

# Integration von ERP-Systemen und ePages 6 mit Webservices

- Stand 10/2011 -

Einleitung.....	2
Grundlagen .....	2
Überblick Datenaustausch .....	3
Ablauf .....	4
Verbindungstest .....	4
Testen mit Beispieldaten .....	4
Daten-Mapping .....	5
Definition gewünschter Funktionen .....	5
Beispiel für Datenübertragung .....	7
Verbindungstest .....	7
Produkt anlegen .....	7

# Einleitung

Das Dokument beschreibt die Grundlagen und eine mögliche Vorgehensweise für die Integration von ePages und ERP-Systemen.

Folgende Voraussetzungen müssen erfüllt sein, um die Systeme erfolgreich zu integrieren:

- Kenntnisse über die Programmierung und Anwendung von Webservices
- Kenntnis der Shop-URL und der Zugangsdaten zur Shop-Administration
- Für den Shoptyp muss der Zugriff per Webservices freigeschaltet sein (z.B. durch Verwendung der Shoptypen "ePages Flex" oder "ePages Enterprise")
- Kenntnisse über die Datenbankabfrage aus dem ERP-System
- Netzwerkverbindung zwischen ERP-System und ePages-Shop-System

Hinweis und Erfahrungen bezüglich ePages und Webservices werden auch im ePages-Developer Forum diskutiert: <http://www.epages.com/forum/>.

## Grundlagen

Bei der Anbindung des ePages-Shop-Systems an ein ERP-System kommt es zu folgender funktionalen Aufteilung:

### ERP-System

Erstellung, Bearbeitung und Verwaltung der Daten wie Produkte, Kunden und Bestellungen

### ePages-System

Funktionen und Abläufe für Präsentation und Verkauf der Produkte über Online-Kanäle – Bestellprozess, Kundenkommunikation, Kundenregistrierung usw.

Das ERP-System ist in aller Regel das führende System. Datenänderung und Datenverarbeitung (z.B. Bestellungen, Kunden) werden im ERP-System vorgenommen. Der Shop fungiert in diesem Fall quasi nur als Satellit. Im Shop können aber Aktionen wie eBay- und Amazon-Verkauf gestartet werden insofern das ERP-System diese Funktionen nicht bietet.

ePages ist vorbereitet für die Integration in ERP-Systeme. Das System bietet fertige API auf Basis von Webservices an. Alle notwendigen Business-Objekte wie Produkte, Kunden usw. sind verfügbar. Für den Standarddatenaustausch ist ePages-seitig kein Entwicklungsaufwand notwendig.

Einen Überblick über alle verfügbaren ePages-Webservice finden Sie hier: <http://www.epages.com/WebService/WebServices.html>. In der Liste sind alle Versionen aufgeführt. Die URN gibt an, wie aktuell ein Webservice ist. Es können immer alle Versionen eines Webservices benutzt werden, auch die Funktionalität älterer Version ist gewährleistet. Neue Versionen stellen neue Funktionen zur Verfügung und enthalten aber auch die Funktionen der Vorgängerversionen.

Auf Seiten des ERP-Systems müssen die entsprechenden Webservice-Clients entwickelt werden. Der Webservice-Client kann dabei ein Dienst innerhalb des ERP-Systems oder ein separates Programm "zwischen" ERP-System und Shop sein. ePages liefert für ePages- Webservices entsprechende Beispiel-Clients in den Programmiersprachen Java, .net und PERL. Diese Beispiele sollen Ihnen bei der Entwicklung Ihrer Integration helfen.

Um neue Bestellung aus dem Shop-System abzuholen, muss ein Spooler auf Seiten des ERP-Systems implementiert werden. Der Shop schickt neue Bestellungen nicht von selbst zum ERP-System, das ERP-System muss diese periodisch abfragen (z.B. jede Stunde).

