Instructions on integration with the Actionpay affiliate network through Google Tag Manager

Versão 7.3 - 12/09/17

SUMÁRIO

STARTING

Before starting the integration make sure you have:

- Partner_id and Goal ID – obtained from Actionpay account manager;
- Have a link to your campaign – obtained from the Actionpay account manager;
- Utm analytical tags and information on target actions – obtained from the account holder on the part of the advertiser (you) or in the contract appendix;
- Commission window (Post-click period) – Obtain from the person responsible for the account (contract);
- Information on how the tracking of the last paid traffic source is performed on your side (last cookie expires).

1. TAG ACTIONPAY

The following outline of work consists of three parts and is acceptable for most campaigns. The typical example used here is for an “online” store. However, campaigns may have individual peculiarities or specific techniques that require customization. If it is not possible to use the standard procedure, other integration alternatives are possible, contact our team to verify alternatives. Our contact details are at the end of this manual. The default integration consists of three main parts:

1. Process the transaction between Actionpay's network and the advertiser's website;
2. Notification of Actionpay's network about the action performed (pixel);
3. Notification of the Actionpay network about the change of status of the previously executed action (optional) – Executed outside the GTM.

Each part is described in detail hereafter.

1.1. PROCESSING THE ACTIONPAY NETWORK TRANSITION (TAG ACTIONPAY)

To identify that the transition to the advertiser's website was made through Actionpay's network rather than another source. There are two alternative forms of identification: the full form and the compact form. It is up to the advertiser to decide which of the two ways to use, then it is necessary to inform the account manager, or integration manager, of the decision to configure the system. Links of this type will be used for your campaigns by our affiliates.

http://apycomm.com/click/53b29c288b30a8c0598b458b/81092/67345/

The user who follows this link will, for example, see the following URL in the browser's address bar, in the full format:
http://www.test.ru/?actionpay=27606ca9-b7ee-c96f-ef18-01493ec212cf.67345&utm_medium=actionpay_BR

These variables are:

utm_medium = Origin of traffic

actionpay = Identification of transition and source

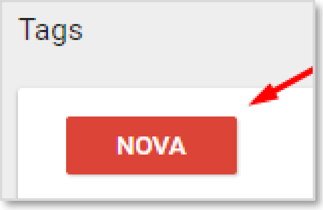
The transition and font identifiers, which are indicated in the example as “0”, are represented by alphanumeric sequences. The maximum length is 50 characters. You can add various Utm parameters for your internal tracking and tracing. Please inform the integration team which parameters should be added.

For exemple: **utm_medium=actionpay_BR**& utm_campaign=actionpay

1.1.1. Add URL parameters in Cookies

It is necessary to store the values of the GET parameters in the users' cookies to be able to identify the user when they return and also for the correct functioning of the tracking. The lifetime of access to the “cookie” must correspond to the period of the “Post-Click” or commission window of the campaign, a value specified in the annex to the contract (for example, if the commission window is equal to 30 days, the cookie will be active until the moment which is a sum of the transition date plus 30 days). The Actionpay network will automatically test the correct setting of cookies (correct name, meaning and validity) periodically. Along with the cookie, the advertiser's Website may save the parameters received during the transition of Actionpay's network with its internal systems (database, user profile and others). However, adjusting the cookie is critical. There are a few ways to set a cookie in Google Tag Manager. The example below is just a suggestion. In this example we are using the compact link, if you want to use the full one, do not forget to inform our integration team to change the link. The next step is to create a custom HTML TAG to create the actionpay cookie when traffic comes from our affiliates.

Figure 1: New Tag

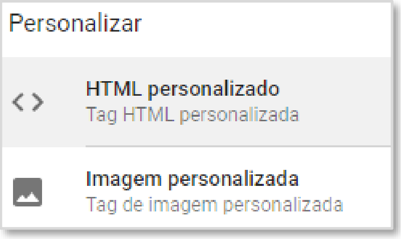


Go to Tag and select NEW.

