



Benutzerhandbuch – Webservice „Befragung und Bestellung Auszug EO-Register“

Klassifizierung*	Intern
Status**	zur Nutzung
Name des Projektes	Register für Erwerbsausfallentschädigungen
Projektkürzel	EO-Register
Projektnummer	
Projektleiter	François-Xavier Conrad
Auftraggeber	Daniel Bonvin
Autor	Olivier Schmitt
Kürzel des Autors	SHM
Co-Autor	Stéphane Krattinger
Prüfer	Laurent Dumont, Christophe Canel, Sandra Monnier
Genehmiger	Daniel Bonvin, Belgacem Ben Mabrouk
Zur Kenntnisnahme	

* Nicht klassifiziert, Intern, Vertraulich

** in Arbeit, In Prüfung, Genehmigt, zur Nutzung

Änderungskontrolle, Prüfung, Genehmigung

Version	Datum	Beschreibung / Bemerkung	Name / Rolle
0.1	10.08.2011	Erstversion	SHM
0.2	16.09.2013	Durchsicht	KRA
1.0	23.10.2013	Abschluss	SHM
1.1	25.10.2013	Korrektur von Rechtschreibfehlern	CAI
1.2	27.01.2014	XSD 4.2	SAP
1.3	12.02.2014	Verschiedene keystore Abna und Prod	SAP
1.4	29.04.2014	Durchsicht des Dokuments (Registerauszug Korrekturen)	SAP/COR

Verwendete Begriffe und Abkürzungen

Begriff / Abkürzung	Bedeutung

Referenzierte Dokumente

«HERMES - Führen und Abwickeln von Projekten der Informations- und Kommunikationstechnik (IKT)» ist ein offener Standard der schweizerischen Bundesverwaltung. Die Schweizerische Eidgenossenschaft, vertreten durch das Informatikstrategieorgan Bund (ISB), ist Inhaberin der Urheberrechte an HERMES.



Ref.	Titel / Beschreibung
------	----------------------

Inhaltsverzeichnis

1	Vorwort	3
2	Zweck des Dokumentes	3
3	Bestandteile (des Webservice).....	3
3.1	Rapg-webservice-1-0.wsdl	3
3.2	Schema rapg-ws-1-0.xsd	4
3.2.1	Aktion findAnnonces.....	4
3.2.1.1	Element rapgAnnoncesRequest	4
3.2.1.2	Element rapgAnnoncesResponse.....	4
3.2.2	Aktion getRegisterStatus.....	5
3.2.2.1	Element: registerStatusRequest	5
3.2.2.2	Element: registerStatusResponse.....	6
3.3	Zertifikat in SOAP	7
3.4	Zugriffe auf den Webservice	7
3.5	Fehlermeldungen	7
3.5.1	Aktion findAnnonce	7
3.5.2	Aktion getRegisterStatus.....	7
3.6	Fehler-Codes des sedex Authentifizierungs-Service:	8
4	Test Client.....	9
4.1	Verwendete Frameworks.....	9
4.2	Voraussetzungen.....	9
4.2.1	Installation des m2e connectors buildhelper.....	9
4.3	Installation	10
4.3.1	Herunterladen der Datei	10
4.3.2	Erstellen eines leeren Maven-Projektes.....	10
4.3.3	Projekt in Eclipse importieren.....	12
4.3.4	Erzeugen des Source Codes: Klasse generieren und kompilieren	13
4.4	Struktur des Projektes	14
4.5	Beschreibung der Klasse CommandLauchner.....	15
4.5.1	main.....	15
4.6	Beschreibung der Klasse RAPGConsultationSSLSession	15
4.6.1	findAnnonces.....	15
4.6.2	configureSSLonTheClient	16
4.6.3	getRAPGConsultationService	17
4.7	Ausführen der Klasse CommandLauchner	17
5	Beispiele	20
5.1	Anfragemeldung	20
5.2	Antwortmeldung	20

Abbildungsverzeichnis

Abbildung 1: Verpflichtung Rapg-webservice-1-0.wsdl 1	3
Abbildung 2: Element rapgAnnoncesRequest 1.....	4
Abbildung 3: Element rapgAnnoncesResponse.....	5
Abbildung 4: Element registerStatusRequest.....	5
Abbildung 5: Element registerStatusResponse.....	7
Abbildung 6: Installation des m2e Konnektors	10
Abbildung 7: Création d'un projet maven vide – Schritt 1	11
Abbildung 8: Création d'un projet maven vide – Schritt 2	11
Abbildung 9: Création d'un projet maven vide – Schritt 3	12
Abbildung 10: Importation du projet – Schritt 1	12
Abbildung 11: Importation du projet – Schritt 2	13
Abbildung 12: Erzeugen des Source Codes: Klasse generieren und kompilieren – Schritt 1	13
Abbildung 13: Aktualisierung der Konfiguration des Maven-Projektes – Schritt 2	14
Abbildung 14 : Ausführen der Klasse CommandLauncher – Schritt 1	17
Abbildung 15 : Ausführen der Klasse CommandLauncher – Schritt 2	18
Abbildung 16 : Ausführen der Klasse CommandLauncher – Schritt 3	18