# Überblick Datenaustausch

Folgende Daten werden im Allgemeinen zwischen Shop- und ERP-System ausgetauscht:

## Produkte

- Anlegen und Pflegen im ERP-System, Export zu ePages
- Export der Lagerbestände zu ePages
- Export der kundenspezifischen Preise zu ePages
- Export von Bildern
- Export von Produkttypen (Kategorisierung)

## Kategorien

- Anlegen in ePages, Export in das ERP-System
- Anlegen im ERP-System, Export zu ePages
- Produktzuordnung zu Kategorien im ERP-System

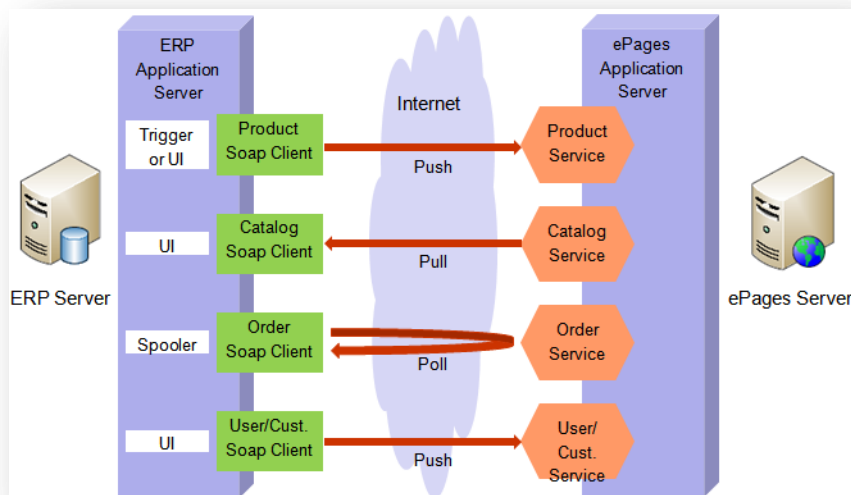
## Kunden

- Anlegen und Verwalten der Kunden im ERP-System, Export zu ePages
- Anlegen neuer Kunden im ERP-System aus den Bestelldaten

## Bestellungen

- Abrufen neuer Bestellungen aus ePages durch das ERP-System
- Verwaltung und Bearbeitung im ERP-System
- Export von Bestellstatusänderungen zu ePages

Die folgende Abbildung zeigt den Datenaustausch zwischen den Systemen:



Prinzipiell wird der Datenaustausch durch das ERP-System angestoßen.

# Ablauf

Erfahrungen aus der Praxis zeigen, dass Sie bei der Anbindung von ePages an ein ERP-System wie folgt vorgehen sollten:

1. Verbindungstest
2. Testen mit Beispieldaten
3. Daten-Mapping
4. Definition gewünschter Funktionen
5. Implementation und Test

Einige dieser Schritte werden in den folgenden Abschnitten näher erläutert.

## Verbindungstest

Im ersten Schritt soll überprüft werden, ob die Verbindung zwischen den beiden System eingerichtet ist. Das bedeutet man muss per Webservice auf das Shopsystem zugreifen können.

Ziel ist es also, die Verbindung zwischen Webservice-Client auf Seite des ERP-Systems und dem ePages-Shop herzustellen und zu testen.

Dazu laden Sie sich die Demo-Clients von dieser Seite:

<ftp://epages-software.de/pub/products/epages6/current/WebServiceExamples/>

Wählen Sie einen Webservice, z.B. *PageCacheTestCase.java*.

Passen Sie in *WebServiceConfiguration.java* die Verbindungsparameter an:

```
// public final String WEBSERVICE_URL = getWebserviceURL();  
...  
public final static String WEBSERVICE_URL = http://[server name]/epages/Store.soap";  
public final static String WEBSERVICE_LOGIN = "/Shops/[shop name]/Users/[user name]";  
public final static String WEBSERVICE_PASSWORD = "[user_password]";  
...
```

Kompilieren Sie die Dateien neu und starten Sie den Webservice:

```
%JAVA% -cp bin;lib/* org.junit.runner.JUnitCore  
de.epages.WebServices.PageCacheService.PageCacheTestCase
```

In der Ausgabe wird Ihnen angezeigt, ob die Verbindung erfolgreich hergestellt werden konnte.

