



Orientation in Objects

Einführung in BPEL4WS

) Schulung)

AUTOR



Andreas Spall
Orientation in Objects GmbH

) Beratung)

Veröffentlicht am: 1.4.2005

ABSTRACT

) Entwicklung)

Basierend auf dem Fundament der Basistechnologien SOAP, WSDL und UDDI ist es möglich, eine serviceorientierte Architektur umzusetzen, bei welcher es nicht von Interesse ist, wie die einzelnen Dienste (Services) implementiert sind.

Zur Abbildung von Prozessen auf dieser Basis wurden eine Vielzahl von Ansätzen wie bspw. die BPML (Business Process Modeling Language) Spezifikation entwickelt. Die Spezifikation mit der größten Unterstützung von international agierenden Unternehmen stellt die BPEL4WS (Business Process Execution Language for Web Services) dar.

BPEL4WS liefert eine Sprache für die formale Spezifikation von Geschäftsprozessen und -interaktionsprotokollen. Somit steht ein interoperables Integrationsmodell zur Verfügung, das die automatisierte Prozessintegration sowohl innerhalb von Unternehmen als auch im Bereich B2B-Integration ermöglicht.

Dieser Artikel wird diese und die für sie relevanten Spezifikationen näher betrachten.

) Artikel)

Orientation in Objects GmbH

Weinheimer Str. 68
D-68309 Mannheim

Tel. +49 (0) 6 21 - 7 18 39 - 0
Fax +49 (0) 6 21 - 7 18 39 - 50

www.oio.de info@oio.de

Java, XML, UML, XSLT, Open Source, JBoss, SOAP, CVS, Spring, JSF, Eclipse

THE INITIAL TRIO OF WEB SERVICES SPECIFICATIONS

Web Services werden als sich selbstbeschreibende Software Systeme verstanden, deren Dienste (Services) Nachrichten über internetbasierte Protokolle in einem XML-Format erhalten und senden können. Sie können verwendet werden, um eine serviceorientierte Architektur (SOA) umzusetzen.

Im Folgenden wird der Fokus auf die Spezifikationen WS-Coordination, WS-Transaction und BPEL4WS gesetzt. Wie aus Abbildung 1 ersichtlich, können WS-Transaction und WS-Coordination unter dem Begriff Quality of Service zusammengefasst werden. Diese ermöglichen es, spezifische Standardprotokolle für Transaktionssysteme, Workflowsysteme oder andere Applikationen, die eine Koordination von Web Services wünschen, zu etablieren. Die Spezifikation BPEL4WS dient, wie in Abbildung 1 dargestellt, der Umsetzung von Geschäftsprozessen über Web Services. Über BPEL4WS ist eine, den Geschäftsregeln entsprechende, Komposition von Web Services möglich.

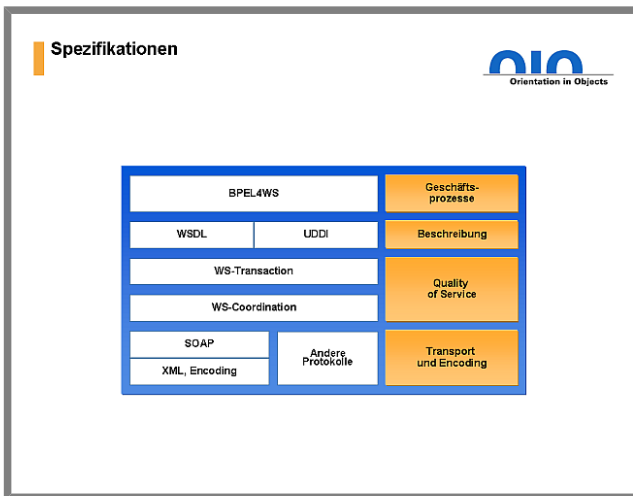


Abbildung 1: Stack der Spezifikationen

Aber betrachten wir zunächst *the initial trio of Web Services specifications* SOAP, WSDL und UDDI.

SOAP

SOAP beschreibt wie ein Dienst eines Web Services aufgerufen werden soll. Um eine plattformunabhängige Kommunikation zu ermöglichen, ist SOAP ein XML basiertes Nachrichtenaustauschformat. Dabei kann SOAP über verschiedene Protokolle wie beispielsweise HTTP oder SMTP übertragen werden.

