European Centre for the Development of Vocational Training

Version 1.0, September 2011

Europass team

Europass Web Services Version 2.0 Documentation

Table of Contents

THE EUROPASS INITIATIVE **2**

THE EUROPASS ONLINE EDITORS **2**

GETTING THE MOST OUT OF THE EUROPASS API **2**

1. INTRODUCTION **4**

1.1. SCOPE 4
1.2. NORMATIVE REFERENCES 4
1.3. ABBREVIATIONS AND ACRONYMS 4

2. GETTING STARTED **5**

2.1. CHANGELOG 6
2.2. KNOWN ISSUES AND ENHANCEMENTS LIST 6
2.2.1. Known Issues *6*
2.2.2. Enhancement List *6*

3. EUROPASS SOAP SERVICES **7**

3.1. OVERVIEW 7
3.2. ENCODING 8
3.3. SECURITY 8
3.4. GENERAL NOTE 9
3.5. FAULT / EXCEPTION HANDLING 10
3.6. EUROPASS SOAP SERVICES 11
*3.6.1. **Europass XML – Based** (EuropassXMLConversionService)* *12*
*3.6.2. **Adobe PDF + Europass XML** – Based (PDFWithEuropassXMLConversionService)* *17*
*3.6.3. **HR-XML – Based** (HRXMLConversionService)* *22*
*3.6.4. **Adobe PDF+HR-XML – Based** (PDFWithHRXMLConversionService)* *26*
*3.6.5. **Europass JSON – Based** (JSONConversionService)* *30*
4.1. JAVA EXAMPLES 36
4.1.1. Generate Necessary Code from WSDL *36*
4.1.2. Utilise Europass Java Client *36*
4.2. PHP EXAMPLES 37
4.2.1. Prerequisites *37*
4.2.2. Examples *37*

THE EUROPASS INITIATIVE

Europass is a portfolio of five documents that describe and provide evidence on your skills, work experience, qualifications and competences in a transparent manner. Europass helps you make your skills and qualifications clearly and easily understood in Europe, supporting this way your chances of mobility in Europe.

The five Europass documents are the **Curriculum Vitae**, the **Language Passport**, the **Mobility**, the **Diploma Supplement** and the **Certificate Supplement.** All documents follow a specific Europass template. The use of a common template is considered a key feature in achieving transparency and neutrality while presenting your personal skills and competences. Of the five Europass documents, you may fill-in the Curriculum Vitae and the Language Passport yourself, while the rest three documents are issued by the appropriate authorities.

THE EUROPASS ONLINE EDITORS

The Europass portal provides two online editors for creating and editing your Europass Curriculum Vitae and Language Passport. By using these editors you fill-in simple web-based forms and finally receive your document. The document can be received as XML file, according to the Europass XML Schema (current version 2.0), as Microsoft Word 97-2003 file (.doc), as OpenDocument ver 2 Text file (.odt), as Adobe Portable Document Format with Europass XML attachment (.pdf), or as HTML.

By receiving your document in Europass XML or PDF with XML attachment you ensure the reusability and portability of your personal data. Since XML is a machine-interpretable representation of your data, it is possible that you upload your document on the Europass online editor (or any other online application compatible with Europass) and have your information available for further editing or any other use. To further promote interoperability Europass offers a public set of web services (API).

GETTING THE MOST OUT OF THE EUROPASS API

To achieve maximum reusability of your personal data and at the same time increase the acceptance and visibility of Europass documents, Europass provides a public set of web services. These services offer conversion methods between various file formats supported by Europass.

Europass encourages integration with systems and environments that maintain a repository of personal data and offer:

- job matching services (eg job portals, employment agencies)
- promotion services (eg. career offices, human resource management offices)
- profile maintaining services (eg blog publishing tools, social networking sites)

The supported file formats are: Europass XML, HR-XML, Adobe PDF with Europass XML attachment, Adobe PDF with HR-XML attachment, Europass JSON, Microsoft Word 97-2003, OpenDocument ver 2 Text and HTML.

1. INTRODUCTION

1.1. Scope

The scope of this document is to provide documentation on the Europass public API and thus facilitate the reuse of the public services by developers. Towards this end each web service is thoroughly documented in terms of its behaviour, functionality, input and output schemes, and elaborated by a set of usage scenaria and examples.

1.2. Normative References

The following references documents are indispensable for the application of Europass web services.

Europass XML Schema v2.0, available online:

http://europass.cedefop.europa.eu/TechnicalResources/XML/xsd/EuropassSchema_V2.0.xsd

Europass JSON Schema v2.0, available online:

http://europass.cedefop.europa.eu/TechnicalResources/JSON/schema/Europass_JSONSchema_2.0.json

HR-XML Candidate Schema v2.5, available online:

http://ns.hr-xml.org/2_5

1.3. Abbreviations and Acronyms

ECV	Europass Curriculum Vitae
ELP	Europass Language Passport
EXML	Europass XML file
HR-XML	HR-XML file
PDF+EXML	Europass Adobe Portable Document Format file with Europass XML attachment
PDF+HR-XML	Europass Adobe Portable Document Format file with HR-XML attachment
EJSON	Europass JSON object/file
DOC	Microsoft Word file
ODT	OpenDocument ver 2 Text Document file
HTML	HyperText Markup Language file
WSDL	Web Services Description Language used to describe the Europass web services.
SOAP	Simple Object Access Protocol specification for exchanging structured information in the implementation of web services.

2. GETTING STARTED

1. **Locate the WSDL of the available services**:

EXML–based:
http://europass.cedefop.europa.eu/soapws/services/EuropassXMLConversionService?wsdl

PDF+XML-based:
http://europass.cedefop.europa.eu/soapws/services/PDFWithEuropassXMLConversionService?wsdl

JSON–based:
http://europass.cedefop.europa.eu/soapws/services/JSONConversionService?wsdl

HR-XML–based:
http://europass.cedefop.europa.eu/soapws/services/HRXMLConversionService?wsdl

PDF+HR-XML-based:
http://europass.cedefop.europa.eu/soapws/services/PDFWithHRXMLConversionService?wsdl

2. **Read the API Documentation and the implementation examples**

3. **Build and Test your application**

4. **Launch your application** (and optionally if it is open-source submit it to OSOR.eu Europass)

Feedback

We welcome your feedback and anticipate your comments, complaints, ideas!

Contact us: europass-feedback@cedefop.europa.eu

Technical Community

If you are interested on the technical developments and resources around Europass participate in the

Europass Technical Community within OSOR.eu (http://www.osor.eu)

The objectives of this community are mainly:

- to bring together all individuals, technical developers, semantic experts and organisations interested in Europass and working on related domains such as, personal information or resume management, competencies and skills representation, hr and cv-job matching services, etc.
- to act as a point of reference where individuals and organisations will contribute their projects and experiences around the Europass semantic assets, web services and tools.
- to host discussions, reviews and contributions related to Europass, which could potentially influence Europass further developments of semantic assets and services.

If you agree to participate, please follow the steps below:

(1) Register as a member with OSOR.eu (https://register.osor.eu/)

(2) Join the Europass Technical Community (http://www.osor.eu/communities/europass)

2.1. Changelog

September 2011: First release of the Europass API Documentation

2.2. Known Issues and Enhancements List

2.2.1. Known Issues

The Europass SOAP services support MTOM/XOP optimisation of SOAP messages. Therefore you need to take this into consideration while parsing the response message. We recommend using an out-of-the-box library to help you handle the SOAP-based communication.

2.2.2. Enhancement List

Support the validation of the received JSON objects according to the Europass JSON Schema;

3. EUROPASS SOAP SERVICES

3.1. Overview

The Europass API currently consists of five services that provide conversion methods between the various file formats supported by Europass. The services accept a machine-interpretable representation of an individual's personal data (eg ECV, ELP) and return the same data in a different file format. The file formats used as input are the Europass XML (EXML), the HR-XML (HR-XML), the Adobe PDF with Europass XML attachment (PDF+EXML), the Adobe PDF with HR-XML attachment (PDF+HR-XML) and the Europass JSON (EJSON). The returned file formats are –apart from those used as input- the Microsoft Word 97-2003 (DOC), the OpenDocument ver 2 Text (ODT), and the HTML. The conversion services methods support conversions for both ECV and ELP data, since both data can be represented in EXML, HR-XML or JSON formats.