Abbildung 17 : Ausführen der Klasse CommandLauncher – Schritt 4 19
 Abbildung 18 : Antwort des CommandLaunchers 20
 Abbildung 19: Exemple de message d’appel 20

1 Vorwort

Das Schema der in den Webservices verwendeten Meldungen ist in der Version 4.2.

2 Zweck des Dokumentes

Der Webservice zur Befragung des Registers für Erwerbsausfallentschädigungen erlaubt es einer Ausgleichskasse, einen Auszug der EO-Meldungen einer Person (Bezieher) anhand einer AHV- und Ausgleichskassen (deliveryOffice)-Nummer zu erhalten.

Er ermöglicht ebenfalls den Auftrag an einen Batch zur Erstellung eines Auszugs des EO-Registers zu erteilen.

3 Bestandteile (des Webservice)

Alle unten abgebildeten Schemata stehen unter der folgenden URL zur Verfügung:
<http://www.zas.admin.ch/org/00721/00851/00959/00972/index.html?lang=de>.

Bestandteil	Typ	Beschreibung
rapg-webservice-1-0.wsdl	wsdl	Verpflichtungen des Services v1.0
rapg-ws-1-0.xsd	xsd	Schema, welches die Datentypen für Ein- und Ausgang des Service festlegt.
eCH-xsd	rép.	Verzeichnis, welches die eCH-xxxx Schemata, welche in den XSD Schemata des Webservices verwendet werden, beinhaltet.
eCH-0044-2-0f.xsd	xsd	Schema zur Personenidentifikation.
eCH-0058-4-0.xsd	xsd	Schnittstellenstandard Meldungsrahmen.
eahv-iv-2015-common-4-2.xsd	xsd	Gemeinsames Schema für alle EO-Meldungen.
eahv-iv-2015-000601-4-2.xsd	xsd	Schema, welches Meldungen des Typs 601 des EO-Registers beschreibt.
eahv-iv-2015-000701-4-2.xsd	xsd	Schema, welches die Austauschformate zur Bestellung eines EO-Register Auszugs beschreibt.

3.1 Rapg-webservice-1-0.wsdl

Es handelt sich um die Verpflichtung des Webservices. Diese besteht aus zwei Funktionen:

1. *findAnnonces*, welche als Eingang *findAnnoncesRequestMessage* erwartet und *findAnnoncesResponseMessage* zurückliefert. Die entsprechenden WSDL Definitionen sind im Schema rapg-ws-1-0.xsd festgehalten.
2. *getRegisterStatus* welche als Eingang *registerStatusRequestMessage* erwartet und *registerStatusResponseMessage* zurückliefert. Die entsprechenden WSDL Definitionen sind im Schema rapg-ws-1-0.xsd festgehalten

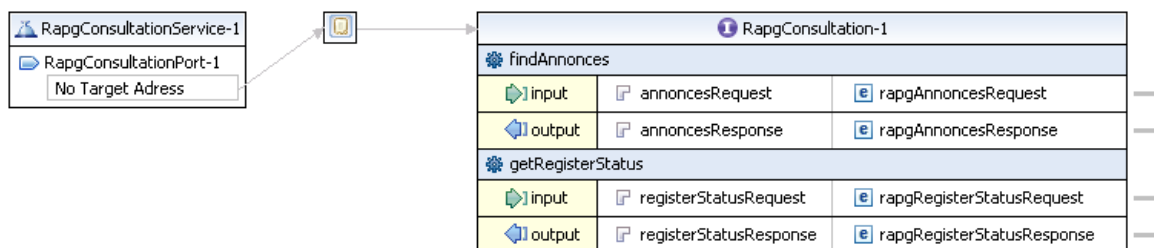


Abbildung 1: Verpflichtung Rapg-webservice-1-0.wsdl 1

3.2 Schema rapg-ws-1-0.xsd

Es handelt sich um das Schema, in welchem die Aktionen definiert werden.

