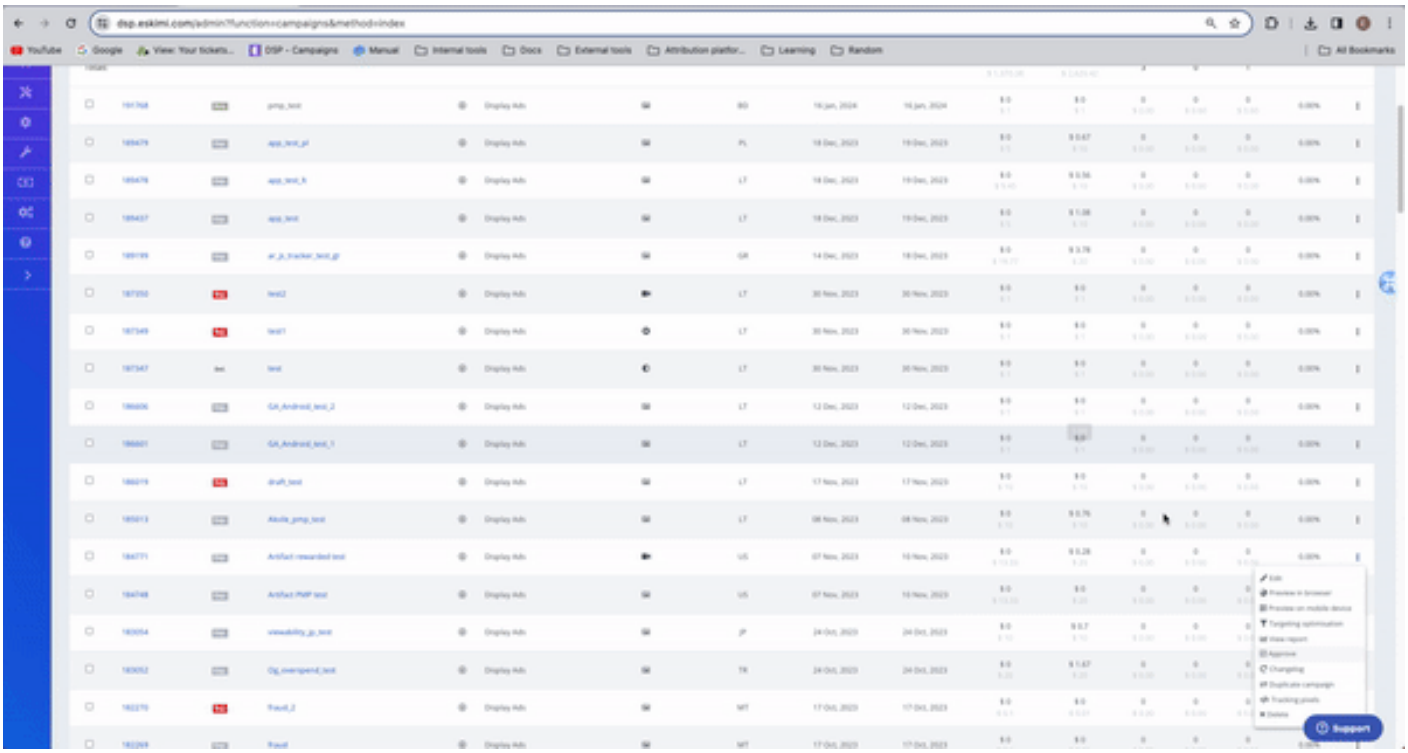


3. When you receive the wrapped tags, create an actual campaign **B** that will run with those tags.

4. Leave the original campaign **A** as Pending until the campaign **B** ends - this is important, as if the campaign A (the original campaign from where we took the tag that was sent to the client to wrap), get expired - the actual campaign will stop spending.

5. (Optional).If the original campaign **A** gets expired before the campaign **B** ends, you can create a new campaign **C**, but it has to have the same creative settings as the original campaign **A** had before we sent it for wrapping (If the original campaign did not have any OMID/VPAID parameters checked, the new one also needs to have them unchecked for example. If the original creative had only VPAID selected, then the new one should have it selected as well and etc). Then from the new campaign **C** we take the new VAST URL from the campaigns approval page and change it in the VAST URI:

```
&originalVast=https://dsp-media.eskimi.com/vast?
e=0lmb8oMQVJj0yEnGgCyv80W1t7OatF8Pm7Zn7si7J0JT9a5Egr2JFN7H8Y7ugeyBsppmSqkKmTlCU8lbQm7g83l6Z
2hxb2pha2xjbm1ydHU
```



Filtering reasons:

Here are the main filtering reasons for VAST campaigns you can see in the stattools, and the explanations for each:

video_max_duration	Filtering: Video ad max duration exceeded
video_vast_version	Filtering: Video ad VAST version is not supported
video_mime	Filtering: Video ad video file type is not supported

video_max_duration - Video is too long (can't really control this part from our part).

video_vast_version - As mentioned previously, campaign is probably running on only 1 vast version or only using VAST 4.0, need to change the version or add additional ones.

video_mime - Wrong creative settings (For example campaign is using OMID but we do receive traffic for OMID).