Currently the conversion services are offered as SOAP web services. A brief overview of the available services along with the expected inputs and outputs is presented in the table below:

Input	Endpoint	Output
1. EuropassXMLConversionService		
Europass XML 2.0	convertToPDFwithXMLCV / convertToPDFwithXMLLP	PDF+Europass XML 2.0
	convertToPDFwithHRXMLCV / convertToPDFwithHRXMLLP	PDF+HR-XML 2.5
	convertToMSWordCV / convertToMSWordLP	MS WORD
	convertToODTCV / convertToODTLP	ODT
	convertToHTMLCV / convertToHTMLLP	HTML
2. PDFWithEuropassXMLConversionService		
PDF+Europass XML 2.0	extractXML / convertToXMLCV / convertToXMLLP	Europass XML 2.0
	convertToHRXML	HR-XML 2.5
	convertToPDFwithHRXMLCV / convertToPDFwithHRXMLLP	PDF+HR-XML 2.5
	convertToMSWordCV / convertToMSWordLP	MS WORD
	convertToODTCV / convertToODTLP	ODT
	convertToHTMLCV / convertToHTMLLP	HTML
3. HRXMLConversionService		
Europass XML 2.0	convertToHRXML	HR-XML 2.5
HR-XML 2.5	convertToXMLCV / convertToXMLLP	Europass XML 2.0

	convertToPDFwithXMLCV / convertToPDFwithXMLLP	PDF+Europass XML 2.0
	convertToPDFwithHRXMLCV / convertToPDFwithHRXMLLP	PDF+HR-XML 2.5
	convertToMSWordCV / convertToMSWordLP	MS WORD
	convertToODTCV / convertToODTLP	ODT
	convertToHTMLCV / convertToHTMLLP	HTML
4. PDFWithHRXMLConversionService		
PDF+HR-XML 2.5	convertToHRXML	HR-XML 2.5
	convertToXMLCV / convertToXMLLP	Europass XML 2.0
	convertToPDFwithXMLCV / convertToPDFwithXMLLP	PDF+Europass XML 2.0
	convertToMSWordCV / convertToMSWordLP	MS WORD
	convertToODTCV / convertToODTLP	ODT
	convertToHTMLCV / convertToHTMLLP	HTML
5. JSONConversionService		
Europass XML 2.0	convertFromXMLCV / convertFromXMLLP	Europass JSON 2.0
HR-XML 2.5	convertFromHRXML	Europass JSON 2.0
Europass JSON 2.0	convertToXMLCV / convertToXMLLP	Europass XML 2.0
	convertToHRXML	HR-XML 2.5
	convertToPDFwithXMLCV / convertToPDFwithXMLLP	PDF+Europass XML 2.0
	convertToPDFwithHRXMLCV / convertToPDFwithHRXMLLP	PDF+HR-XML 2.5
	convertToMSWordCV / convertToMSWordLP	MS WORD
	convertToODTCV / convertToODTLP	ODT
	convertToHTMLCV / convertToHTMLLP	HTML

3.2. Encoding

The Europass API expects all data to be UTF-8 encoded.

3.3. Security

Given that the services involve personal data (ie Curriculum Vitae) data transport is realised through Secure Socket Layer (SSL).

3.4. General Note

As a general note we would like to stress the fact that the SOAP Message Transmission Optimisation Mechanism (SOAP MTOM) is used to optimise the transmission of base64 encoded data. This is the case for the services that return a binary file (eg Adobe PDF, Microsoft Word and OpenDocument ver 2 Text). According to the SOAP MTOM specification, binary attachments are references by the use of the XOP:Include element.

By using MTOM/XOP to optimize a SOAP message, the message is serialised into a MIME Multipart/Related message. The base64Binary data are extracted from the SOAP message and packaged as separate binary attachments within the MIME message.

Below you may see a SOAP message including an Adobe PDF file and its MTOM/XOP optimised version:

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
      <soapenv:Body>
            <europassws:FileResponse xmlns:europassws="http://
services.europass.cedefop.europa.eu">
                  <europassws:file xsi:type=”base64binary”>
                        4dfr… … … </soapws:file>
            </soapws:FileResponse>
      </soapenv:Body>
</soapenv:Envelope>
```

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
      <soapenv:Body>
            <europassws:FileResponse xmlns:europassws="http://
services.europass.cedefop.europa.eu">
                  <europassws:file>
                        <xop:Include
href="cid:1.urn:uuid:2B9F810A452836D1CB1305189222177@apache.org"

xmlns:xop="http://www.w3.org/2004/08/xop/include" />
                  </soapws:file>
            </soapws:FileResponse>
      </soapenv:Body>
</soapenv:Envelope>
--MIMEBoundaryurn_uuid_2B9F810A452836D1CB1305189222160
1ff8
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID:
<1.urn:uuid:2B9F810A452836D1CB1305189222177@apache.org>
%PDF-1.4
… … …
```

To handle such response we recommend using corresponding web services frameworks and libraries that support MTOM/XOP handling. For example you may use Apache Axis 2 for Java or C, and WSO2 Web Services Framework for PHP.

3.5. Fault / Exception Handling

All of the possible problems that are related to the SOAP communication between the client and the server of the Axis 2 web services are handled by Apache Axis2 Faults (e.g. communication problems, soap message formatting problems, etc).

Additionally AxisFault exceptions relay the exceptions thrown on the Europass server side when the execution of the client request fails.

Generally the **document generation process** is described below, and differs according to the file format of the input and the output.

(1) **If the requested output is HTML, Europass XML, HR-XML or Europass JSON**, then the input document is properly transformed / translated to the requested output format (if the input is Adobe PDF with attachment, the attachment XML is extracted before it is transformed).

(2) **If the requested output is Adobe PDF with Europass XML, Adobe PDF with HR-XML, MS Word or OpenDocument Text** then:

(a) the extracted/ transformed/ translated Europass XML from the input file is used to fill-in a Europass CV/LP Form;

(b) the Europass CV/LP Form is used to produce an OpenDocument Text file (which is actually a zipped file with content and styles in XML format).

(c) If the requested output is Adobe PDF with Europass XML, Adobe PDF with HR-XML or MS Word, the OpenDocument server is utilised to convert the OpenDocument Text file created at the previous step to the requested file format.

(d) Finally the file is converted to a byte array, and sent to the client within the SOAP response.

The possible **Faults** you may receive when calling the Europass Conversion Methods are listed below:

- **org.apache.axis2.AxisFault: message: "Failed to parse xml into CV/ LP Form."** When server fails to load a Europass CV/ LP form (as Java bean) from the submitted/ extracted/ transformed/ translated Europass XML.

- **org.apache.axis2.AxisFault: message: "Failed to generate document."** When server fails to produce an OpenDocument Text file to either be returned as output or to further be converted to Adobe PDF or MS Word. Possible causes are:

  – server failed to store temporarily the submitted file;

- server failed to created the OpenDocument Text file as a zipped output;

- server failed to communicate with the database

- **org.apache.axis2.AxisFault: message: "No conversion server available."** When server failed to reach the OpenDocument server to perform the necessary conversion (OpenDocument to Adobe PDF or MS Word).

- **org.apache.axis2.AxisFault: message: "Could not retrieve locale."** When server fails to communicate with the database and retrieve information for the requested locale. Note that if no locale exists in the input parameters, or it is invalid (el is considered invalid, while el_GR is valid) the generated document will be in the default locale, en_GB.

- **org.apache.axis2.AxisFault: message: "Copy XML to temporary file failed."** When server fails to temporarily store in the file system the XML file to be attached to a temporary Adobe PDF.

- **org.apache.axis2.AxisFault: message: "Failed to create XML attachment to PDF."** When server fails to attach the XML to the temporary Adobe PDF file, by means of the iText library.

- **org.apache.axis2.AxisFault: message: "Failed to copy back pdf with attachment."** When server fails to copy the temporarily created Adobe PDF file, which was used to complete the XML attaching, back to the requested Adobe PDF output file.

