Assessment PHP Developer

Author: Vincent de Lau
Date: 22-04-2014
Version: 1.0
Number of pages: 4

Contents

1. Introduction

This assessment is designed to assess the PHP and XML skills for a candidate applying for a job as PHP Developer with Voiceworks. For this assessment, the task is to create an API entry point for a fictitious XML API. The API is using a custom XML format as specified in attached XSD schema's and explained in the document.

1.1 Goals

- Create an entry point that can handle the specified XML messages and return an appropriate answer.
- Make sure new types of messages can be added easily.
- If an error occurs in processing the request, reply with an appropriate error.
- If an unspecified or invalid XML document is sent, reply with an appropriate error.
- Use PHP's OOP functionality. PHP 5.2 is minimal, features from newer version are allowed.
- Implement the two reference functions, ping_request and reverse_request

1.2 Hints and tips

- PHP's ErrorException
- Return the appropriate HTTP headers and status codes.
- Things that are not specified in this document or its additional files, are open for interpretation by the candidate. Choices made should be documented in code or an added README file.
- Submit all created code, including tests used during development or other code that might be of interest.

1.3 Additional file provided

Provided are XSD's and a sample XML file for:

- nack
- ping_request
- ping_response
- reverse_request
- reverse_response

If any files are missing, please contact us immediately!

2. API description

The API is not a standard SOAP or REST API. This fictitious API is modelled after actual API's in use in the Dutch telecom market. Aspects removed from this fictitious API include security, versioning and alternative transport mechanisms (like email).

2.1 HTTP transport mechanism

The HTTP transport mechanism uses HTTP POST requests, with the XML message in the HTTP body. The answer XML is returned as the body of the response.

2.2 XML message details

The XML messages share a common structure. The XML documents don't use namespaces and don't have XSD references in the XML itself. By looking at the header element which is present in each message, the entry point can determine the type of the message and thus the XSD validate against.

All messages consist for a root element, containing a header and a body element. The header element is common for all message types and consists of:

- type, the type of message
- sender, a code for the sender
- recipient, a code for the recipient
- reference, a reference set in the request, copied into the responses for correlation (optional)
- timestamp, the time when the message was created (optional, end-point should always include the timestamp in its responses)

The content of the body differs per type of request. In response messages, there is an optional error element to signal that the request could not be performed.

When the entry point cannot process the message for some reason (parse error, unknown function etcetera), it must respond with a nack message. The body of the nack must contain an error element describing the error.

3. Functions to implement

3.1 ping

The first function to implement is the ping function. The client sends a ping_request message with an optional echo element in the body. In response to this request, the ping service answers with a ping_response message, optionally copying the echo element in the body.

3.2 reverse

The reverse function reverses a given string. The client sends a reverse_request message with a string element in the body. In the reverse_response, both the original string and the reversed string are returned.