Für die Webservices können Sie auch einen eigenen Administrationszugang anlegen. Dadurch sind die Webservices vom Shop-Administrator abgekoppelt. Falls für diesen die Zugangsdaten geändert werden, müssen diese Änderungen nicht für die Webservice nachgezogen werden.

Es ist empfehlenswert, die Verbindung über einen Proxy herzustellen. Dieser erleichtert die Protokollierung und Auswertung der Verbindungsdaten sowie die Kommunikation in Support-Fällen.

## Testen mit Beispieldaten

Wenn die Verbindung erfolgreich hergestellt ist, sollten Sie die bereitgestellten Webservice-Beispiele testen.

Diese Testphase ermöglicht einfaches Einarbeiten in verfügbare Funktionen. Sie bekommen einen Überblick, wie welche Daten zwischen Webservice und Shop-System übertragen werden und welche Funktionen verfügbar sind.

Wir empfehlen, dazu den ePages-DemoShop auf dem Shop-System anzulegen, weil die Beispiele auf diesen Shop abgestimmt sind.

Die Beispieldaten der Webservice-Clients sind als konstante Strings im Code hinterlegt.

## Daten-Mapping

Die Beispiele beschreiben Datenaustausch zwischen Webservice-Client und ePages. Offen ist die Frage, wie die Daten zwischen Client und Datenbank des ERP-Systems ausgetauscht werden. Datenmodell und Datenorganisation in der ERP-Datenbank sind mit Sicherheit anders als in der ePages Datenbank.

Deshalb muss ein Mapping der Daten stattfinden.

Welche Daten wie und mit welchen Typ nach ePages übertragen werden, wird in den WSDL und XSD-Dateien der Webservices beschrieben, siehe <http://www.epages.com/WebService/WebServices.html>.

Der Webservice-Client muss die Daten aus der Datenbank des ERP-Systems auslesen, auf die entsprechende Datenstruktur, die im WSDL definiert ist, mappen und dann diese Daten übertragen.

Ein Beispiel für das Daten-Mapping sehen Sie in der folgenden Abbildung:

<b>Datamapping</b>				
<b>Products</b>				
ERP			ePages	
Table	Field	API-Value	WebService	Field
OITM	ItemCode	Item_Code	products	Alias
OITM	ItemName	Item_Name	products	Name.Value
				Name.Language==de
OITM	UserText	User_Text	products	Description.Value
				Description.Language==de
EPA_CONFIG	p_pl		products	Prices.Value
				Prices.Currency==EU
				Prices.TaxModel==1
OITM	SWeight1	SalesUnitWeight	products	Weight
OITM	MinLevel	MinInventory	products	StockLevelAlert
OITW	MinStock	MinimalStock		
OITM	SalUnitMsr	SalesItemsPerUnit	products	Minorder
OITM	SuppCatNum	Mainsupplier	products	
EPA_TAXCLASS_MAP	epkey	---	products	Taxclass
EPA_MANUFACT_MAP	epkey	---	products	Manufacture
---	---	---	--	--

## Definition gewünschter Funktionen

Ausgehend von den Anforderungen kann nun definiert werden, welche Funktionen durch die Webservices bereitgestellt werden sollen. Die Praxis hat gezeigt, dass meistens die folgenden Grundfunktionen benötigt werden:

### Initialer Import in das Shop-System

Damit werden die vorhandenen Basisdaten aus dem ERP-System in das Shop-System übertragen.

### Update bei Datenänderungen

Datenänderungen müssen zwischen den System abgeglichen werden. Da das ERP-System das führende System ist, müssen die Webservices Daten schicken und abholen können (push/pull).

**Hinweis:** Nach Änderungen am Inhalt muss der Seiten-Cache zurückgesetzt werden, um die aktuellen Daten anzuzeigen. Dafür kann der Webservice " PageCacheService" verwendet werden.

## Steuerung des Datenaustausches

Die Webservices müssen die Daten sowohl in festgelegten Intervallen als auch auf Anforderung übertragen können.