- **org.apache.axis2.AxisFault: message: "File is too big."** When the size of the produced file (in Adobe PDF, MS Word or OpenDocument Text) is greater than the max value of a Java integer (e.g. 2^31-1).

- **org.apache.axis2.AxisFault: message: "Could not completely read file."** When server fails to read the entire produced file (in Adobe PDF, MS Word or OpenDocument Text).

- **org.apache.axis2.AxisFault: message: "Could not convert file to byte array."** When server fails to read the bytes of the produced file (in Adobe PDF, MS Word or OpenDocument Text) in order to send it back to the client.

In case you face persistent problems with using the services, do not hesitate to contact us. When you do so it would be most helpful to include information on your programming environment, on the input parameters you try to pass (or the SOAP message you send) and the fault message you receive.

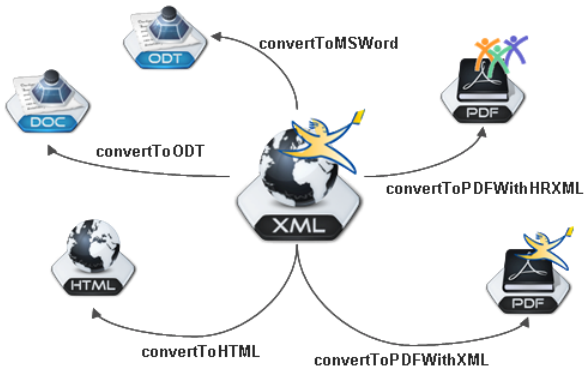
3.6. Europass SOAP Services

Examples are briefly given for each service, but are fully elaborated in sections “Java Examples” and “PHP Examples”.

3.6.1. ***Europass XML – Based*** *(EuropassXMLConversionService)*

Cue: Convert from Europass XML

Synopsis: This conversion service returns a Europass Curriculum Vitae or Language Passport document in one of the following formats: Adobe PDF with Europass XML attachment (PDF+EXML), Adobe PDF with HR-XML attachment (PDF+HR-XML), Microsoft Word 97-2003 (DOC), OpenDocument ver 2 Text (ODT), and HTML.



Request: The request object is an XMLLocaleRequest. This is a custom complex data type consisting of two string elements: “xml”, which holds the input Europass XML document as string, and “locale”, which holds the code of the language in which the returned document will be translated. Note that **the locale is required** and when missing a Fault (message “Could not retrieve locale”) will occur:

```
<element name=" XMLLocaleRequest"/>
<complexType>
      <sequence>
            <element name="xml" type="xsd:string"/>
            <element name="locale" type="xsd:string"/>
      <sequence>
</complexType>
```

In your implementation you will most probably need to create a data structure consisting of these two variables (eg JavaBean in Java or StdClass in PHP5) and pass it as a request parameter to the service.

Response: The response object is either a FileResponse or an HTMLResponse. Both are custom complex data types consisting of one element. The difference is that in the case of FileResponse this element, “file”, represents base64 encoded binary content, while in the case of HTMLResponse this element, “html” represents textual content.

```
<element name=" FileResponse"/>
<complexType>
      <sequence>
            <element name="file" type="xsd:base64binary"/>
      <sequence>
```

```
</complexType>

<element name=" HTMLResponse"/>
<complexType>
      <sequence>
            <element name="html" type="xsd:string"/>
      <sequence>
</complexType>
```

To consume the response you will most probably need to create a data structure consisting of this variable (eg JavaBean in Java or StdClass in PHP5) and handle the binary or the string content properly.

Please note that “locale” consists of the language and country ISO codes, eg. “fr_FR”, “de_DE”.

Methods: The methods provided by this service support the conversions to the various file types.

convertToPDFwithXMLCV / convertToPDFwithXMLLP

Will return the base64 binary content of an Adobe PDF file with Europass XML attachment, as ECV or ELP document respectively.

convertToPDFwithHRXMLCV / convertToPDFwithHRXMLLP

Will return the base64 binary content of an Adobe PDF file with HR-XML attachment, as ECV or ELP document respectively.

convertToMSWordCV / convertToMSWordLP

Will return the base64 binary content of a Microsoft Word 97-2003 file as ECV or ELP document respectively.

convertToODTCV / convertToODTLP

Will return the base64 binary content of OpenDocument ver 2 text file as ECV or ELP document respectively.

convertToHTMLCV / convertToHTMLLP

Will return a string representing the HTML of ECV or ELP document respectively.

Use Scenario: You maintain a repository of CVs and you are able to export a CV in Europass XML format. By using this service you can offer the option to your users to receive their CV anytime formatted according to the Europass template in Adobe PDF.

Examples: The implementation depends on the programming language you will use. Below we provide code snippets for calling the service in Java and PHP5.

3.6.1.1. *Example: "Convert Europass XML to Europass CV in HTML"*

JAVA

Note: The classes used in the example are automatically generated by Apache Axis 2 parsing the WSDL of the service.

```
#Get the url of the service
String url =
"http://europass.cedefop.europa.eu/soapws/services/EuropassXML
ConversionService.EuropassXMLConversionServiceHttpSoap12Endpoi
nt";
//Get the SOAP client implementation
EuropassXMLConversionServiceStub client =
  new EuropassXMLConversionServiceStub( url );
//Prepare the input complex type
EuropassXMLConversionServiceStub.XMLLocaleRequest   request   =

  new EuropassXMLConversionServiceStub.XMLLocaleRequest();
request.setXml( exml_to_string );
request.setLocale("fr_FR");
//Call the method
EuropassXMLConversionServiceStub.HTMLResponse response =
 client.convertToHTMLCV( request );
//Get the response
String html = response.getHtml();
```

PHP – SOAP Client

```
#Get the url of the service
$url =
"http://europass.cedefop.europa.eu/soapws/services/EuropassXML
ConversionService?wsdl";
#Create the input complex type
$inRequest= new StdClass();
$inRequest->xml =
  new SoapVar($xml_to_string ,XSD_STRING,null,null,"xml",$ns);
$inRequest->locale =
  new SoapVar("en_GB",XSD_STRING,null, null, "locale", $ns);
#Create parameter for the Soap Call
$input = new SoapVar($inRequest, XSD_ANYTYPE, $inType, $ns);
#Create a new SOAP Client (WSDL mode)
$sc = new SoapClient($url,$soap_options);
#Call the method
$result = $sc->convertToHTMLCV ($input);
#Get the response
echo $result->html;
```

PHP – WS02 WSF

```
#Get the url of the service
$url =
"http://europass.cedefop.europa.eu/soapws/services/EuropassXML
ConversionService?wsdl";
#Create the input complex type
$input_array = array( "xml" => $xml_to_string,
                      "locale" => "de_DE") ;
#Create the SOAP WS Client and Obtain a proxy
$client = new WSClient(array("wsdl"=>$url));
$proxy = $client->getProxy();
#Call the method and Get the response
$return_array = $proxy->convertToHTMLCV($input_array);
echo $return_array['html'];
```

3.6.1.2. *Example: "Convert Europass XML to Europass CV in ODT"*

JAVA

Note: The classes used in the example are automatically generated by Apache Axis parsing the WSDL of the service.

```
#Get the url of the service
String url =
"http://europass.cedefop.europa.eu/soapws/services/EuropassXML
ConversionService.EuropassXMLConversionServiceHttpSoap12Endpoi
nt";
//Get the SOAP client implementation
EuropassXMLConversionServiceStub client =
  new EuropassXMLConversionServiceStub( url );
//Prepare the input complex type
EuropassXMLConversionServiceStub.XMLLocaleRequest   request   =

  new EuropassXMLConversionServiceStub.XMLLocaleRequest();
request.setXml( exml_to_string );
request.setLocale("fr_FR");
//Call the method
EuropassXMLConversionServiceStub.FileResponse response =
 client.convertToODTCV( request );
//Get the response file
DataHandler file = response.getFile();
```

