

# Developer Assessment

## Introduction

XML is an often used medium in affiliate marketing used to transport data between parties. TradeTracker can interpret, store and provide XML-feeds in several formats and character sets.

One of the challenges faced by TradeTracker is handling very large XML-feeds without putting too much pressure on either the CPU or the memory. Because of that feeds can no longer be put into memory, but have to be processed in an alternative way.

## The assignment

The assignment consists of two parts that should take about an equal amount of time (make sure that neither part gets neglected due to the other).

### Part 1: User Interface

Create a page that will allow the user to submit a URL of a product feed. On submitting the form, the URL should be processed (as described in following section), and the results of that processing should be shown to the user.

### Part 2: Feed Processing

The feed processing function should be able to handle very large feeds of a fixed format. For an example feed, the following test can be used:

<http://pf.tradetracker.net/?aid=1&type=xml&encoding=utf-8&fid=251713&categoryType=2&additionalType=2>

The feed is a few hundred megabytes large and contains thousands of products. The basic structure of the feed is described in appendix section.

Your code should:

1. Fetch the contents of given URL without downloading entire file beforehand,
2. Extract the following fields:
  - a. productID
  - b. name
  - c. description
  - d. price & currency
  - e. category (all)
  - f. productURL
  - g. imageURL
3. Render above extracted data in user-friendly way on front-end

## Restrictions

- HTML5 based markup
- Use a front-end MV\* framework to make the interface interactive
- Make use of a CSS framework for responsive / adaptive layout
- Use a MVC based PHP framework for back-end
- No database is needed
- Any PHP used should be E\_STRICT compliant and limited to 32MB of memory

## Criteria

The code will be judged on the following criteria:

- Strategy for solving the problem
- Use of OOP
- Compliance to coding standards (PSR-2)
- Readability and inline documentation (preferably code should explain itself and not need additional comments)
- Security of the interface
- Scalability of the feed processing solution
- Resilience of the solution
- README file to setup working solution
- Solution should be functional and complete
- Tests and code coverage of at least 30%

## Bonus

- Use of Symfony (or its derivate) as a backend framework
- Use of CSS pre-processor
- Displaying processed data in real-time
- Code coverage of 90% or higher
- Provide a Docker image for solution. Preferably in code repository

## Appendix

### Feed structure

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Generated on 11/22/07 at 16:50:00 (http://www.tradetracker.net)-->
<products>
  <product>
    <productID>0062-16156</productID>
    <name>ATTEMA Att Hoofdschakelaar 2p 40A</name>
    <price currency="EUR">2378.81</price>

    <productURL>http://www.hardware.nl/tt/?tt=541_0_1_&r=attema.hardware.nl%2F16
156</productURL>
    <imageUrl />
    <description><![CDATA[Att Hoofdschakelaar 2p 40A]]></description>
    <categories>
      <category path="INSTALLATIEKASTEN">INSTALLATIEKASTEN</category>
    </categories>
    <additional>
      <field name="priceExVat">1999.00</field>
      <field name="EAN">16156</field>
      <field name="deliveryCosts">5.95</field>
      <field name="deliveryTime">2-6 dagen</field>
    </additional>
  </product>
  <product>...</product>
  ...
</products>
```