Weiterführend müssen Sie definieren, für welche Businessobjekte welche Daten ausgetauscht werden. Aus dem Abgleich zwischen den verfügbaren ePages-Webservices und Ihren Anforderungen ergibt sich der Bedarf an neu zu implementieren Webservices.

Bei der Definition und Nutzung bzw. Implementierung der geforderten Funktionen empfiehlt sich ein Vorgehen in verschiedenen Phasen, z. B.:

- Phase 1: Übertragung und Aktualisierung von Produkt- und Bestelldaten
- Phase 2: Übertragung der Kundendaten
- Phase 3: Übertragung von Lager- und Preisinformationen in Echtzeit

Der Datenaustausch in Echtzeit ist vor allem performance-kritisch und zwar für das ERP-System. Müssen z.B. für eine Listenanzeige im Shop mit mehreren Produkten für jedes einzelne Produkt kundenspezifische Preise oder Lagerbestände im ERP-System ermittelt werden, kann das zu langen Wartezeiten im Shop führen.

# Beispiel für Datenübertragung

Im Folgenden sehen Sie anhand eines praktischen Beispiels, wie mit Hilfe der Demo-Clients eine Verbindung hergestellt und ein Produkt im Ziel-Shop angelegt wird.

Im Beispiel werden die Java-Versionen der Clients verwendet. Vorausgesetzt werden:

- Java Kenntnisse
- Java-Umgebung zum Kompilieren der Klassen
- DemoShop auf dem ePages-Zielsystem

Die Tests können mit einem anderen Shop ausgeführt werden. Dann müssen aber mehr Änderungen als die beschriebenen vorgenommen werden.

## Verbindungstest

Der Verbindungstest wird wie folgt durchgeführt:

1. Laden Sie das Paket mit den Demo-Clients herunter und entpacken Sie es z.B. in das Verzeichnis *WS\_DemoClients*:

[ftp://epages-software.de/pub/products/epages6/6.12/AddOns/WebServiceExamples.6.12.0/WS\\_DemoClients.zip](ftp://epages-software.de/pub/products/epages6/6.12/AddOns/WebServiceExamples.6.12.0/WS_DemoClients.zip)

2. Passen Sie die Verbindungsparameter an:

Öffnen Sie dazu die Datei

*WS\_DemoClients\java\src\de\epages\WebServices\WebServiceConfiguration.java*

Passen Sie die Parameter wie folgt an:

```
// public final String WEBSERVICE_URL = getWebserviceURL();  
public final static String WEBSERVICE_URL =  
"http://whitelinux.epages.de/epages/Store.soap";
```

Der Server *whitelinux.epages.de* ist ein öffentliches Testsystem, welche für die Webservice-Tests genutzt werden kann.

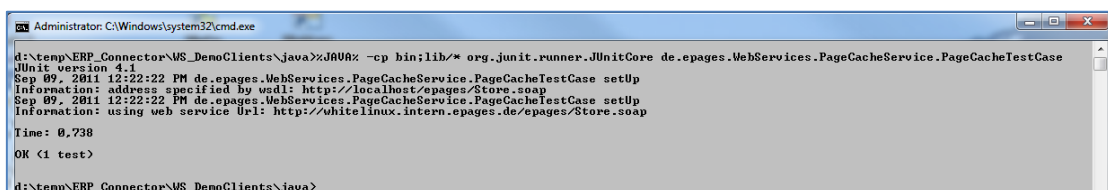
3. Kompilieren Sie die Datei neu im Verzeichnis *WS\_DemoClients\java*:

```
%JAVAC% -cp bin;lib/* -d bin  
src\de\epages\WebServices\WebServiceConfiguration.java
```

4. Testen Sie die Verbindung

```
%JAVA% -cp bin;lib/* org.junit.runner.JUnitCore  
de.epages.WebServices.PageCacheService.PageCacheTestCase
```