PHP – SOAP Client

Note: When XOP:include is used, the response will need to be parsed differently

```
#Get the url of the service
$url =
"http://europass.cedefop.europa.eu/soapws/services/EuropassXML
ConversionService?wsdl";
#Create the input complex type
$inRequest= new StdClass();
$inRequest->xml =
  new SoapVar($xml_to_string ,XSD_STRING,null,null,"xml",$ns);
$inRequest->locale =
  new SoapVar("en_GB",XSD_STRING,null, null, "locale", $ns);
#Create parameter for the Soap Call
$input = new SoapVar($inRequest, XSD_ANYTYPE, $inType, $ns);
#Create a new SOAP Client (WSDL mode)
$sc = new SoapClient($url,$soap_options);
#Call the method
$result = $sc->convertToODTCV ($input);
#Write the response to a file
$fp=fopen($out_file,"w");
fwrite($fp,$result->file);
fclose($fp);
```

PHP – WS02 WSF

```
#Get the url of the service
$url =
"http://europass.cedefop.europa.eu/soapws/services/EuropassXML
ConversionService?wsdl";
#Create the input complex type
$input_array = array( "xml" => $xml_to_string,
                      "locale" => "de_DE") ;
#Create the SOAP WS Client and Obtain a proxy
```

```
$client = new WSClient(array("wsdl"=>$url,
                             “responseXOP”=>true));
$proxy = $client->getProxy();
#Call the method and Get the response
$return_array = $proxy->convertToODTCV($input_array);
$returned_file = $return_array['file_encoded'] ;
```

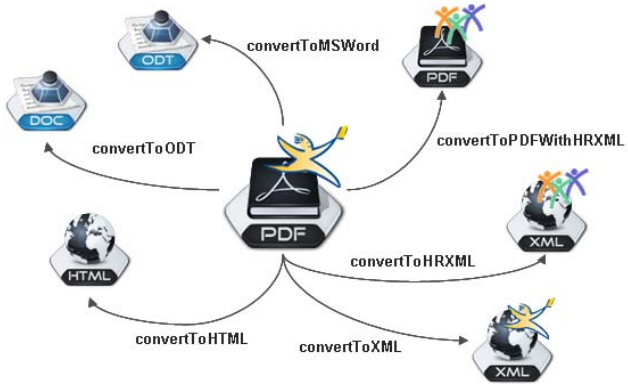
Related APIs: For Java-based implementations, the Apache Axis implementation of SOAP could be considered useful.

For PHP-based implementations, the PHP Soap Module is necessary.

3.6.2. *Adobe PDF + Europass XML – Based (PDFWithEuropassXMLConversionService)*

Cue: Convert from Adobe PDF with Europass XML attachment

Synopsis: This conversion service returns a Europass Curriculum Vitae or Language Passport document in one of the following formats: Europass XML (EXML), HR-XML (HRXML), Adobe PDF with HR-XML attachment (PDF+HR-XML), Microsoft Word 97-2003 (DOC), OpenDocument ver 2 Text (ODT), and HTML.



Request: The request object is either a FileRequest or a FileLocaleRequest. Both are custom complex data types consisting of sub-elements. More specifically, FileRequest consists of a "file" element", which holds the input Europass PDF document as base64 binary content, while FileLocaleRequest, has the additional element "locale", which holds the code of the language in which the returned document will be translated. Note that **the locale is required** and when missing a Fault (message "Could not retrieve locale") will occur:

```
<element name=" FileRequest"/>
<complexType>
      <sequence>
            <element name="file" type="xsd:base64binary"/>
      <sequence>
</complexType>
<element name=" FileLocaleRequest"/>
<complexType>
      <sequence>
            <element name="file" type="xsd:base64binary"/>
            <element name="locale" type="xsd:string"/>
      <sequence>
</complexType>
```

In your implementation you will most probably need to create a data structure consisting of these two variables (eg JavaBean in Java or StdClass in PHP5) and pass it as a request parameter to the service.

Response: The response object is either a FileResponse, or an XMLResponse, or an HTMLResponse. All three are custom complex data types consisting of one element. The difference is that in the case of FileResponse this element, "file", represents

base64 encoded binary content, while in the case of XMLResponse and HTMLResponse this element, “xml” and “html” respectively, represents textual content.

```
<element name=" FileResponse"/>
<complexType>
      <sequence>
            <element name="file" type="xsd:base64binary"/>
      <sequence>
</complexType>
<element name=" XMLResponse"/>
<complexType>

      <sequence>
            <element name="xml" type="xsd:string"/>
      <sequence>
</complexType>
<element name=" HTMLResponse"/>
<complexType>
      <sequence>
            <element name="html" type="xsd:string"/>
      <sequence>
</complexType>
```

To consume the response you will most probably need to create a data structure consisting of this variable (eg JavaBean in Java or StdClass in PHP5) and handle the binary or the string content properly.

Please note that “locale” consists of the language and country ISO codes, eg. “fr_FR”, “de_DE.

Methods: The methods provided by this service support the conversions to the various file types.

extractXML / convertToXMLCV / convertToXMLLP

Will return a string representing the Europass XML of ECV or ELP document respectively.

convertToHRXML

Will return a string representing the HR-XML of ECV or ELP document respectively.

convertToPDFwithHRXMLCV / convertToPDFwithHRXMLLP

Will return the base64 binary content of an Adobe PDF file with HR-XML attachment, as ECV or ELP document respectively.

convertToMSWordCV / convertToMSWordLP

Will return the base64 binary content of a Microsoft Word 97-2003 file as ECV or ELP document respectively.

convertToODTCV / convertToODTLP

Will return the base64 binary content of OpenDocument ver 2 text file as ECV or ELP document respectively.

convertToHTMLCV / convertToHTMLLP

Will return a string representing the HTML of ECV or ELP document respectively.

Use Scenario: You receive a Europass CV in Adobe PDF format and you need to store the information in a database. Instead of trying to parse the PDF itself or manually transferring the information, you use this service to extract the attached XML. Then you proceed with the parsing of the XML, which is a much more straightforward procedure, given the plethora of available tools and libraries in all programming languages.

Examples: The implementation depends on the programming language you will use. Below we provide code snippets for calling the service in Java and PHP5.

3.6.2.1. Example: "Extract the Europass XML from an Adobe PDF with EXML CV"

JAVA

Note: The classes used in the example are automatically generated by Apache Axis parsing the WSDL of the service.

```
#Get the url of the service
String url =
"http://europass.cedefop.europa.eu/soapws/services/PDFWithEuro
passXMLConversionService.PDFWithEuropassXMLConversionServiceHt
tpSoap12Endpoint";
//Get the SOAP client implementation
PDFWithEuropassXMLConversionServiceStub client =
  new PDFWithEuropassXMLConversionServiceStub( url );
//Prepare the input complex type
PDFWithEuropassXMLConversionServiceStub.FileRequest request =
     new
PDFWithEuropassXMLConversionServiceStub.FileRequest();
request.setFile( pdf_exml_as_datahandler );
//Call the method
PDFWithEuropassXMLConversionServiceStub.XMLResponse response =

 client.extractXML( request );
//Get the response
String xml = response.getXml();
```

PHP – SOAP Client

```
#Get the url of the service
$url =
"http://europass.cedefop.europa.eu/soapws/services/PDFWithEuro
passXMLConversionService?wsdl";
#Create the input complex type
$inRequest= new StdClass();
$inRequest->file =
new SoapVar($file_as_string,
XSD_BASE64BINARY,null,null,"xml",$ns);
#Create parameter for the Soap Call
$input = new SoapVar($inRequest, XSD_ANYTYPE, $inType, $ns);
```

```
#Create a new SOAP Client (WSDL mode)
$sc = new SoapClient($url,$soap_options);
#Call the method
$result = $sc-> convertToXMLCV($input);
#Get the response
echo $result->xml;
```

PHP – WS02 WSF

```
#Get the url of the service
$url =
"http://europass.cedefop.europa.eu/soapws/services/
PDFWithEuropassXMLConversionService?wsdl";
#Create the input complex type
$input_array = array( "file" =>
base64_encode($file_as_string)) ;
#Create the SOAP WS Client and Obtain a proxy
$client = new WSClient(array("wsdl"=>$url,
                             "responseXOP"=>true));
$proxy = $client->getProxy();
#Call the method and Get the response
$return_array = $proxy->convertToXMLCV($input_array);
$returned_file = $return_array['xml'] ;
```