WSDL

Die Web Service Description Language ist eine anbieter- und plattformunabhängige Beschreibung von Web Services. Über eine XML-Syntax können die Schnittstellen, bestehend aus den Methodennamen und den Parametern, definiert werden. Über diese Beschreibung ist es möglich, eine SOAP Nachricht für einen bestimmten Dienst eines Web Services zu erzeugen und zu verarbeiten.

UDDI

Der Hauptbestandteil der Universal Description Discovery and Integration Spezifikation ist eine zentrale Registrierung (UDDI-Registry), an welcher Web Services angemeldet werden können. Durch diese Registrierungsdatenbank ist es einem Web Service Client möglich, nach einem Web Service, basierend auf dessen Klassifikation und Angaben zur Firma bzw. zum Dienst, zu suchen.

WS-COORDINATION

WS-Coordination ist ein Framework zur Implementierung von spezifischen Koordinationsarten. Es ermöglicht, dass die beteiligten Web Services einen geteilten Kontext erhalten. Über diesen Kontext ist es möglich, dass Web Services Informationen eintragen und abfragen können.

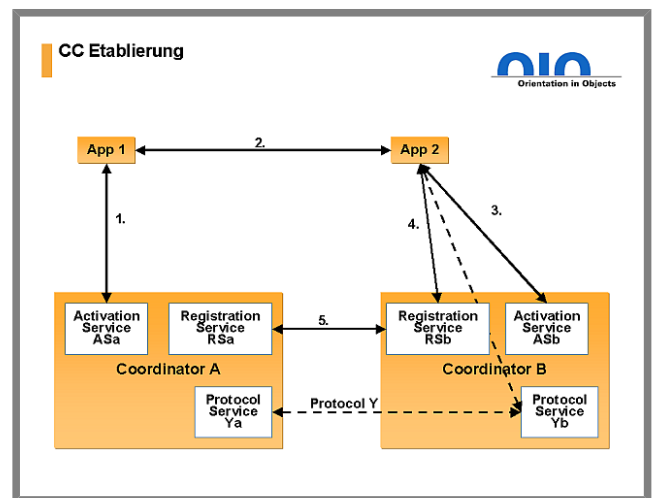


Abbildung 2: Ablauf der CC Etablierung

Abbildung 2 stellt den in der Spezifikation vorgeschriebenen Prozess zur Etablierung eines Koordinationskontextes (CC) dar.

Der Ablauf lautet wie folgt:

1. Die Applikation 1 erstellt über den Activation Service des Frameworks einen neuen Koordinationskontext (Coordinator A).
2. Die Applikation 1 sendet eine Mitteilung an die Applikation 2 im Anhang befindet sich der Coordinator A.
3. Die Applikation 2 erstellt über den Activation Service einen Koordinationkontext (Coordinator B) unter der Zuhilfenahme des Coordinator A.
4. Die Applikation 2 teil ihrem Coordinator B mit, dass sie über ein spezifisches Protokoll angesprochen werden möchte.
5. Der Coordinator B teilt dem Coordinator A der Applikation 1 mit, dass die Applikation 2 über das von ihr spezifizierte Protokoll angesprochen werden muss. Das Protokoll und der geteilte Koordinationskontext gilt als etabliert.

WS-TRANSACTION

WS-Transaction basiert auf WS-Coordination und besteht aus zwei Teilen:

- **Atomic Transaction:**
Dieser Teil wurde von der WS-AtomicTransaction Spezifikation ersetzt
- **Business Activity (WS-BusinessActivity):**
Definiert Protokolle für langlaufende Transaktionen

Die WS-Transaction Spezifikation wurde von den eigenen Kindern, der WS-Atomic Transaction und der WS-BuisnessActivity Spezifikation, überholt.

ATOMIC TRANSACTION (WS-ATOMICTRANSACTION)

WS-AtomicTransaction dient der Erzeugung eines neuen Koordinationskontextes (CC) für eine neue atomare Transaktion, oder um einen zwischengeschalteten Koordinationskontext zu einer Transaktion hinzuzufügen.