Die Ausgabe bei erfolgreichem Test sieht wie folgt aus:



```
Administrator: C:\Windows\system32\cmd.exe  
d:\temp\ERP_Connector\WS_DemoClients\java>java -cp bin;lib/* org.junit.runner.JUnitCore de.epages.WebServices.PageCacheService.PageCacheTestCase  
JUnit version 4.1  
Sep 09, 2011 12:22:22 PM de.epages.WebServices.PageCacheService.PageCacheTestCase setUp  
Information: address specified by wsdl: http://localhost/epages/Store.soap  
Sep 09, 2011 12:22:22 PM de.epages.WebServices.PageCacheService.PageCacheTestCase setUp  
Information: using web service url: http://whitelinux.intern.epages.de/epages/Store.soap  
Time: 0.738  
OK (1 test)  
d:\temp\ERP_Connector\WS_DemoClients\java>
```

## Produkt anlegen

Um die Datenübertragung zu testen, legen Sie ein Produkt im DemoShop auf dem Zielsystem an:

1. Passen Sie den ProductService für den Test an:

Öffnen Sie dazu die Datei

*WS\_DemoClients\java\src\de\epages\WebServices\ProductService\ProductTestCase.java*

2. Kommentieren Sie folgende Zeilen im unteren Teil der Datei aus:

```
//      testDelete();  
//      testExists(false);
```

Das ist notwendig, damit Sie das Produkt auch im Shop sehen können. Anderenfalls wird es gleich wieder gelöscht, da das Löschen Teil des Testfalles ist.

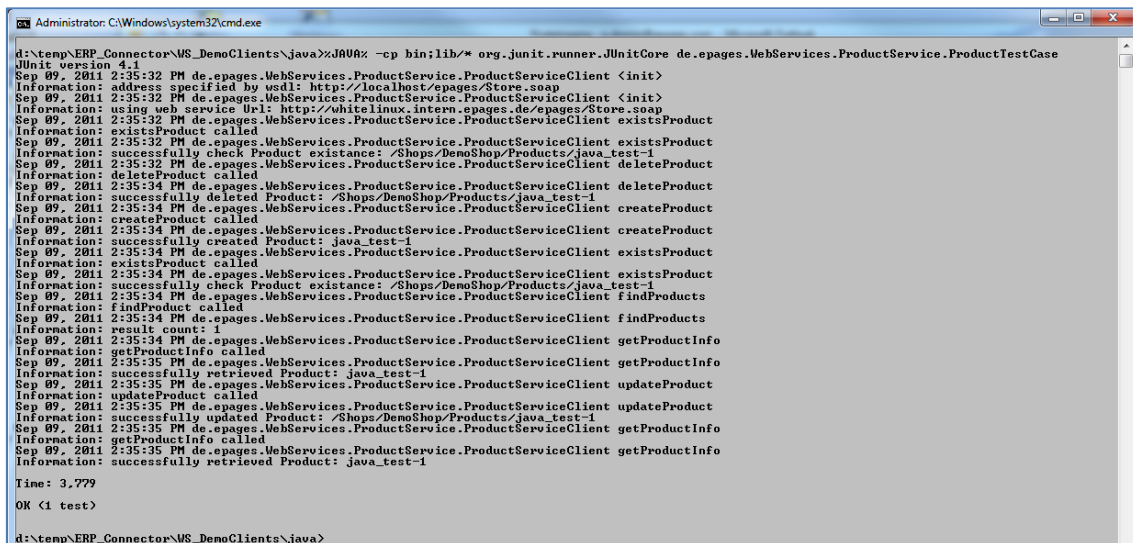
3. Kompilieren Sie die Datei im Verzeichnis *WS\_DemoClients\java*:

```
%JAVAC% -cp bin;lib/* -d bin  
src\de\epages\WebServices\ProductService\ProductTestCase.java
```

4. Starten Sie den Testfall:

```
%JAVA% -cp bin;lib/* org.junit.runner.JUnitCore  
de.epages.WebServices.ProductService.ProductTestCase
```