3.6.2.2.Example: "Convert Adobe PDF with Europass XML to Europass CV in DOC"

JAVA

Note: The classes used in the example are automatically generated by Apache Axis parsing the WSDL of the service.

```
#Get the url of the service
String url =
"http://europass.cedefop.europa.eu/soapws/services/PDFWithEuro
passXMLConversionService.PDFWithEuropassXMLConversionServiceHt
tpSoap12Endpoint";
//Get the SOAP client implementation
PDFWithEuropassXMLConversionServiceStub client =
  new PDFWithEuropassXMLConversionServiceStub( url );
//Prepare the input complex type
PDFWithEuropassXMLConversionServiceStub.FileLocaleRequest
request = new
PDFWithEuropassXMLConversionServiceStub.FileLocaleRequest();
fileLocale.setFile( pdf_exml_as_datahandler );
fileLocale.setLocale("fr_FR");
//Call the method
PDFWithEuropassXMLConversionServiceStub.FileResponse response
=
 client.convertToMSWordCV( request );
//Get the response file
DataHandler file = response.getFile();
```

PHP – SOAP Client

Note: When XOP:include is used, the response will need to be parsed differently

```
#Get the url of the service
$url =
"http://europass.cedefop.europa.eu/soapws/services/PDFWithEuro
passXMLConversionService?wsdl";
#Create the input complex type
$inRequest= new StdClass();
$inRequest->file =
new SoapVar($xml, XSD_BASE64BINARY,null,null,"xml",$ns);
$inRequest->locale =
  new SoapVar(“en_GB”,XSD_STRING,null, null, "locale", $ns);
#Create parameter for the Soap Call
$input = new SoapVar($inRequest, XSD_ANYTYPE, $inType, $ns);
#Create a new SOAP Client (WSDL mode)
$sc = new SoapClient($url,$soap_options);
#Call the method
$result = $sc->convertToMSWordCV ($input);
#Write the response to a file
$fp=fopen($out_file,"w");
fwrite($fp,$result->file);
fclose($fp);
```

PHP – WS02 WSF

```
#Get the url of the service
$url =
"http://europass.cedefop.europa.eu/soapws/services/
PDFWithEuropassXMLConversionService?wsdl";
#Create the input complex type
$input_array = array( "file" =>
base64_encode($file_as_string),
                      “locale” => “de_DE”) ;
#Create the SOAP WS Client and Obtain a proxy
$client = new WSClient(array("wsdl"=>$url,
                             “responseXOP”=>true));
$proxy = $client->getProxy();
#Call the method and Get the response
$return_array = $proxy-> convertToMSWordCV($input_array);
$returned_file = $return_array['file_encoded'] ;
```

Related APIs: For Java-based implementations, the Apache Axis implementation of SOAP could be considered useful.

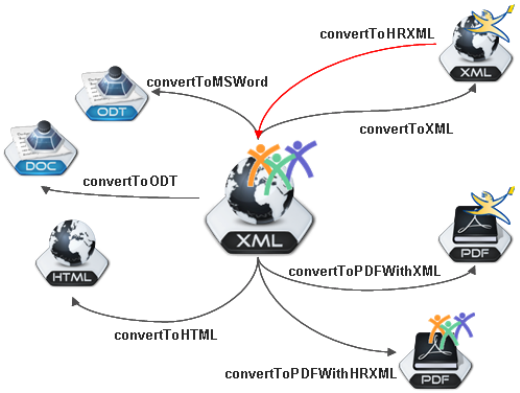
For PHP-based implementations, the PHP Soap Module is necessary.

3.6.3. ***HR-XML – Based*** *(HRXMLConversionService)*

Cue: Convert from HR-XML and to HR -XML

Synopsis: This conversion service returns a Europass Curriculum Vitae or Language Passport document in one of the following formats: Europass XML (EXML), Adobe PDF with Europass XML attachment (PDF+EXML), Adobe PDF with HR-XML attachment (PDF+HR-XML), Microsoft Word 97-2003 (DOC), OpenDocument ver 2 Text (ODT), and HTML.

Moreover, it returns an HR-XML document when the input is a Europass XML document.



Request: The request object is an XMLRequest or an XMLLocaleRequest. Both are custom complex data types consisting of sub-elements. More specifically, XMLRequest consists of a “xml” element”, which holds the input HR-XML document, while XMLLocaleRequest, has the additional element “locale”, which holds the code of the language in which the returned document will be translated. Note that **the locale is required** and when missing a Fault (message “Could not retrieve locale”) will occur:

```
<element name=" XMLRequest"/>
<complexType>
      <sequence>
            <element name="xml" type="xsd:string"/>
      <sequence>
</complexType>

   <element name=" XMLLocaleRequest"/>
<complexType>
      <sequence>
            <element name="xml" type="xsd:string"/>
            <element name="locale" type="xsd:string"/>
      <sequence>
</complexType>
```

In your implementation you will most probably need to create a data structure consisting of these two variables (eg JavaBean in Java or StdClass in PHP5) and pass it as a request parameter to the service.

Response: The response object is either a FileResponse, or an XMLResponse, or an HTMLResponse. All three are custom complex data types consisting of one element. The difference is that in the case of FileResponse this element, “file”, represents base64 encoded binary content, while in the case of XMLResponse and HTMLResponse this element, “xml” and “html” respectively, represents textual content.

```
<element name=" FileResponse"/>
<complexType>
      <sequence>
            <element name="file" type="xsd:base64binary"/>
      <sequence>
</complexType>
<element name=" XMLResponse"/>
<complexType>

      <sequence>
            <element name="xml" type="xsd:string"/>
      <sequence>
</complexType>
<element name=" HTMLResponse"/>
<complexType>
      <sequence>
            <element name="html" type="xsd:string"/>
      <sequence>
</complexType>
```

To consume the response you will most probably need to create a data structure consisting of this variable (eg JavaBean in Java or StdClass in PHP5) and handle the binary or the string content properly.

Please note that “locale” consists of the language and country ISO codes, eg. “fr_FR”, “de_DE.

Methods: The methods provided by this service support the conversions to the various file types.

convertToHRXML (!)

Will return a string representing the HR-XML, **when the input is a Europass XML**.

convertToXMLCV / convertToXMLLP

Will return a string representing the Europass XML of ECV or ELP document respectively

convertToPDFwithXMLCV / convertToPDFwithXMLLP

Will return the base64 binary content of an Adobe PDF file with Europass XML attachment, as ECV or ELP document respectively.

convertToPDFwithHRXMLCV / convertToPDFwithHRXMLLP

Will return the base64 binary content of an Adobe PDF file with HR-XML attachment, as ECV or ELP document respectively.

convertToMSWordCV / convertToMSWordLP

Will return the base64 binary content of a Microsoft Word 97-2003 file as ECV or ELP document respectively.

convertToODTCV / convertToODTLP

Will return the base64 binary content of OpenDocument ver 2 text file as ECV or ELP document respectively.

convertToHTMLCV / convertToHTMLLP

Will return a string representing the HTML of ECV or ELP document respectively

Use Scenario: You maintain a repository of CVs and you are already familiar with the HR-XML specification. By using this service you can offer the option to your users to receive their personal information anytime formatted according to the Europass template.

Examples: The implementation depends on the programming language you will use. Below we provide code snippets for calling the service in Java and PHP5.