PROTOKOLLE

Hierzu wird in der WS-AtomicTransaction Spezifikation eine Vielzahl von Protokollen definiert. Diese müssen von Applikationen, die diese Spezifikation implementiert haben, unterstützt werden und lauten wie folgt:

- Completion - (cp) Ein Teilnehmer (meist die Applikation, die den CC erzeugt) registriert sich für dieses, um commit oder rollback aufzurufen
- Volatile Two phase commit - (v2pc) Ein Teilnehmer, bspw. Caches, der vor der Verwendung des Durable 2PC Protokolls informiert werden möchte, registriert sich für dieses Protokoll beim CC
- Durable Two phase commit - (d2pc) Teilnehmer wie bspw. Resource Manager registrieren sich für dieses Protokoll, damit der CC eine Commit-Abbruch-Entscheidung über alle Resource Manager durchführen kann

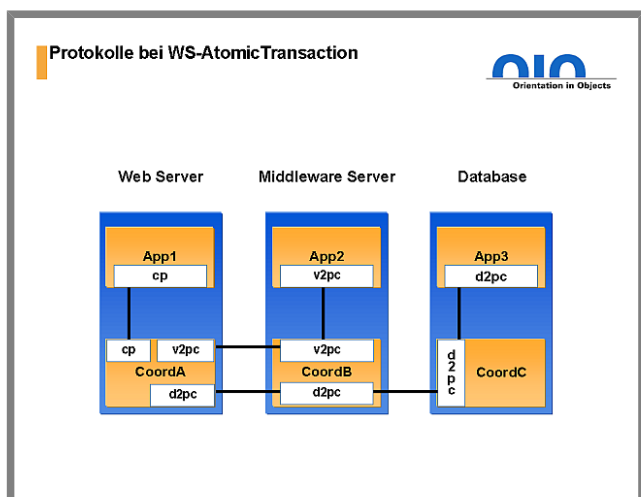


Abbildung 3: Beispiel für den Protokolleinsatz bei WS-AtomicTransaction

Diese Protokolle können verwendet werden, um bspw. eine Transaktion sicherzustellen. Wie in Abbildung 3 dargestellt:

- hat sich die Applikation 1 bei dem Koordinationskontext CoordA für das Completion Protokoll registriert,

- hat sich die Applikation 2 über ihren CoordB bei dem CoordA für das Volatile Two phase commit Protokoll angemeldet und es
- hat sich die Applikation 3 über den CoordC und CoordB bei dem CoordA für das Protokoll Durable Two phase commit registriert

Sendet die Applikation 1 an den CoordA einen Commit, informiert dieser die Applikation 2 mittels Volatile Two phase commit Protokoll über das anstehende Commit. Der Applikation 2 wird es dadurch ermöglicht ihren Cache an die Datenbank zu übertragen, bevor der Commit an die Datenbank gesendet wird. Im Anschluß daran sendet die Applikation 2 an den CoordA, über ihren CoordB, eine Mitteilung, dass die Information eingegangen und verarbeitet wurde. Nach dem Erhalt der Mitteilung, beendet der CoordA die Transaktion unter Zuhilfenahme des Durable Two phase commit Protokolls und überträgt das Commit an die Applikation 3 über den CoordB und CoordC. Im Anschluß wird der Applikation 1 mitgeteilt, dass der Commit erfolgreich durchgeführt werden konnte.

BUSINESS ACTIVITY (WS-BUSINESSACTIVITY)

In einer Web Service Umgebung, kann eine Business Activity mehrere Tage dauern. Das Sperren von Ressourcen ist in diesem Fall nicht empfehlenswert. Aus diesem Grund werden Aktionen sofort durchgeführt und Änderungen an Daten sind permanent. Im Falle eines Fehlers werden Aktionen verwendet, um die schon durchgeführten Modifikationen zu kompensieren.

Eine Business Activity kann in mehrere untergeordnete Scopes aufgeteilt werden. Ein Scope ist ein Geschäftsauftrag einer beliebigen Berechnung, ausgeführt als beschränkte Menge von Operationen einer Sammlung von Web Services, die in einer bestimmten Reihenfolge ablaufen müssen. Hierzu wurden folgende Protokolle definiert.

- BusinessAgreement - Ein Teilnehmer eines untergeordneten Scopes registriert sich für dieses Protokoll mit Hilfe des vom Vorgänger übergebenen Koordinationskontextes, so dass dieser ihn steuern kann. Er bekommt unter anderem mitgeteilt, wann er seine Arbeit für eine Business Activity abgeschlossen hat und fragt, ob er die durchgeführten Tätigkeiten kompensieren muss oder sich beenden kann. Beim Fehlschlagen einer Aktivität hat er die Aufgabe, den Vorgänger zu informieren.

