
E-Business Technologies – Lab – Summer Term 2017

Assignment 3:
XML Processing with Java
Implementation of a simple
E-Business System for Wholesalers
— 40 Points —

1 Overall Scenario

Assume, you are a wholesaler for ICT products and you have to implement an E-Business system with product data management functionality. This E-Business system should be able to import product catalogs from your suppliers. Furthermore, the E-Business system should be able to export your product catalog to your customers. Fig. 1 depicts the overall scenario.

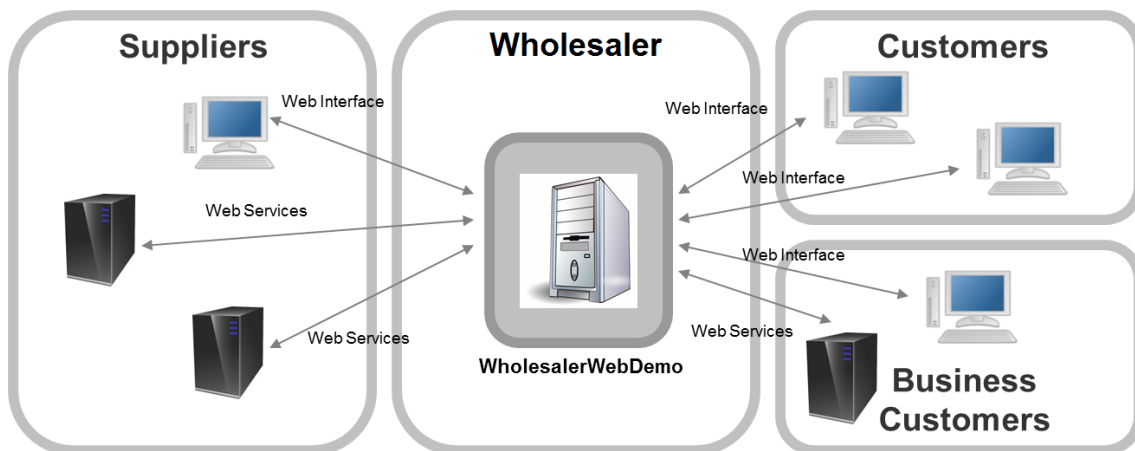


Figure 1: Overall scenario.

2 Initial Situation

You already have a basic E-Business system running (the *WholesalerWebDemo* application) which is based on the *E-Business Software Framework* (EBSF).

The EBSF implements the data management layer (based on a DBS and an OR mapping) and a simple business logic layer for E-Business applications. It also offers an application programming interface (API) for the business logic layer.¹

3 Assignment

Design and implement the following **import / export functionality** for your E-Business system, meeting the following functional requirements:

- Import product catalogs (*simple BMEcat XML format*) into your existing E-Business system.
- Export articles from your E-Business system into a product catalog (*simple BMEcat XML format*).
- Export articles from your E-Business system as a *simple XHTML* document.
- The export functionality has to offer *two options*:
 - Export of the whole product catalog (i.e., all articles in your database).
 - Export of only those articles where the short description matches a given search string, i.e., case-insensitive substring matching (not exact matching).
- Export and import functionality should be available via a *Web GUI*.

Non-functional requirements:

- The language of the Web GUI has to be *English*.
- Only use the *Java APIs* (JAXP) for all XML processing (no JDOM, dom4j, etc.).
- Use the *Document Object Model* (DOM) as *API* to implement the import and export functionality.
- Use the *XML validation package* from Java to validate XML documents against the simple BMEcat XML format etc.
- Usage of the *XPath package* from Java may ease DOM processing for the product catalogue import.
- Use *XSLT* together with the *XML transformation packages* from Java to create the (valid) XHTML output.
- Comment your code (classes, properties, methods etc.) *sufficiently* in *English*, preferably using JavaDoc.

¹You can find detailed information on *Moodle*.