3.6.3.1. Example: "Convert HR-XML to Europass CV in PDF"

JAVA

Note: The classes used in the example are automatically generated by Apache Axis parsing the WSDL of the service.

```
#Get the url of the service
String url =
"http://europass.cedefop.europa.eu/soapws/services/HRXMLConver
sionService.HRXMLConversionServiceHttpSoap12Endpoint";
//Get the SOAP client implementation
HRXMLConversionServiceStub client =
 new HRXMLConversionServiceStub( url );
//Prepare the input complex type
HRXMLConversionServiceStub.XMLLocaleRequest request =
 new HRXMLConversionServiceStub.XMLLocaleRequest();
request.setXml( hrxml_to_string );
request.setLocale("fr_FR");
//Call the method
HRXMLConversionServiceStub.FileResponse response =
 client.convertToPDFwithXMLCV( request );
//Get the response file
DataHandler file = response.getFile();
```

PHP – SOAP Client

Note: When XOP:include is used, the response will need to be parsed differently

```
#Get the url of the service
$url =
"http://europass.cedefop.europa.eu/soapws/services/HRXMLConver
sionService?wsdl";
#Create the input complex type
$inRequest= new StdClass();
$inRequest->xml =
  new SoapVar($hrxml, XSD_STRING,null,null,"xml",$ns);
$inRequest->locale =
  new SoapVar(“en_GB”, XSD_STRING,null, null, "locale", $ns);
#Create parameter for the Soap Call
$input = new SoapVar($inRequest, XSD_ANYTYPE, $inType, $ns);
#Create a new SOAP Client (WSDL mode)
$sc = new SoapClient($url,$soap_options);
#Call the method
$result = $sc->convertToPDFWithXMLCV ($input);
#Write the response to a file
$fp=fopen($out_file,"w");
fwrite($fp,$result->file);
fclose($fp);
```

PHP – WS02 WSF

```
#Get the url of the service
$url =
"http://europass.cedefop.europa.eu/soapws/services/HRXMLConver
sionService?wsdl";
#Create the input complex type
$input_array = array( "xml" => $hrxml_to_string,
                      “locale” => “de_DE”) ;
#Create the SOAP WS Client and Obtain a proxy
$client = new WSClient(array("wsdl"=>$url,
                             “responseXOP”=>true));
$proxy = $client->getProxy();
#Call the method and Get the response
$return_array = $proxy-> convertToPDFWithXMLCV($input_array);
$returned_file = $return_array['file_encoded'] ;
```

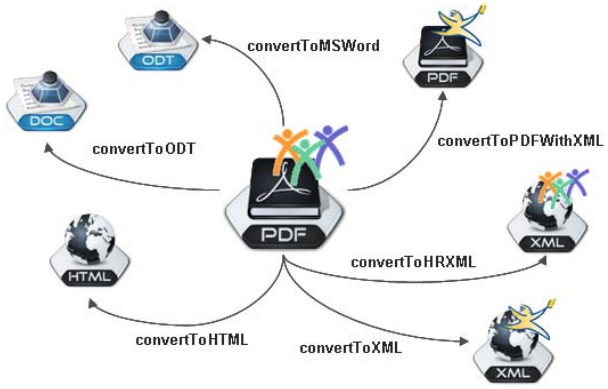
Related APIs: For Java-based implementations, the Apache Axis implementation of SOAP could be considered useful.

For PHP-based implementations, the PHP Soap Module is necessary

3.6.4. ***Adobe PDF+HR-XML – Based*** *(PDFWithHRXMLConversionService)*

Cue: Convert from Adobe PDF with HR-XML attachment

Synopsis: This conversion service returns a Europass Curriculum Vitae or Language Passport document in one of the following formats: Europass XML (EXML), HR-XML (HRXML), Adobe PDF with Europass XML attachment (PDF+EXML), Microsoft Word 97-2003 (DOC), OpenDocument ver 2 Text (ODT), and HTML.



Request: The request object is either a FileRequest or a FileLocaleRequest. Both are custom complex data types consisting of sub-elements. More specifically, FileRequest consists of a "file" element", which holds the input Europass PDF document as base64 binary content, while FileLocaleRequest, has the additional element "locale", which holds the code of the language in which the returned document will be translated. Note that **the locale is required** and when missing a Fault (message "Could not retrieve locale") will occur:

```
<element name=" FileRequest"/>
<complexType>
      <sequence>
            <element name="file" type="xsd:base64binary"/>
      <sequence>
</complexType>

<element name=" FileLocaleRequest"/>
<complexType>
      <sequence>
            <element name="file" type="xsd:base64binary"/>
            <element name="locale" type="xsd:string"/>
      <sequence>
</complexType>
```

In your implementation you will most probably need to create a data structure consisting of these two variables (eg JavaBean in Java or StdClass in PHP5) and pass it as a request parameter to the service.

Response: The response object is either a FileResponse, or an XMLResponse, or an HTMLResponse. All three are custom complex data types consisting of one element. The difference is that in the case of FileResponse this element, "file", represents

base64 encoded binary content, while in the case of XMLResponse and HTMLResponse this element, “xml” and “html” respectively, represents textual content.

```
<element name=" FileResponse"/>
<complexType>
      <sequence>
            <element name="file" type="xsd:base64binary"/>
      <sequence>
</complexType>
<element name=" XMLResponse"/>
<complexType>

      <sequence>
            <element name="xml" type="xsd:string"/>
      <sequence>
</complexType>
<element name=" HTMLResponse"/>
<complexType>
      <sequence>
            <element name="html" type="xsd:string"/>
      <sequence>
</complexType>
```

To consume the response you will most probably need to create a data structure consisting of this variable (eg JavaBean in Java or StdClass in PHP5) and handle the binary or the string content properly.

Please note that “locale” consists of the language and country ISO codes, eg. “fr_FR”, “de_DE.

Methods: **convertToHRXML**

Will return a string representing the extracted HR-XML

convertToXMLCV / convertToXMLLP

Will return a string representing the Europass XML of ECV or ELP document respectively

convertToPDFwithXMLCV / convertToPDFwithXMLLP

Will return the base64 binary content of an Adobe PDF file with Europass XML attachment, as ECV or ELP document respectively.

convertToMSWordCV / convertToMSWordLP

Will return the base64 binary content of a Microsoft Word 97-2003 file as ECV or ELP document respectively.

convertToODTCV / convertToODTLP

Will return the base64 binary content of OpenDocument ver 2 text file as ECV or ELP document respectively.

convertToHTMLCV / convertToHTMLLP

Will return a string representing the HTML of ECV or ELP document respectively

Use Scenario: You may use this service to handle the cases when you receive an HR-XML attached in an Adobe PDF document and you need to return the document formatted according to the Europass template.

Examples: The implementation depends on the programming language you will use. Below we provide code snippets for calling the service in Java and PHP5.

3.6.4.1. Example: "Convert Adobe PDF + HR-XML to Europass XML CV"

JAVA

Note: The classes used in the example are automatically generated by Apache Axis parsing the WSDL of the service.

```
#Get the url of the service
String url =
"http://europass.cedefop.europa.eu/soapws/services/PDFWithHRXM
LConversionService.PDFWithHRXMLConversionServiceHttpSoap12Endp
oint";
//Get the SOAP client implementation
PDFWithHRXMLConversionServiceStub client =
 new PDFWithHRXMLConversionServiceStub( url );
//Prepare the input complex type
PDFWithHRXMLConversionServiceStub.FileRequest request =
 new PDFWithHRXMLConversionServiceStub.FileRequest();
request.setFile( pdf_hrxml_as_datahandler );
//Call the method
PDFWithHRXMLConversionServiceStub.XMLResponse response =
 client.convertToXMLCV( request );
//Get the response
String xml = response.getXml();
```

PHP – SOAP Client

```
#Get the url of the service
$url =
"http://europass.cedefop.europa.eu/soapws/services/PDFWithHRXM
LConversionService?wsdl";
#Create the input complex type
$inRequest= new StdClass();
$inRequest->file =
 new                                SoapVar($file_as_string,
XSD_BASE64BINARY,null,null,"xml",$ns);
#Create parameter for the Soap Call
$input = new SoapVar($inRequest, XSD_ANYTYPE, $inType, $ns);
#Create a new SOAP Client (WSDL mode)
$sc = new SoapClient($url,$soap_options);
#Call the method
$result = $sc->convertToXMLCV ($input);
#Get the response
echo $result->xml;
```

PHP – WS02 WSF

```
#Get the url of the service
$url =
"http://europass.cedefop.europa.eu/soapws/services/PDFWithHRXM
LConversionService?wsdl";
#Create the input complex type
$input_array = array( "file" => $file_as_string) ;
#Create the SOAP WS Client and Obtain a proxy
$client = new WSClient(array("wsdl"=>$url,
                             "responseXOP"=>true));
$proxy = $client->getProxy();
#Call the method and Get the response
$return_array = $proxy-> convertToXMLCV ($input_array);
$returned_file = $return_array['xml'] ;
```

Related APIs: For Java-based implementations, the Apache Axis implementation of SOAP could be considered useful.