1. Give the Tag a name, for example: **Tag_Actionpay**

2. Choose the type of Tag: HTML personalizado.

Figure 2: HTML personalized



3. In HTML write the following Java script code

```
<script language="JavaScript" type="text/javascript">
    var sourcevar = getQueryVariable("utm_medium");
    var apclickvar = getQueryVariable("actionpay");
    var postclick = 30; // Postclick
    var domain = "exemplo.com"; // Domain without www.
    var partnerId = "12345"; // PartnerId
    function getQueryVariable(variable) {
        var query = window.location.search.substring(1);
        var vars = query.split("&");
        for (var i = 0; i < vars.length; i++) {
            var pair = vars[i].split("=");
            if (pair[0] == variable) {
                return pair[1];
            }
        }
    }
    function setCookie(cname, cvalue, exdays) {
        var d = new Date();
        d.setTime(d.getTime() + (exdays * 24 * 60 * 60 * 1000));
        var expires = "expires=" + d.toUTCString();
        document.cookie = cname + "=" + cvalue + "; path=/;" + expires + "; domain=" +domain;
        return cookieap;
    }
    if (sourcevar == "actionpay_BR") {
        var cookieap = setCookie("Actionpay", apclickvar, postclick);
        var cookieap = setCookie("Origem", "actionpay", postclick);
    } else if (sourcevar != undefined) {
        var cookieap = setCookie("Origem", sourcevar, postclick);;
    }
    (function() {
        var s = document.createElement('script');
        s.type = 'text/javascript';
        s.async = s.defer = true;
        s.src = '//aprtn.com/code/' + partnerId + '/';
        var p = document.getElementsByTagName('body')[0] ||
            document.getElementsByTagName('head')[0];
        if (p) p.appendChild(s);
    })();
</script>
```

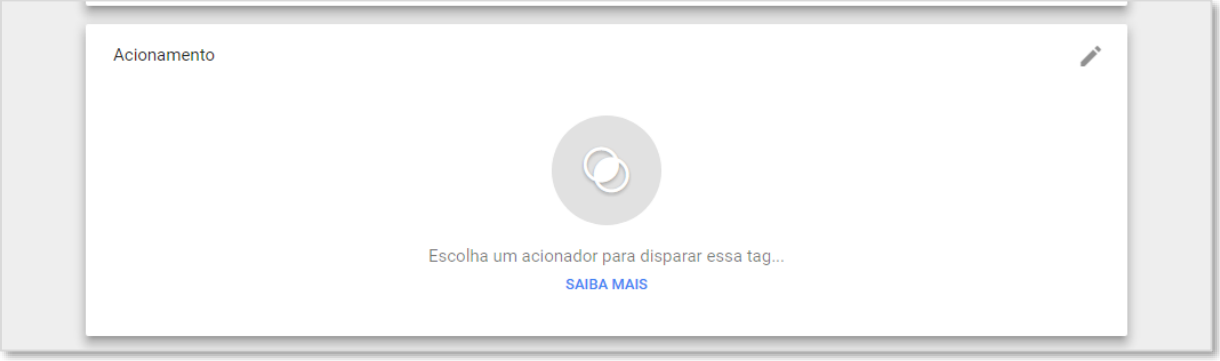
You must obtain the Parter_id from your account manager. Please remember to change example.com to your domain without www. That is, if your site is "www.exemplo.com.br" the domain will be "example.com.br". You must change the marked values of 30, for the number of days of your commission window, (postclick period), this value is defined in the contract with Actionpay. The example code 1 on the previous page will create two different cookies, the Actionpay cookie=0000-0000-0000-0000.0000000 and the Origin cookie=actionpay. In addition to carrying out a traffic check, preventing fraud and optimizing Actionpay campaigns.

Select Support to document.write

Choose the Escolha o trigger for activating this tag

The next step is to define a triggering rule for the Tag within the newly created Tag_Actionpay.

Figure 3: Rule of drive



Escolha um acionador para disparar essa tag...

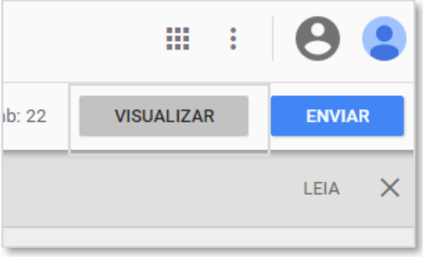
Select All Pages, a rule that is already pre-configured in tagmanager.

1.1.1.1. Last paid source management

The second cookie created (Origin) in the code above is the way to track the last paid source. If you use another cookie to manage this, just change the word Origin in the code to the name of the cookie used by you, also changing the value of "actionpay" by the code that represents our network. If you already have another Tag to manage the lastclick, you can skip this part.