Google Web Designer (GWD) Banner Creation

There are a few requirements that have to be met when you are creating a banner using Google Web Designer (GWD). This article will help you to understand those requirements and will allow easy deployment of your created banners.

Requirements:

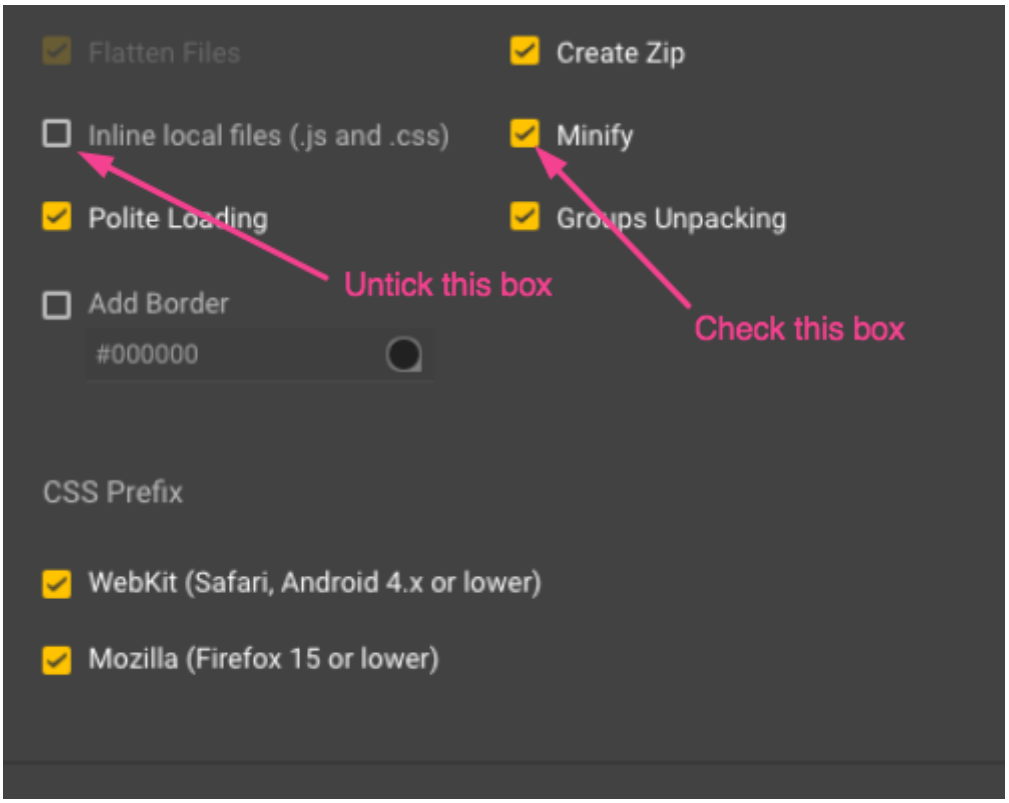
1. The target window for the click-through URL must be set to **"blank"** so the click-through will open in a new window. Do not leave the target statement undeclared. Declare this as if you are doing the click-through HTML tags. **Most importantly the banner should be clickable!**
2. If you are defining the click use the **window.open()** function, if you are making the click as a JavaScript solution.
3. All creatives made with GWD should contain our click URL macro. So the banners landing page should be set as **{CLICK_URL}. This is very important, as we take the banners landing page based on the campaign.**

4. When publishing the creative from GWD, two settings have to be changed:

- Uncheck**: In-line local files** and
- Check**: Minify**

The main creative file (index.html) should contain no more than 10,000 symbols. This is because part of the exchanges only supports this limit of symbols in bid response.

See the screenshot below to see how your publishing settings screen should look like.



After publishing you should get a ZIP file.

5. Consider this as a TIP: when making animated elements, it is best practice to turn the whole animated creative to a .GIF image file, this is to reduce the creative size and reduce the loading time.

DoubleClick Creatives

Most of the times doubleclick ad creatives are without click_url placeholder, example:

```
<SCRIPT language='JavaScript1.1'
SRC="https://ad.doubleclick.net/ddm/adj/N424802.280377IRCTC/B32309828.400403988;
abr=!ie;sz=300x250;ord=[timestamp];dc_lat=;dc_rdid=;tag_for_child_directed_treatment=;tfua=;dc_sdk_apis=[
APIFRAMEWORKS];dc_omid_p=
[OMIDPARTNER];gdpr=${GDPR};gdpr_consent=${GDPR_CONSENT_755};dc_tdv=1?">
</SCRIPT>
```

In such case in the end of the JS tag you need to **click={click_url}**, example:

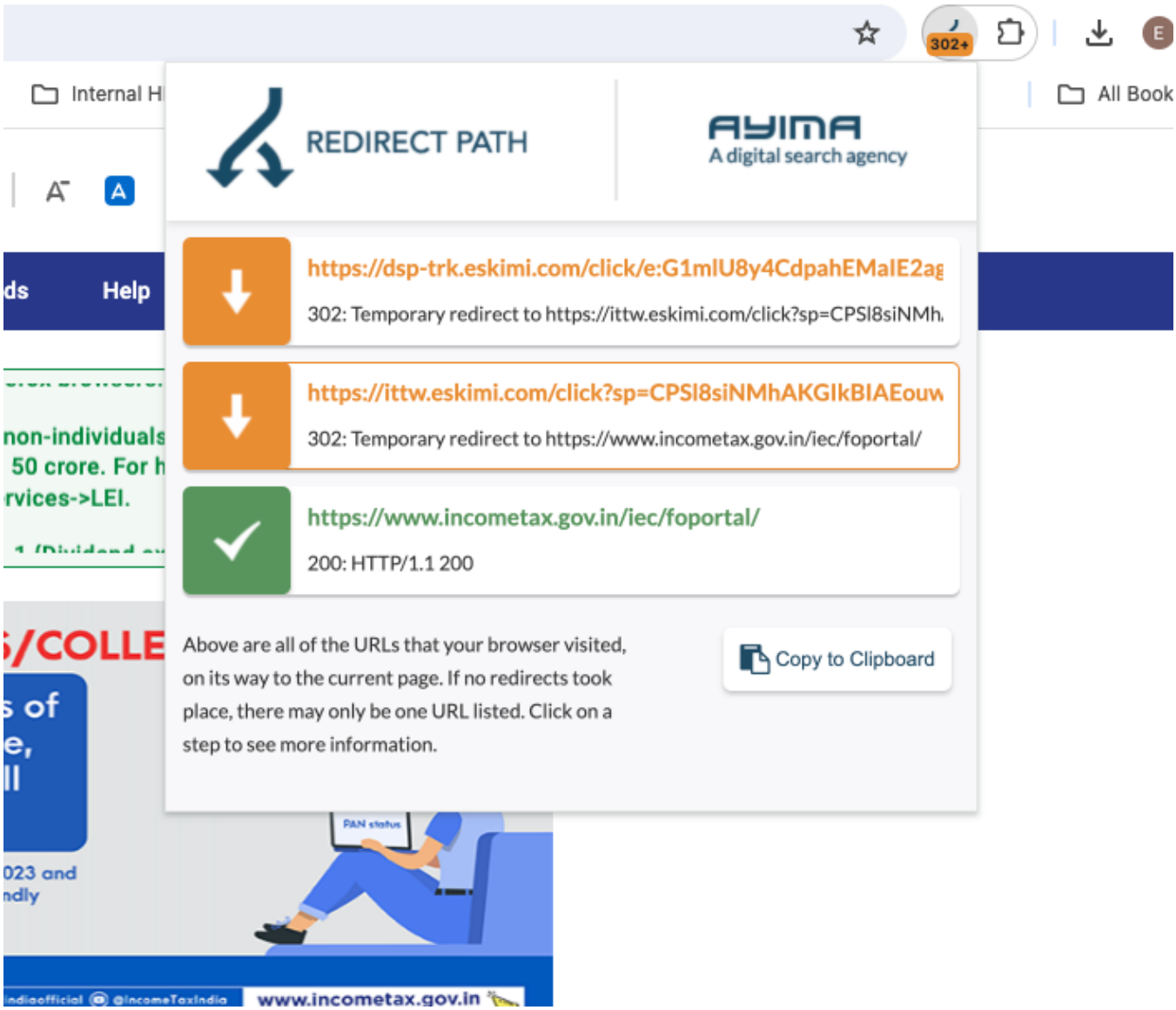
```
<SCRIPT language='JavaScript1.1'
SRC="https://ad.doubleclick.net/ddm/adj/N424802.280377IRCTC/B32309828.400403988;abr=!ie;sz=300x250;or
d=
{CACHEBUSTER};dc_lat=;dc_rdid=;tag_for_child_directed_treatment=;tfua=;dc_sdk_apis=[APIFRAMEWORKS];dc
_omid_p=[OMIDPARTNER];gdpr=
{GDPR};gdpr_consent={GDPR_CONSENT};dc_tdv=1;click={click_url}">
</SCRIPT>
```

Also, it's very important to remove **$** before the **GDPR** macros, as they break them witch causes any tracking to break as well (Also **{GDPR_CONSENT_755}** macro has to be changed to { **GDPR_CONSENT}**).

After these changed always validate if the redirections happens correctly:

1. Get Redirect path extension for your browser.
2. Launch a test campaign with the updated tag on https://display.eskimi-creatives.com/ (Eskimi Standar exchange, only for Display And IBV).
3. Catch the ad live and click on it.

4. After the redirections open the Redirect Path extension and check the redirection path:



5. If it goes through Eskimi server, it's working as expected.
6. Double check the tests campaigns report to see if the clicks are really being collected.