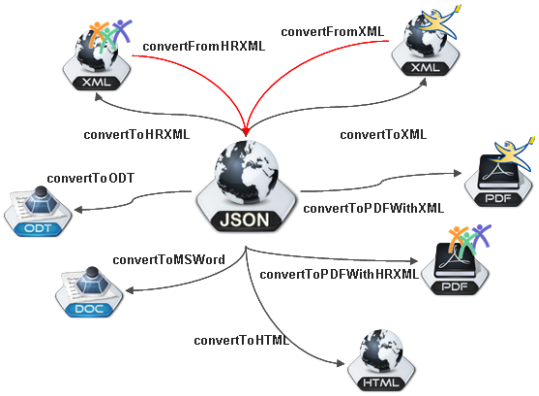
For PHP-based implementations, the PHP Soap Module is necessary

3.6.5. ***Europass JSON – Based*** *(JSONConversionService)*

Cue: Convert from JSON and to JSON

Synopsis: This conversion service returns a Europass Curriculum Vitae or Language Passport document in one of the following formats: Europass XML (EXML), HR-XML (HRXML), Adobe PDF with Europass XML attachment (PDF+EXML), Adobe PDF with HR-XML attachment (PDF+HR-XML), Microsoft Word 97-2003 (DOC), OpenDocument ver 2 Text (ODT), and HTML.

Moreover, it returns an JSON object when the input is a Europass XML or an HR-XML document.



Request: The request object is either a JSONRequest or a XMLRequest, or an JSONLocaleRequest. All three are custom complex data types consisting of sub-elements. More specifically, JSONRequest includes a “json” element”, which holds the input Europass JSON object, while XMLRequest includes a “xml” element”, which holds the input Europass XMLdocument. JSONLocaleRequest has the additional element “locale”, which holds the code of the language in which the returned document will be translated. Note that **the locale is required** and when missing a Fault (message “Could not retrieve locale”) will occur:

```
<element name="JSONLocaleRequest"/>
<complexType>
      <sequence>
            <element name="json" type="xsd:string"/>
            <element name="locale" type="xsd:string"/>
      <sequence>
</complexType>

   <element name="JSONRequest"/>
<complexType>
      <sequence>
            <element name="json" type="xsd:string"/>
      <sequence>
</complexType>

<element name="XMLRequest"/>
<complexType>
      <sequence>
            <element name="xml" type="xsd:string"/>
      <sequence>
```

```
</complexType>
```

In your implementation you will most probably need to create a data structure consisting of these two variables (eg JavaBean in Java or StdClass in PHP5) and pass it as a request parameter to the service.

Response: The response object is either a JSONResponse, or a FileResponse, or an XMLResponse, or an HTMLResponse. All four are custom complex data types consisting of one element. The difference is that in the case of FileResponse this element, “file”, represents base64 encoded binary content, while in the case of JSONResponse, XMLResponse and HTMLResponse this element, “json”, “xml” and “html” respectively, represents textual content.

```
<element name=" FileResponse"/>
<complexType>
      <sequence>
            <element name="file" type="xsd:base64binary"/>
      <sequence>
</complexType>

   <element name=" JSONResponse"/>
<complexType>

      <sequence>
            <element name="json" type="xsd:string"/>
      <sequence>
</complexType>

<element name=" XMLResponse"/>
<complexType>

      <sequence>
            <element name="xml" type="xsd:string"/>
      <sequence>
</complexType>

<element name=" HTMLResponse"/>
<complexType>
      <sequence>
            <element name="html" type="xsd:string"/>
      <sequence>
</complexType>
```

To consume the response you will most probably need to create a data structure consisting of this variable (eg JavaBean in Java or StdClass in PHP5) and handle the binary or the string content properly.

Please note that “locale” consists of the language and country ISO codes, eg. “fr_FR”, “de_DE.

Methods: The methods provided by this service support the conversions to the various file types.

convertFromXMLCV / convertFromXMLLP (!)

Will return a string representing the JSON object **when the input is a Europass XML**.

convertFromHRXML (!)

Will return a string representing the JSON object **when the input is a HR-XML**.

convertToXMLCV / convertToXMLLP

Will return a string representing the Europass XML of ECV or ELP document respectively

convertToHRXML

Will return a string representing the HR-XML of ECV or ELP

convertToPDFwithXMLCV / convertToPDFwithXMLLP

Will return the base64 binary content of an Adobe PDF file with Europass XML attachment, as ECV or ELP document respectively.

convertToPDFwithHRXMLCV / convertToPDFwithHRXMLLP

Will return the base64 binary content of an Adobe PDF file with HR-XML attachment, as ECV or ELP document respectively.

convertToMSWordCV / convertToMSWordLP

Will return the base64 binary content of a Microsoft Word 97-2003 file as ECV or ELP document respectively.

convertToODTCV / convertToODTLP

Will return the base64 binary content of OpenDocument ver 2 text file as ECV or ELP document respectively.

convertToHTMLCV / convertToHTMLLP

Will return a string representing the HTML of ECV or ELP document respectively

Use Scenario: implementation You maintain a Web 2.0 web application and is more straightforward and lightweight for you to manipulate the CV-related information in JSON format. You may use this web service to deliver to your users the CV information to any file format.

Examples: The implementation depends on the programming language you will use. Below we provide code snippets for calling the service in Java and PHP5.

3.6.5.1. *Example: "Convert Europass JSON to Europass XML CV"*

JAVA

Note: The classes used in the example are automatically generated by Apache Axis 2 parsing the WSDL of the service.

```
#Get the url of the service
String url =
"http://europass.cedefop.europa.eu/soapws/services/JSONConvers
ionService. JSONConversionServiceHttpSoap12Endpoint";
//Get the SOAP client implementation
JSONConversionServiceStub client =
  new JSONConversionServiceStub( url );
//Prepare the input complex type
JSONConversionServiceStub. JSONRequest request =
  new JSONConversionServiceStub. JSONRequest();
request.setJson( json_to_string );
//Call the method
JSONConversionServiceStub.XMLResponse response =
 client.convertToXMLCV( request );
//Get the response
String xml = response.getXml();
```

PHP – SOAP Client

```
#Get the url of the service
$url =
"http://europass.cedefop.europa.eu/soapws/services/JSONConvers
ionService?wsdl";
#Create the input complex type
$inRequest= new StdClass();
$inRequest->json =
  new                                    SoapVar($json_to_string
,XSD_STRING,null,null,"xml",$ns);
#Create parameter for the Soap Call
$input = new SoapVar($inRequest, XSD_ANYTYPE, $inType, $ns);
#Create a new SOAP Client (WSDL mode)
$sc = new SoapClient($url,$soap_options);
#Call the method
$result = $sc->convertToXMLCV ($input);
#Get the response
echo $result->xml;
```

PHP – WS02 WSF

```
#Get the url of the service
$url =
"http://europass.cedefop.europa.eu/soapws/services/JSONConvers
ionService?wsdl";
#Create the input complex type
$input_array = array( "json" => $json_to_string) ;
#Create the SOAP WS Client and Obtain a proxy
$client = new WSClient(array("wsdl"=>$url,
                             "responseXOP"=>true));
$proxy = $client->getProxy();
#Call the method and Get the response
$return_array = $proxy-> convertToXMLCV ($input_array);
$returned_file = $return_array['xml'] ;
```

3.6.5.2.Example: “Convert Europass JSON to Europass CV in PDF”