1.1.1.2. Testing the cookies created

Figure 4: Visualize Debugger



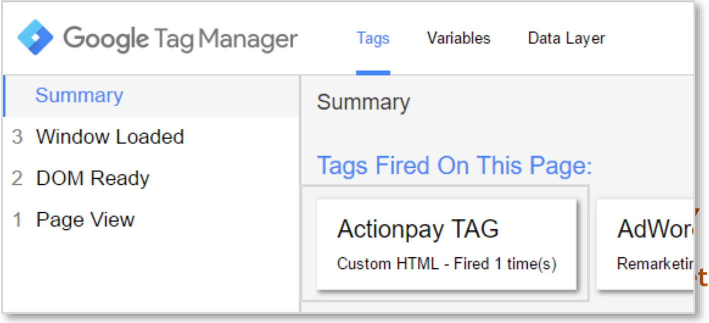
With the TAG created with its respective rules and variables, publish the container to perform the test.

Using the Google Chrome browser, enter Preview and select Preview and Debug as shown in Figure 4.

This command will simulate the tag inside your site. Go to your website, below will open the tagmanager debugger. The Tag_Actionpay tag must be fired, on all pages. If it hasn't been fired, check the Tag's firing rules.
Then, in the same window created by the debug command, copy and paste the link generated by your account manager. The link should be similar to the link below, pointing to your website. Check if the TAG has been triggered.

Example:
http://apypxl.com/click/54501e6b8b30b837748b4567/00000/70505/subaccount
(this example link doesn´t work!!)

Figure 5: Tag Actionpay Triggering

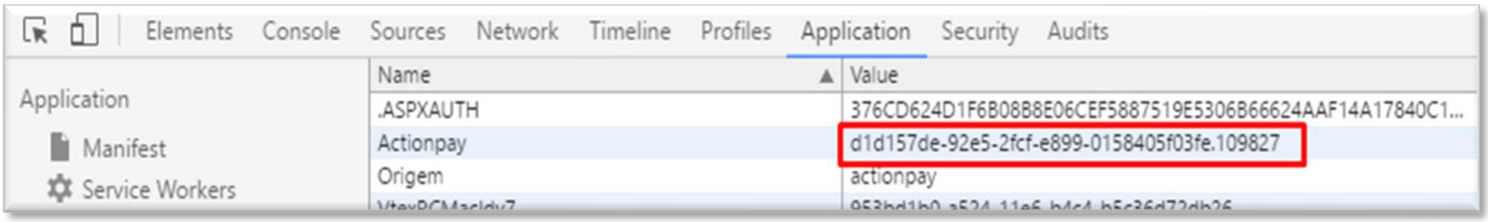


tivo.

Select Chrome development mode (f12) in Application / Cookies / (your website) field

Go to Application/(your domain) and check if the Actionpay cookie was created correctly by copying the apclick and apsource data into the cookie. Also check that the Origin cookie is equal to “Actionpay”. Both cookies must expire as set in the Campaign Commission Window.

Figure 6: Chrome developement tab; Application



Name	Value
.ASPXAUTH	376CD624D1F6B08B8E06CEF5887519E5306B66624AAF14A17840C1...
Actionpay	d1d157de-92e5-2fcf-e899-0158405f03fe.109827
Origem	actionpay
VtexRCMacIdv7	953bd1b0-a524-11e6-b4c4-b5c36d72db26

1.2. NOTIFICATION ON THE EXECUTION OF THE ACTION (PIXEL)

The Tracking-pixel, or tracking pixel, (hereinafter only “pixel”) is used to inform Actionpay's network about the action performed on the Advertiser's Site. In other words, at the moment the action is being performed; a code that loads a 1x1 pixel sized transparent image from the Actionpay server will be substituted into the code shown to the page user.

This image is not shown to the user but serves as a way of requesting the Actionpay server to save the action parameters. There are other forms of integration that can be used if the installation of the pixel is not feasible. To find out more, please contact our support team. See our contacts in item 2 of this manual.

The pixel code with the transmission of parameters is similar to:

Code 1: Example of pixel code

```
<img
src="https://apretailer.com.br/ok/0000.png?actionpay={{cookie_actionpay}}&apid={{ID_Transa
cao}}&price={{Valor_Transacao}}&width=1&height=1" width= "1" height= "1"/>
```

Where:

0000 - It´s the identifier of the action on the Actionpay network (Id da Meta)

actionpay= 0000-0000-0000-0000.0000000 it´s the identifier of the transition.

apid=NNNN **If there is a unique identifier assigned to each action performed on the advertiser's side, this identifier must be transmitted to the pixel with the help of the “apid” parameter.**

The unique identifier can be, for example:

- Order number
- Order number

- User identifier
- Payment number
- Action identifier created in the advertiser's system
- Etc.