- findAnnonces, welche aus *rapgAnnoncesRequest* und *rapgAnnoncesResponse* besteht
- getRegisterStatus, welche aus *rapgregisterStatusRequest* und *rapgregisterStatusResponse* besteht

3.2.1 Aktion findAnnonces

Diese Aktion ermöglicht es, die Meldungen eines EO-Beziehers im gesamten Register zu erfragen.

3.2.1.1 Element rapgAnnoncesRequest

Dieses Element ist vom Typ *rapgAnnoncesRequestType*. Es handelt sich um den Bestandteil, welcher definiert werden muss, um den Webservice aufzurufen. Er besteht aus zwei Pflichtelementen:

Element	Typ	V	Beschreibung
vn	vnType	1	AHV Nummer der Person, d.h. eine 13-stellige Nummer mit einem Wertebereich zwischen 7560000000000 und 7569999999999 (siehe in eCH-0044)
deliveryOffice	deliveryOfficeType	1	Kennung der Kasse, mit welcher der Aufruf an den Webservice gemacht wird. Das Element (bzw. der Typ <i>deliveryOffice</i>) besteht aus: <ul style="list-style-type: none"> • officeIdentidier vom Typ Integerzahl im Wertebereich von 1 bis 999. Hiermit ist die Nummer der Ausgleichskasse gemeint. • branch vom Typ Integerzahl im Wertebereich von 0 bis 999. Hiermit ist die Kennung einer Zweigstelle der Ausgleichskasse gemeint.

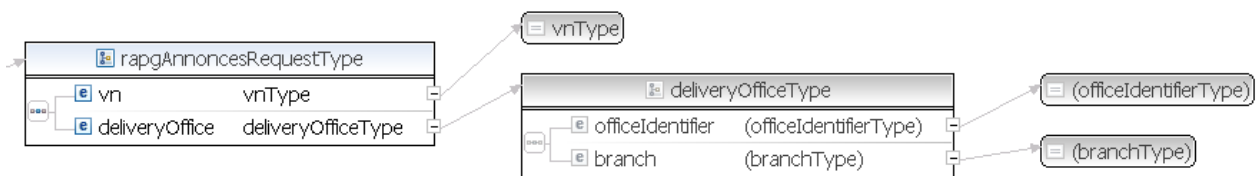


Abbildung 2: Element rapgAnnoncesRequest 1

3.2.1.2 Element rapgAnnoncesResponse

Es handelt sich um die Antwort der Anfrage, welche vom Typ *rapgAnnoncesResponseType* ist und wie folgt aufgebaut ist:

Element	Typ	V	Beschreibung
ack	ackType	1	Es handelt sich um die Empfangsbestätigung, welche folgenden Inhalt haben kann: <ul style="list-style-type: none"> • failure: Fehler, der die Bearbeitung der Anfrage verhindert. • partialfailure: Fehler, der lediglich der Bearbeitung eines Teils der Anfrage erlaubt. • success: Die Anfrage konnte erfolgreich verarbeitet werden. • warning: Die Anfrage konnte bearbeitet werden, aber es gibt Warnmeldungen, die bei der Verarbeitung aufgetreten sind.
errors	erreurMessageType	0..n	Fehlermeldungen, die bei der Verarbeitung aufgetreten sind. Diese bestehen aus einer den Fehler beschreibenden Textmeldung, einem Fehlercode (fakultativ), einer Fehlerklassifizierung (business, technical, other) und der Fehlerstufe (WARN, ERROR, FATAL).
message		0..1	Die antwortmeldung, welche aus header und content aufgebaut ist

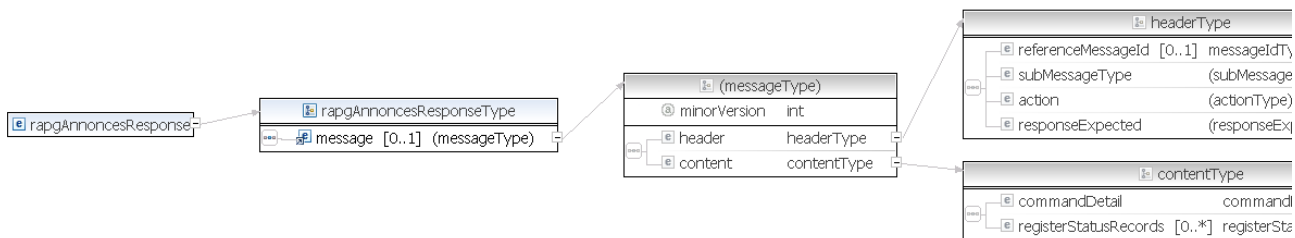


Abbildung 3: Element rapgAnnouncesResponse

Das Element registerStatusRecords ist vom Typ registerStatusRecordType, welches auf das Schema eahv-iv-2015-000601-4-2.xsd referenziert.