5. Prüfen Sie die Ausgabe auf der Kommandozeile:



```
Administrator: C:\Windows\system32\cmd.exe  
d:\temp\ERP_Connector\WS_DemoClients\java>%JAVA% -cp bin;lib/* org.junit.runner.JUnitCore de.epages.WebServices.ProductService.ProductTestCase  
JUnit version 4.1  
Sep 09, 2011 2:35:32 PM de.epages.WebServices.ProductService.ProductServiceClient <init>  
Information: address specified by wsdl: http://localhost/epages/Store.soap  
Sep 09, 2011 2:35:32 PM de.epages.WebServices.ProductService.ProductServiceClient <init>  
Information: using web service Url: http://white.linux.intern.epages.de/epages/Store.soap  
Sep 09, 2011 2:35:32 PM de.epages.WebServices.ProductService.ProductServiceClient existsProduct  
Information: existsProduct called  
Sep 09, 2011 2:35:32 PM de.epages.WebServices.ProductService.ProductServiceClient existsProduct  
Information: successfully check Product existence: /Shops/DemoShop/Products/java_test-1  
Sep 09, 2011 2:35:32 PM de.epages.WebServices.ProductService.ProductServiceClient deleteProduct  
Information: deleteProduct called  
Sep 09, 2011 2:35:34 PM de.epages.WebServices.ProductService.ProductServiceClient deleteProduct  
Information: successfully deleted Product: /Shops/DemoShop/Products/java_test-1  
Sep 09, 2011 2:35:34 PM de.epages.WebServices.ProductService.ProductServiceClient createProduct  
Information: createProduct called  
Sep 09, 2011 2:35:34 PM de.epages.WebServices.ProductService.ProductServiceClient createProduct  
Information: successfully created Product: java_test-1  
Sep 09, 2011 2:35:34 PM de.epages.WebServices.ProductService.ProductServiceClient existsProduct  
Information: existsProduct called  
Sep 09, 2011 2:35:34 PM de.epages.WebServices.ProductService.ProductServiceClient existsProduct  
Information: successfully check Product existence: /Shops/DemoShop/Products/java_test-1  
Sep 09, 2011 2:35:34 PM de.epages.WebServices.ProductService.ProductServiceClient findProducts  
Information: findProduct called  
Sep 09, 2011 2:35:34 PM de.epages.WebServices.ProductService.ProductServiceClient findProducts  
Information: result count: 1  
Sep 09, 2011 2:35:34 PM de.epages.WebServices.ProductService.ProductServiceClient getProductInfo  
Information: getProductInfo called  
Sep 09, 2011 2:35:35 PM de.epages.WebServices.ProductService.ProductServiceClient getProductInfo  
Information: successfully retrieved Product: java_test-1  
Sep 09, 2011 2:35:35 PM de.epages.WebServices.ProductService.ProductServiceClient updateProduct  
Information: updateProduct called  
Sep 09, 2011 2:35:35 PM de.epages.WebServices.ProductService.ProductServiceClient updateProduct  
Information: successfully updated Product: /Shops/DemoShop/Products/java_test-1  
Sep 09, 2011 2:35:35 PM de.epages.WebServices.ProductService.ProductServiceClient getProductInfo  
Information: getProductInfo called  
Sep 09, 2011 2:35:35 PM de.epages.WebServices.ProductService.ProductServiceClient getProductInfo  
Information: successfully retrieved Product: java_test-1  
Time: 3.779  
OK (1 test)  
d:\temp\ERP_Connector\WS_DemoClients\java>
```

6. Prüfen Sie im DemoShop, ob das Produkt *java\_test-1* in der Produktliste angezeigt wird.

Bei Problemen richten Sie Ihre Anfragen an: [support@epages.com](mailto:support@epages.com).



Die in diesem Dokument enthaltenen Informationen können jederzeit ohne Benachrichtigung geändert werden.

Dieses Werk ist einschließlich aller seiner Teile urheberrechtlich geschützt. Alle Rechte sind ausdrücklich vorbehalten, einschließlich der Rechte auf Vervielfältigung, Reproduktion, Übersetzung, Mikroverfilmung, Speicherung auf elektronischen Medien und Verarbeitung in elektronischer Form.

Alle Firmen-, Produkt- und Markennamen sind Warenzeichen oder eingetragene Warenzeichen der entsprechenden Inhaber.

Copyright © 2011 ePages GmbH. Alle Rechte vorbehalten.