Dieses Protokoll findet bei der BPEL4WS Spezifikation bei der Fehler und Kompensationsbehandlung Anwendung.

- BusinessAgreementWithComplete - Das BusinessAgreementWithComplete Protokoll ist gleichbedeutend zu dem BusinessAgreement Protokoll, mit der Ausnahme, dass ein untergeordneter Scope eine Nachricht erhält, wenn er alle Requests, die zu einer bestimmten Business Activity gehören, erhalten hat und abarbeiten kann.

BPEL4WS

Die Business Process Execution Language for Web Services (BPEL4WS) ermöglicht es, eine Menge von Web Services zu einem neuen Web Service zusammenzusetzen.

Ziel der BPEL4WS ist es, Prozesse zu beschreiben und diese falls benötigt auch zu instanzieren.

KONZEPTE

Zur Realisierung des Ziels werden innerhalb der BPEL4WS die unterstützenden Konzepte Variable, Fault handling, Compensation handler, Partner Link Types und Links verwendet.

- Daten für Prozessinstanzen können in der Ablauflogik referenziert werden. z.B. Nachrichten und Statusinformationen der an dem Prozeß beteiligten Web Services speichern.
- Fehler können mittels eigenen Mechanismen (fault handling) abgefangen werden.
- Mittels compensation handler können jene Aktivitäten definiert werden, die durchgeführt werden sollen, wenn eine zuvor durchgeführte Aktivität nicht mehr rückgängig gemacht werden kann, aber kompensiert werden muss. Über Scopes kann der Aktionsradius dieser Kompensation definiert werden.
- Mittels Partner Link Types wird die Beziehung zwischen zwei Diensten beschrieben. Dafür wird das Konzept von Roles (Rollen) verwendet, wobei sich eine Rolle auf einen WSDL portType bezieht. Basierend auf diesen Partner Link Types können Partner Links erzeugt und innerhalb von Aktivitäten verwendet werden.
- Die Synchronisation von einzelnen Aktivitäten kann durch das Verwenden von sogenannten Links realisiert werden. Dazu wird zunächst ein Link (<link name="einName">) definiert. Dieser Link kann nun innerhalb der Aktivität als Quelle (<source linkName="einName">) oder als Ziel (<target linkName="einName">) verwendet werden.

AKTIVITÄTEN

Aktivitäten dienen der deklarativen Zusammenstellung eines Geschäftsprozesses über eine XML konforme Datei mit der Endung bpel. Innerhalb des Tags <process> können hierzu eine Vielzahl von Aktivitäten (vgl. folgende Tabelle) verwendet werden.

Aktivität	Beschreibung
<receive> <reply>	Warten auf Nachricht (<receive>) und Antwort auf eine Nachricht (<reply>). Die Kombination beider Aktivitäten ermöglicht eine request - response Operation für einen WSDL portType des Prozesses
<invoke>	Aufruf eines Web Services
<assign>	Updaten von Variablen mit neuen Daten
<throw> <catch>	interne Fehler eines Geschäftsprozesses signalisieren (throw) bzw. abfangen (catch)
<terminate>	explizites beenden des Prozesses
<wait>	der Prozess wartet eine definierte Zeit
<empty>	leere Aktivität, wird verwendet um Fehler abzufangen oder um Aktivitäten eines Geschäftsprozesses zu synchronisieren
<sequence>	sequentielle Ausführung der nachgereihten Aktivitäten
<swich>	Aktivität zur Modellierung von Verzweigungen. Kann eine oder mehrere Bedingungen (<case condition="boolescher Ausdruck">) und optional einen alternativen Zweig (<otherwise>) enthalten
<while>	Mittels <while> wird die betroffene Aktivität so lange wiederholt ausgeführt, bis der boolesche Ausdruck nicht mehr erfüllt wird
<pick>	Diese Aktivität wartet auf das Auftreten eines Ereignisses aus einer Menge von Ereignissen und führt danach die entsprechend angeführte Aktivität aus
<flow>	Das <flow> Konstrukt ermöglicht die parallele Verarbeitung von einer oder mehreren Aktivitäten. Mittels dem <link> Konstrukt können direkt oder indirekt eingebettete Aktivitäten synchronisiert werden
<scope>	Das <scope> Konstrukt ermöglicht die Verbindung einer eingebetteten Aktion mit Variablen, einer Fehlerbehandlung und Kompensationsbehandlung. Zur Koordination wird hierbei das BusinessAgreement Protokoll der WS-Transaction Spezifikation verwendet
<compensate>	<compensate> wird verwendet, wenn eine in einen <scope> eingebettete Aktion erfolgreich durchgeführt wurde, diese Aktion aber kompensiert werden muss. Dieses Konstrukt kann nur von einem anderen compensation handler oder von dem fault handling aufgerufen werden