Any sequence of alphanumeric symbols with punctuation marks but no spaces can be an identifier.

An identifier is important so that actions can be compared in the future and named as approved or cancelled.

price= 000.0000 **- If any amount of money is assigned to each action performed on the advertiser's side, this amount must be passed to the pixel with the help of the "apprice" parameter.**

This amount can be, for example:

- An order/buy amount;
- A payment amount;
- A replenishment (recharge) amount;
- Etc.

In this case, the unit of amount exchanged can be in any currency including virtual currency (eg paypal). In a number with a fractional part, the decimal symbol must be a period. (American model). The maximum number of characters after the period is 4.

If there is no VALUE assigned to the action, as in the example of a registration, this parameter can be excluded from the pixel.

It is important that the pixel is only called when the action originates from Actionpay, so the control of the last click is essential. If it is impossible to track the last click, it is possible to work without it, however, your offer will be ranked worse by the affiliates due to the rejection rate of the actions.

1.2.1. Pixel configuration in GTM

1.2.1.1. Variables to capture the pixel

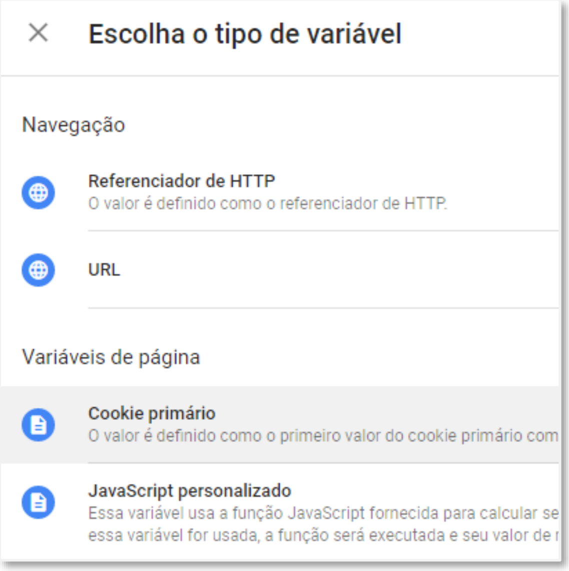
Before creating the TAG pixel, four new variables must be created.

The first two are First-party cookie variables as shown in the example below.

Figure 7: Types of Variable to feed the pixel

1. **Cookie_actionpay: to read the data registered in the Actionpay cookie created in the previous step. Go to Variable, create variable, name the variable Cookie_actionpay, select the type of first-party cookie variable, in Cookie name write "Actionpay".**

2. **LastClick_control: to read the origin cookie created in the previous step. Go to Variable, create variable, name the variable LastClick_control, select the type of first-party cookie variable, in Cookie name write "Origin".**



Escolha o tipo de variável

Navegação

Referenciador de HTTP
O valor é definido como o referenciador de HTTP.

URL

Variáveis de página

Cookie primário
O valor é definido como o primeiro valor do cookie primário com

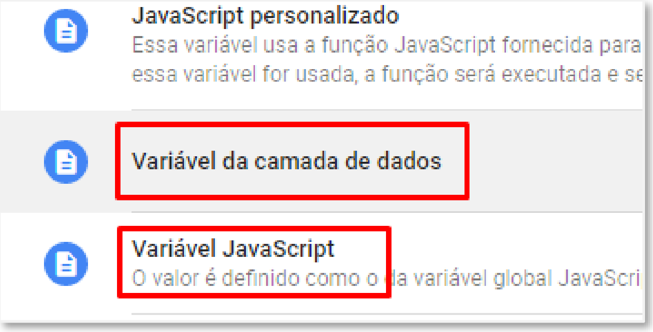
JavaScript personalizado
Essa variável usa a função JavaScript fornecida para calcular se
essa variável for usada, a função será executada e seu valor de r

The last two are Data Layer Variables. They collect data variables on the confirmation page when the following code is run on your website:
dataLayer.push({'Data Layer Variable Name': 'value'})

Please note that the Data Layer code must be saved on your website. Cannot enter via GTM. If these variables are not available, look for your platform's developer documentation. Another alternative for the variables below is to be a JavaScript variable, or Custom JavaScript the important thing is to extract the transaction ID and Sale Amount information from the site and have it in a variable.