4 Hints

All necessary resources, e.g., software, installation instructions, documentations, the XML schema of the simple BMEcat XML format, a sample XML product catalog in the simple BMEcat XML format, etc. are available on Moodle.

- Create a supplier in your database first (e.g., using SQL / a database frontend tool) to enable product catalog import for this supplier.
- Design and implement your import and export functionality in a modular way, so you are able to (re-)use it in several contexts, e.g., within a Web application or a Desktop application or to implement SOAP- or REST-based Web Services later on.
- The upload functionality via the Web GUI *could* be realized, e.g., with Apache File Uploader² (packages: `commons-fileupload-x.jar` and `commons-io-x.jar`, where 'x' stands for the latest version).
- It is not necessary to use temporary files on the server.
- *If you decide to use the WholesalerWebDemo application as a basis for your solution*, use the following two packages (within your WholesalerWebDemo project):
 - `de.htwg_konstanz.ebut.main` → for your import/export business logic
 - `de.htwg_konstanz.ebut.action` → for your action classes (implementing the `IAction` interface)

Do not forget to announce your action Java package to the `ControllerServlet` class! Use *no* additional servlet.

- Use the W3C Markup Validation Services³ to validate your XHTML output.

A possible "architecture" for your catalog import and export is depicted in Fig. 2.

5 Acceptance

For the successful acceptance of your solution for assignment 3, a *checklist* is defined. It will be used for your demonstration and to assess your solution. The following topics are included, among others, in the checklist:

- Upload/download via Web GUI
- XML import/export with DOM, validation, XHTML export, full/selective export, etc.
- Update semantics for catalog import
- User experience (incl. failure handling, error messages, warnings, etc.)
- Comments included in your source code

²commons.apache.org/proper/commons-fileupload/

³validator.w3.org/

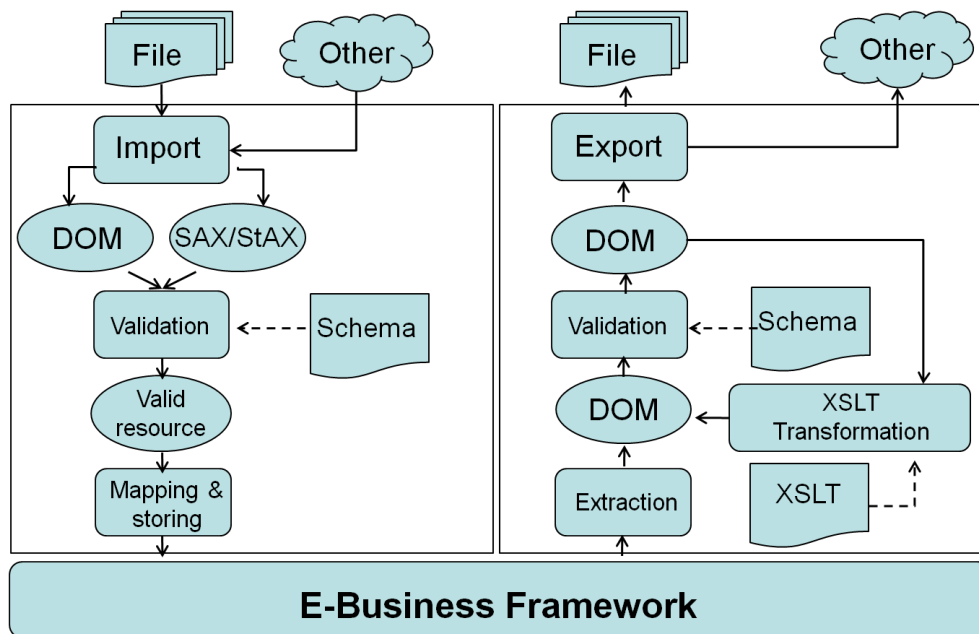


Figure 2: Possible "architecture" for catalog import and export.

Hard **Deadline** for the demonstration of your solution to assignment 3 is **June 13**.