Tabelle 1: Aktivitäten von <process>

UMSETZUNG EINES GESCHÄFTSPROZESSES

Zur Verdeutlichung der Umsetzung eines Geschäftsprozesses, unter der Verwendung der BPEL4WS, wurde hier das Beispiel der Business Process Execution Language for Web Services Version 1.1 Spezifikation gewählt.

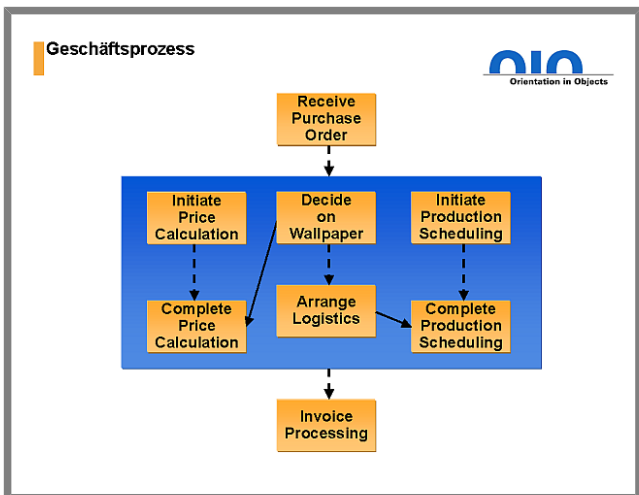


Abbildung 4: Geschäftsprozess

Wie in Abbildung 4 dargestellt, beginnt der Prozess mit dem Eingang einer Bestellung (Receive Purchase Order). Nach dem Erhalt der Bestellung starten drei Teilprozesse. Der erste Prozess dient der Kalkulation des Preises, der zweite Prozess der Lieferungsplanung der Produkte und der dritte Prozess der Produktionsplanung der in der Bestellung angegebenen Güter. Kommen die drei Prozesse zu einem erfolgreichen Ergebnis, gilt die Fakturierung als erfolgreich bzw. der Gesamtprozess wird beendet und der Kunde wird über diesen Erfolg informiert.

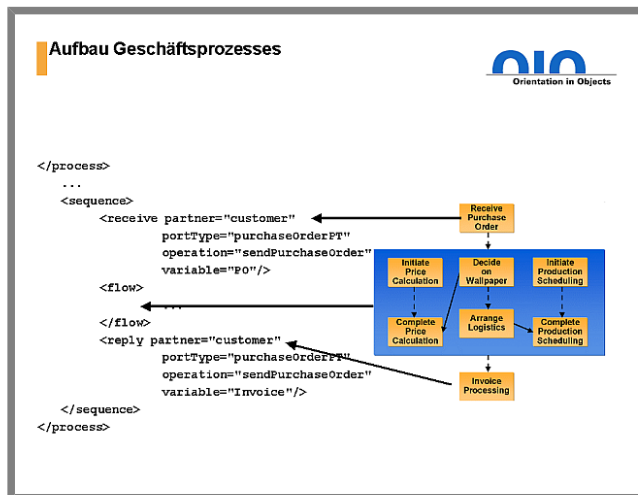


Abbildung 5: Aufbau des Geschäftsprozesses

Hierzu wird innerhalb des Tags <prozess> die Aktivität <sequence> eingefügt. Mit dem Resultat, dass nach dem Eingang einer Bestellung eines Kunden (<receive.../>) die Aktivität <flow> angesprochen wird und der Kunde über die Aktivität <reply> über dessen Ausgang informiert wird. Innerhalb der Aktivität <flow> können nun die zur Kalkulation, Lieferungsplanung und Produktionsplanung benötigten Web Services angesprochen werden.