3. **ID_Transacao: Gets the unit code of the action performed, unique value for each purchase.**
4. **Valor_Transação: Gets the variable representing the Purchase Amount. If your action has no action, such as a registration, or the traded amount of the commission is fixed, you can ignore this variable.**

Figure 8: Type of Variable



JavaScript personalizado
Essa variável usa a função JavaScript fornecida para
essa variável for usada, a função será executada e se

Variável da camada de dados

Variável JavaScript
O valor é definido como o da variável global JavaScri

OBS: **We should only look for the purchase price, without shipping. If it doesn't exist, you need to try to perform a calculation. One way would be to create a variable to subtract the total purchase amount minus shipping.**

In the example below, we created the Transaction_Value variable as a custom JavaScript variable.

```
function(){ var Valor_Transacao = {{TotalPedido}} -
{{ValorFrete}}; return Valor_Transacao.toFixed(4); }
```

1.2.1.2. Custom Image Tag Setup

To configure this process in Tag Manager, we create a new Tag with the custom image type.
Name the TAG Pixel_Actionpay.
Inside the image URL you must type:

Code 2: Example of the actionpay pixel

Where {{cookie_actionpay}}, {{ID_Transacao}} and {{Valor_Transacao}} where the Variables created above. **Use exactly the name of the variables created. The best way is to select the variable created in GTM instead of copying it from the manual, this way we avoid errors.**

Remenber to change the value **0000** **by the identification number of your campaign (goal id), obtained from the manager or your Actionpay account manager.**

1.2.1.2.1. Actionpay Pixel Tag Activation Rule

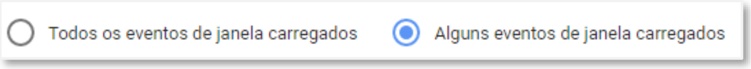
Now it is necessary to create the Pixel_Actionpay TAG activation rule.

Within the TAG Pixel_Actionpay go to Activation and add a new activator on the Upper right hand corner and + name the rule as: **Dispara_Actionpay.**

Under Trigger Setup select the trigger type. If there is an event that marks the end of the purchase, such as Compra_Confirmada, select this event.

Otherwise select Loaded Window (Page View).

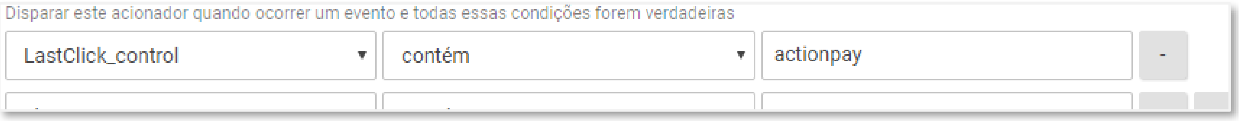
Select the field: Some events of Window loaded:



○ Todos os eventos de janela carregados ● Alguns eventos de janela carregados

You must create two conditions inside this rule, the first is: LastClick_control contains actionpay. Leave it in lower case because this field is sensitive to this variation.

Figure 9: Condition of trigger - LastClick



Disparar este acionador quando ocorrer um evento e todas essas condições forem verdadeiras

LastClick_control	contém	actionpay	-

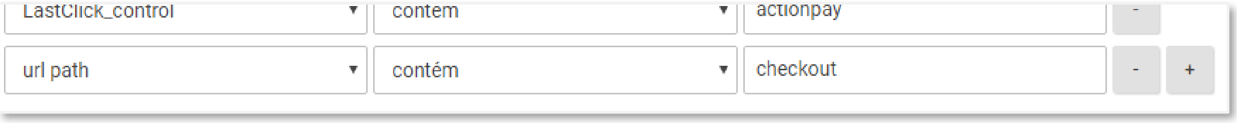
This rule guarantees that the user came from Actionpay.

The second rule must be configured to call the TAG only when the action is performed. If the action is a product sale, for example, the pixel must be called on the thank you for the purchase screen. If it is the register, it must be placed at the end of the register, etc.

This rule must be customized for your website, as it depends on its platform, an example of a rule is:

{{url path}} contém /checkout/ or Event equals checkout

Figure 10: Condition of trigger - Page of Confirmation

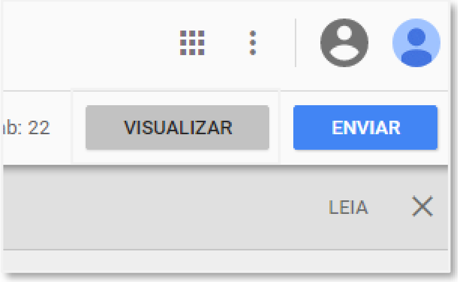


LastClick_control ▾	contem ▾	actionpay	-
url path ▾	contém ▾	checkout	- +

1.2.2. Testing pixel submission.

With the Pixel_Actionpay TAG created with its respective rules and variables, publish the container to perform the test. You can only test the Pixel_Actionpay Tag if the Cookie_Actionpay Tag is working correctly.