3.2.2 Aktion getRegisterStatus

Mit dieser Aktion können die Anfragen zu einem Registerauszug verwaltet werden.

3.2.2.1 Element: registerStatusRequest

Das Element ist vom Typ rapgRegisterStatusRequestType und es handelt sich dabei um die Komponente, mit welcher der Service aufgerufen werden muss. Diese besteht aus dem Pflichtelement *message*, welches im Schema eahv-iv-2015-000701 definiert ist. Das Element enthält die folgenden Attribute und Elemente:

Element	Typ	V	Beschreibung
minorVersion	int	1	Unterversion des verwendeten Webservices
messageId	messageIdType	1	Numerische Kennung der Anfrage.
deliveryOffice/ officeIdentifier	int (1 bis 999)	1	Numerische Kennung der Ausgleichskasse.
deliveryOffice/ branch	int (0 bis 999)	1	Numerische Kennung der Zweigstelle.
commandDetail	commandDetailsType	1	Einzelheiten der Anfrage: Laufende Anfragen (ja, nein, alle) Kollidierende Anfragen (ja, nein, alle) Stornierte Anfragen (ja, nein, alle) Veraltete Anfragen (ja, nein, alle)
requestProcessDate	Date	1	Gewünschtes Datum der Verarbeitung.
action	actionType	1	Aktion: 1 = Neue Anfrage 3 = Stornierung einer Anfrage 4 = Korrektur einer Anfrage. 5 = Abfrage der Anfrage
referenceMessageId	messageIdType	0..1	Nachrichtenennung, die bei den Aktionen 3, 4 und 5 belegt sein muss und in dem Fall den Wert der Nachrichtenennung der Anfrage, welche mit der Aktion 1 gesendet wurde, beinhalten muss.

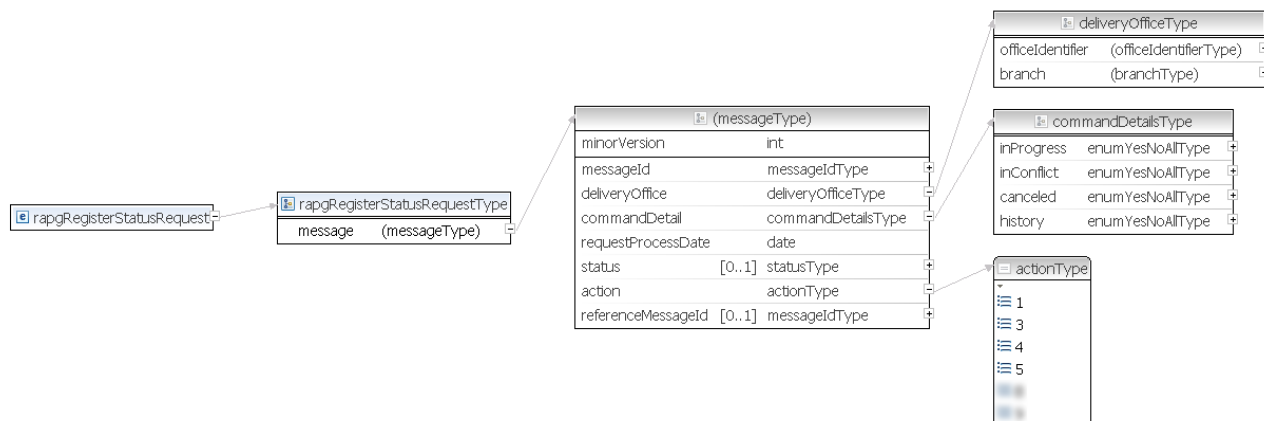


Abbildung 4: Element registerStatusRequest

3.2.2.2 Element: registerStatusResponse

Es handelt sich um die Antwort der Anfrage, welche vom Typ rapgRegisterStatusResponseType ist und wie folgt aufgebaut ist:

Element	Typ	V	Beschreibung
ack	ackType	1	Es handelt sich um die Empfangsbestätigung, welche folgenden Inhalt haben kann: <ul style="list-style-type: none"> • failure: Fehler, der die Bearbeitung der Anfrage verhindert. • partialfailure: Fehler, der lediglich der Bearbeitung eines Teils der Anfrage erlaubt. • success: Die Anfrage konnte erfolgreich verarbeitet werden. warning: Die Anfrage konnte bearbeitet werden, aber es gibt Warnmeldungen, die bei der Verarbeitung aufgetreten sind.
errors	erreurMessageType	0..n	Fehlermeldungen, die bei der Verarbeitung aufgetreten sind. 3.2.2.3 Diese bestehen aus einer den Fehler beschreibenden Textmeldung, einem Fehlercode (fakultativ), einer Fehlerklassifizierung (business, technical, other) und der Fehlerstufe (WARN, ERROR, FATAL).
message		0..1	3.2.2.4 Die Antwortmeldung, welche aus header und content aufgebaut ist