JAVA

Note: The classes used in the example are automatically generated by Apache Axis parsing the WSDL of the service.

```
#Get the url of the service
String url =
"http://europass.cedefop.europa.eu/soapws/services/JSONConvers
ionService. JSONConversionServiceHttpSoap12Endpoint”;
//Get the SOAP client implementation
JSONConversionServiceStub client =
  new JSONConversionServiceStub( url );
//Prepare the input complex type
JSONConversionServiceStub. JSONLocaleRequest request =
  new JSONConversionServiceStub.JSONLocaleRequest();
request.setJson( json_to_string );
request.setLocale("fr_FR");
//Call the method
JSONConversionServiceStub.FileResponse response =
 client.convertToPDFWithXMLCV( request );
//Get the response file
DataHandler file = response.getFile();
```

PHP – SOAP Client

Note: When XOP:include is used, the response will need to be parsed differently

```
#Get the url of the service
$url =
"http://europass.cedefop.europa.eu/soapws/services/JSONConvers
ionService?wsdl";
#Create the input complex type
$inRequest= new StdClass();
$inRequest->json =
  new SoapVar($json_to_string,XSD_STRING,null,null,"xml",$ns);
$inRequest->locale =
  new SoapVar(“en_GB”,XSD_STRING,null, null, "locale", $ns);
#Create parameter for the Soap Call
$input = new SoapVar($inRequest, XSD_ANYTYPE, $inType, $ns);
#Create a new SOAP Client (WSDL mode)
$sc = new SoapClient($url,$soap_options);
#Call the method
$result = $sc->convertToPDFWithXMLCV ($input);
#Write the response to a file
$fp=fopen($out_file,"w");
fwrite($fp,$result->file);
fclose($fp);
```

PHP – WS02 WSF

```
#Get the url of the service
$url =
"http://europass.cedefop.europa.eu/soapws/services/JSONConvers
ionService?wsdl";
#Create the input complex type
$input_array = array( "json" => $json_to_string,
                      “locale” => “de_DE”) ;
#Create the SOAP WS Client and Obtain a proxy
$client = new WSClient(array("wsdl"=>$url,
                             “responseXOP”=>true));
```

```
$proxy = $client->getProxy();
#Call the method and Get the response
$return_array = $proxy-> convertToPDFWithXMLCV ($input_array);
$returned_file = $return_array['file_encoded'] ;
```

Related APIs: For Java-based implementations, the Apache Axis implementation of SOAP could be considered useful.

For PHP-based implementations, the PHP Soap Module is necessary.

4. EUROPASS SOAP SERVICES EXAMPLES

4.1. Java Examples

4.1.1. Generate Necessary Code from WSDL

The easiest way to make a call to the Europass SOAP services is to utilise the WSDL and generate automatically the necessary Java Code (SOAP clients and Input/ Output Objects).

We recommend using the Apache Axis 2 for Java. Documentation and user guide for creating clients can be found online[1]. There also exist various tools and extension for IDEs[2].

4.1.2. Utilise Europass Java Client

Alternatively, you may leverage the Java Client provided by Europass, which allows you to call the services by passing the necessary parameters.

Simply, include the **europass-soapws-client.jar** as library to your project. Please make sure that your project runs on Java 1.6.

An example of how to use the library and create an Adobe PDF with Europass XML attachment from a Europass XML file is given below:

```
import eu.europa.cedefop.europass.EXMLConversionClientTool;
… … …
String europassSoapServicesUrl =
 "https://europass.cedefop.europa.eu/soapws/services";

String pathToEuropassXML = "C:\\IN\\CV_EuropassXML_2.0.xml";

String locale = "fr_FR";

String outputDirectory = "C:\\OUT";

EXMLConversionClientTool tool = new
EXMLConversionClientTool();

tool.initTool( europassSoapServicesUrl,
               pathToEuropassXML,
               locale,
               outputDirectory );

try {
  tool.convertToPDFwithXMLCV();
} catch (Exception e){
  //catch the exception error
}
```

[1] http://axis.apache.org/axis2/java/core/docs/userguide-creatingclients.html#createclients

[2] http://axis.apache.org/axis2/java/core/tools/index.html

4.2. PHP Examples

4.2.1. Prerequisites

PHP, version 5.3.1 or higher – The necessary SOAP module is included by default in this version.

Important Note: The default SOAP module of PHP 5 does not support MTOM/XOP optimisation out of the box. Therefore you may find useful the WSO2 Web Services Framework for PHP.

4.2.2. Examples

Utilising PHP 5 SOAP CLIENT

The Soap Client of the default Soap PHP library does not handle multipart responses so we need to implement an extension of that client that can handle multiparts. You will have to use include this file in any case that Europass Web Services respond with a multipart message.

In that case, the user will have to add the following line to include the file:

include("mySoapClient.php");

To create the Soap Client, the use will have to use the following class:

mySoapClient(<WSDL URL>, <Soap Client Options>)

```
<?php
class MySoapClient extends SoapClient
{
    public function __doRequest($request, $location, $action,
$version, $one_way = 0) {
        $response = parent::__doRequest($request, $location,
$action, $version, $one_way);
        $response = preg_replace('#^.*(<\?xml.*>)[^>]*$#s',
'$1', $response);
        return $response;
    }
}
?>
```

Example: Convert Europass JSON to Europass XML

```
<?php
include("mySoapClient.php");
#WSDL of the server for the given Conversion Type
$url =
"http://europass.cedefop.europa.eu/europassws/services/JSONCo
nversionService?wsdl";
#Namespace
$ns = "http://services.europass.cedefop.europa.eu";
#Operation to call
$operation = "convertToXMLCV";
#Input file
$in = "D:/Test/europass_json_cv.json";
#Output file
$out = "D:/Test/Europass_xml_cv.xml";
#Output file type
$type = "xml";
```

```
$json_to_string = "";
#Read json file to a string
$file=fopen($in,"r");
while(!feof($file)) {
  $json_to_string = $json_to_string.fgets($file);
}
fclose($file);

#Create the two input values
$json =
  new SoapVar($json_to_string,
XSD_ANYTYPE,null,null,$var1,$ns);

#Create the input complex type
$inRequest= new StdClass();
$inRequest->json = $json;

#Enable trace for the web service call
$soap_options = array(
  'trace'       => 1,
  'exceptions'  => 1 );

try {
  #Create parameter for the Soap Call
  $input = new SoapVar($inRequest, XSD_ANYTYPE, $inType,
$ns);
  #Create a new SOAP Client (WSDL mode)
  $sc = new MySoapClient($url,$soap_options);
      #Call the method
  $result = $sc->$operation($input);
  #Write the response to a file
  $fp=fopen($out,"w");
  fwrite($fp,$result->$type);
  fclose($fp);
} catch (SoapFault $sf) {
  echo "SOAP Fault: ".$sf->getMessage()."<br>\n";
}
?>
```

Utilising WS02 Web Services Framework

Example: Convert Europass XML to HTML

```
<?php
#WSDL of the server for the given Conversion Type
$url    =
"http://europass.cedefop.europa.eu/europassws/services/Europa
ssXMLConversionService?wsdl";

#Operation to call
$operation = "convertToHTMLCV";
#Locale of the input file
$locale="en_GB";
#Input file
$in = "D:/Test/europass_exml_cv.xml";
#Output file
$out = "D:/Test/Europass_html_cv.html";
#Output file type
$type = "html";

$xml_to_string = "";
#Read xml file to a string
$file=fopen($in,"r");
while(!feof($file)) {
```

```
  $xml_to_string = $xml_to_string.fgets($file);
}
fclose($file);

#Create the request
$input_array = array( "xml" => $xml_to_string,
                      "locale" => $locale);
#Create the client
try {
  $client = new WSClient(array("wsdl"=>$url,
                                    “responseXOP”=>true));
  #Obtain a proxy and get the response
  $proxy = $client->getProxy();
  $return_array = $proxy->convertToHTMLCV($input_array);
  echo $return_array['html']."\n";
} catch (Exception $e) {
  if ($e instanceof WSFault) {
        echo "Soap Fault: %s\n". $e->Reason . "br/>";
    } else {
        echo "Message = %s\n" . $e->getMessage() . "br/>";
    }
}
?>
```

Limitations:

We noticed that the returned binary content of the response resides under the key “file_encoded” rather than “file’ as expected from the WSDL.

Moreover, the returned encoded file contains carriage returns and possibly other non base 64 encoded characters, which make the decoding cumbersome.