Figure 11- Window for debug the container within Google Tag Manager



Using the Google Chrome browser, go to Preview and select Debug published container.

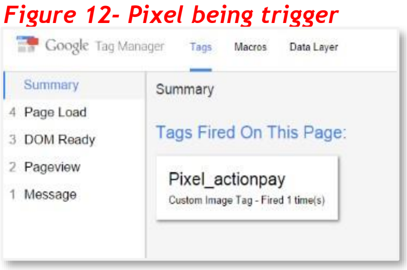
This command will open your website. For now the Pixel_Actionpay Tag must NOT be triggered, browse the site and ensure that it has not been triggered once. If it has been fired, check the Tag's firing rules.

Then, in the same window created by the debug command, copy and paste the link generated by your account manager, similar to the link below, directed to your website. Check if the Tag_actionpay Tag has been fired, on the website. To see the test of this Tag, view the item 1.1.1.2.

Example of tracked link :
http://apypxl.com/click/54501e6b8b30b837748b4567/00000/70505/subaccount (this link doens´t work!!)

With the Actionpay cookie created correctly and present in your browser, take an action and see if the pixel was fired correctly.

Figure 12- Pixel being trigger



Tags Fired On This Page:

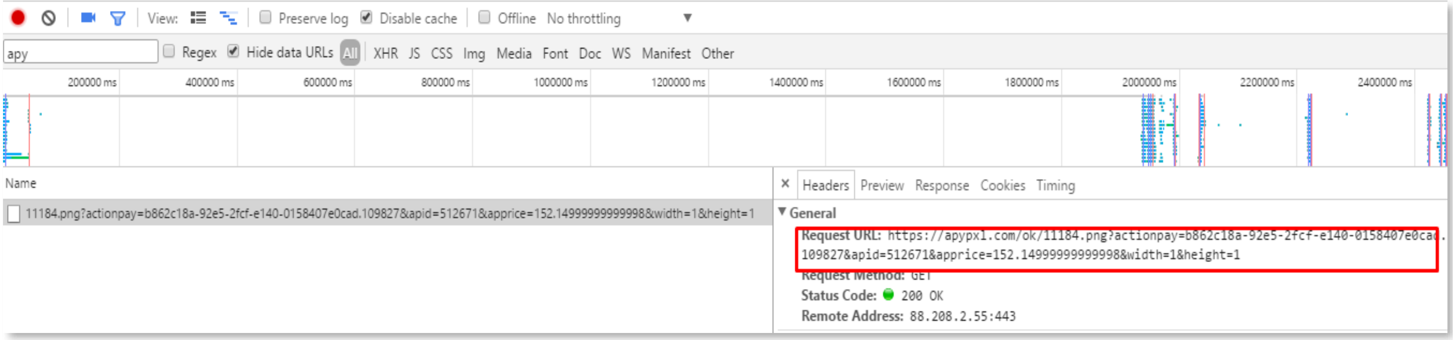
Pixel_actionpay
Custom Image Tag - Fired 1 time(s)

Example: If your action is sale, perform a test purchase, if your action is registration, perform the desired registration. Check the Pixel_actionpay Tag was fired on the desired confirmation page.

In Google Chrome Browser, hit f12 and fire up your development tab. Go to Network and look for actionpay. Reload the page (f5) to register the page load. If when reloading the page it is redirected, you must open the development tab before completing the purchase.

Check that all parameters were sent correctly.

Figure 13- Development Chrome / Network



☐ Preserve log ☑ Disable cache ☐ Offline No throttling ▼

apy ☐ Regex ☑ Hide data URLs

☐ 11184.png?actionpay=b862c18a-92e5-2fcf-e140-0158407e0cad.109827&apid=512671&apprice=152.14999999999998&width=1&height=1

▼ General

```
Request URL: https://apypx1.com/ok/11184.png?actionpay=b862c18a-92e5-2fcf-e140-0158407e0cad.
109827&apid=512671&apprice=152.14999999999998&width=1&height=1
Request Method: GET
Status Code: ● 200 OK
Remote Address: 88.208.2.55:443
```

In this parameter you should see all the [click].[source] parameters in addition to the values [apid] - action id and [price] - action value, if these fields are significant for your campaign.