VERWENDEN DER KONZEPTE

Bevor der Prozess über die BPEL-Datei aufgebaut werden kann, müssen die Konzepte definiert werden, die in diesem Prozess Verwendung finden sollen.

```
<process name="purchaseOrderProcess">
  <variables>
    <variable name="PO" messageType="POMessage" />
    ...
  </variables>
  <faultHandlers>
    <catch faultName="cannotCompleteOrder"
      faultVariable="POFault">
      <reply partner="customer" portType="purchaseOrderPT"
        operation="sendPurchaseOrder"
        variable="POFault"
        faultName="cannotCompleteOrder" />
    </catch>
  </faultHandlers>
</process>
```

Beispiel 1: BPEL-Datei des Geschäftsprozesses

Wie in Listing 1 dargestellt, sollten für den Geschäftsprozess eine Variable und ein fault handler verwendet werden. Durch die Variable ist es den Teilprozessen möglich, untereinander Daten auszutauschen bzw. Nachrichten zu senden. Über den fault handler ist es möglich, dass bei einem Auftreten eines Fehlers innerhalb eines Teilprozesses der Kunde über diesen Fehler informiert wird.

AUFBAU DES GESCHÄFTSPROZESSES

Der Auszug - siehe Abbildung 5 aus der BPEL-Datei - beschreibt den in Abbildung 4 verdeutlichten Prozess zur Bearbeitung einer Bestellung.

FAZIT

Dieser Artikel hat aufgezeigt, dass es über BPEL4WS möglich ist, Geschäftsprozesse unter Zuhilfenahme von Web Services zu erstellen. Hierbei stellt der resultierende Geschäftsprozess wiederum ein Web Service dar, welches als solches in einem weiteren Geschäftsprozess Verwendung finden kann. Somit kann dieses Konzept als geschlossen gelten.

Hinzu kommt, dass die Unterstützung dieser Spezifikation weit vorangeschritten ist. Bspw. unterstützen die Softwarehersteller Oracle und IBM diese Spezifikation mit eigenen Produkten. Bei Oracle ist dies der Oracle BPEL Process Manager. Bei IBM der IBM WebSphere Business Integration Server Foundation der Version 5.1.

In den folgenden Jahren wird BPEL4WS unter dem Namen WS-BPEL wohl an Bekanntheit und Unterstützung anderer Anbieter zunehmen, da die Organization for the Advancement of Structured Information Standards (OASIS) diese Spezifikation voraussichtlich als Standard bestätigen wird.

REFERENZEN

- [1] W3C Note - SOAP 1.2
, World Wide Web Consortium
[\(http://www.w3.org/TR/soap12-part0/\)](http://www.w3.org/TR/soap12-part0/)
- [2] Web Service Description Language (WSDL) 1.1
, World Wide Web Consortium
<http://www.w3.org/TR/wsdl>
- [3] Web Services Coordination (WS-Coordination)
Cabrera, Luis Felipe ; Copeland, George ; Cox, William ; al., et
2002-08-09
[\(http://www.ibm.com/developerworks/library/specification/ws-tx/\)](http://www.ibm.com/developerworks/library/specification/ws-tx/)
- [4] Specification: Web Services Transaction (WS-Transaction)
Cabrera, Felipe ; Copeland, George ; Cox, Bill
2002-08-09
[\(http://www.ibm.com/developerworks/library/specification/ws-tx/\)](http://www.ibm.com/developerworks/library/specification/ws-tx/)
- [5] Business Process Execution Language for Web Services Version 1.1
Andrews, Tony ; Curbera, Francisco ; Dholakia, Hitesh ; al., et
2003-05-05
[\(http://www.ibm.com/developerworks/library/specification/ws-bpel/\)](http://www.ibm.com/developerworks/library/specification/ws-bpel/)
- [6] Oracle BPEL Process Manager
<http://www.oracle.com/technology/products/ias/bpel/index.html>
- [7] WebSphere Business Integration Server Foundation
[\(http://www.ibm.com/software/integration/wbisf/features/\)](http://www.ibm.com/software/integration/wbisf/features/)
- [8] Organization for the Advancement of Structured Information Standards (OASIS)
<http://www.oasis-open.org/home/index.php>