árukereső.hu

An important thing is to run the Trusted code after the customer has confirmed his purchase.

Our code generates an HTML snippet, which has to reach the customer's browser to begin to send purchase information.

Sender Policy Framework information

Our system sends the Trusted Purchase Program survey emails to customer in behalf of you, using your registered email address in our system. If you wish to change this email address, please, contact us at info@arukereso.hu.

We would like to inform you that e-mail servers may verify the sender's origin via Sender Policy Framework (SPF). We'd like to suggest you to verify whether your system is using SPF settings.

If your domain has SPF settings to avoid bounced email traffic (or emails classified as spam) regarding survey feedbacks or quality reviews we would like to recommend adding '_spf-c.arukereso.hu' (our server's IP addresses) into Your SPF settings. We would like to ensure you that our system sends emails using your sender domain exclusively your given email address and survey emails regarding to our Trusted Purchase Program only.

If you avoid or are not able to set SPF settings for your domain name, we would like to suggest to send an email to info@arukereso.hu and inform us and our system will send these emails from megbizhatobolt@arukereso.hu email address.

árukereső.hu

1. API - PHP

- Copy the `TrustedShop.php` file to a place, where it's accessible for you webshop engine.
- Copy the content of the `Example.php` into that page, which runs when the customer confirms his purchase. It usually appears as a "Thank you" page. Here you will need the customer's e-mail address and the name of the purchased products.
- You have to customize the inserted code as follows:
    - Modify `require_once('TrustedShop.php')` according to the path where you placed your TrustedShop.php file.
    - Check the WebApiKey in the constructor and if it differs from your partner's WebApiKey, you have to replace it.
    - Set the customer's e-mail address with the `SetEmail()` method.
    - Add the purchased products with the `AddProduct()` method. It's callable multiple times and you should iterate through the customer's cart probably. It's first parameter is required and it must be the product name. The second parameter should be the product identifier, which is the same as in the partner feed, but it's optional.
    - The `Prepare()` method returns an HTML code snipet. By default we use an `echo` to print it into the webshop's source code, but it's not supported by some of the webshop engines. You have to take care of the generated code reaches the page's source code through the template system of the webshop engine.
    - Optionally you can implement an error handler with the example `try-catch` block.

árukereső.hu

2. API - ASP.NET C#

- Copy the `TrustedShop.cs` file to a place, where it's accessible for you webshop engine.
- Copy the following tag between `<body>` and `</body>` tags into that page, which runs when the customer confirms his purchase. It usually appears as a "Thank you" page. Here you will need the customer's e-mail address and the name of the purchased products:

```
<asp:Literal runat="server" ID="TrustedShopScript"></asp:Literal>
```

- Copy the content of the `Example.aspx.cs` into the same page's `Page_Load()` method.
- You have to customize the inserted code as follows:
    - Check the WebApiKey in the constructor and if it differs from your partner's WebApiKey, you have to replace it.
    - Set the customer's e-mail address with the `SetEmail()` method.
    - Add the purchased products with the `AddProduct()` method. It's callable multiple times and you should iterate through the customer's cart probably. It's first parameter is required and it must be the product name. The second parameter should be the product identifier, which is the same as in the partner feed, but it's optional.
    - The `Prepare()` method returns an HTML code snipet. By default we simply print it into the webshop's source code, but it's not supported by some of the webshop engines. You have to take care of the generated code reaches the page's source code through the template system of the webshop engine.
    - Optionally you can implement an error handler with the example `try-catch` block.

árukereső.hu

3. API - ASP Classic

- Copy the `TrustedShop.asp`, `md5.asp` and `aspJSON1.17.asp` files to a place, where it's accessible for you webshop engine.
- Copy the content of the `Example.asp` into that page, which runs when the customer confirms his purchase. It usually appears as a "Thank you" page. Here you will need the customer's e-mail address and the name of the purchased products.
- You have to customize the inserted code as follows:
    - Modify `<!--#include file="..."-->` parts according to the paths where you placed your .asp files.
    - Check the WebApiKey in the constructor and if it differs from your partner's WebApiKey, you have to replace it.
    - Set the customer's e-mail address with the `SetEmail()` method.
    - Add the purchased products with the `AddProduct()` method. It's callable multiple times and you should iterate through the customer's cart probably. It's first parameter is required and it must be the product name. The second parameter should be the product identifier, which is the same as in the partner feed, but it's optional.
    - The `Prepare()` method returns an HTML code snipet. By default we use an `Response.Write` to print it into the webshop's source code, but it's not supported by some of the webshop engines. You have to take care of the generated code reaches the page's source code through the template system of the webshop engine.
    - Optionally you can implement an error handler with the example `HasError()` and `GetErrorMessage()` methods.

Notes to webshop engine developers

You cannot hardcode the same WebApiKey if you manage several webshops at a time, because every implementation has to use it's own WebApiKey, according to the partner's webshop. We use WebApiKey to identify our partners, that's why it has to be unique for each partners and webshops. Our APIs help with this and also provide to registrate a new webshop into the TrustedShop program, set the appropriate purchase information and send them to us to help the webshop reach the Trusted status. What you should do with our API then? It depends on what kind of language your webshop engine's built on.