Das Element Meldung enthält die folgenden Attribute und Elemente

Elément	Type	V	Description
minorVersion	int	1	Unterversion des verwendeten Webservices
messageld	messageldType	1	Numerische Kennung der Anfrage.
deliveryOffice/ officidentifier	int (1 bis 999)	1	Numerische Kennung der Ausgleichskasse.
deliveryOffice/ branch	int (0 bis 999)	1	Numerische Kennung der Zweigstelle.
commandDetail	commandDetailsType	1	Einzelheiten der Anfrage: Laufende Anfragen (ja, nein, alle) Kollidierende Anfragen (ja, nein, alle) Stornierte Anfragen (ja, nein, alle) Veraltete Anfragen (ja, nein, alle)
requestProcessDate	Date	1	Gewünschtes Datum der Verarbeitung.
status	statusType	0..1	Status der Anfrage : Scheduled inProgress ended error canceled
action	actionType	1	Aktion: 8 = negative Antwort - Fehlerreport zu einer Ereignislieferung. 9 = positive Antwort - Rückmeldung des korrekten Empfangs einer Meldung.
referenceMessageld	messageldType	0..1	Nachrichtenkennung, die bei den Aktionen 3, 4 und 5 belegt sein muss und in dem Fall den Wert der Nachrichtenkennung der Anfrage, welche mit der Aktion 1 gesendet wurde, beinhalten muss.

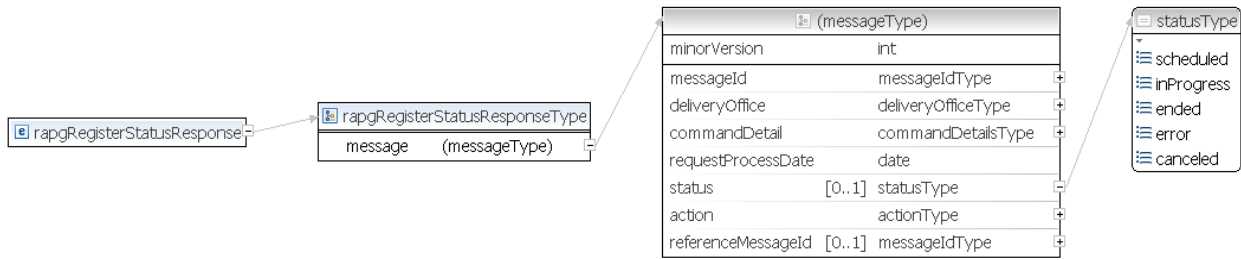


Abbildung 5: Element registerStatusResponse

3.3 Zertifikat in SOAP

Die SOAP Anfrage muss einen Eintrag `ssl_Client_cert` enthalten, in welcher das vom BSV herausgegebene p12 Zertifikat angegeben wird (das Zertifikat muss direkt in der Anfrage und nicht in deren Header enthalten sein).

3.4 Zugriffe auf den Webservice

Der Webservice ist in zwei verschiedenen Umgebungen zugänglich:

1. Abna: Abnahmeumgebung, welche externen Akteuren (z.B. den Ausgleichskassen) zum Testen dient.
2. Prod: Produktionsumgebung

Umgebung	Zone	Wer	URL-Aufruf von Client-Seite aus	Zugriffskontrolle
Abna	SSZ	Alle	https://www.rapg-a.zas.admin.ch/ApgCore/ws/consultation	RACF https
Prod	SSZ	Alle	https://www.rapg.zas.admin.ch/ApgCore/ws/consultation	RACF https

3.5 Fehlermeldungen

Anbei eine Liste der Fehlermeldungen, die vom Webservice zurückgegeben werden können.

3.5.1 Aktion findAnnounce

Typ	Stufe	Nachricht	Ursache
technical	FATAL	Fehlermeldungen der Infrastruktur Sedex, siehe unter 3.6.	Probleme bei der Authentifizierung mittels Sedex EAWS-Service
business	ERROR	The parameter 'deliveryOffice' is blank.	Der Wert für das Element deliveryOffice ist leer muss aber gefüllt sein.
business	ERROR	The sended vn (<i>yournavs</i>) is invalid	Die zugestellte AHV Nummer ist nicht 13-stellig oder die Kontrollnummer stimmt nicht.
business	ERROR	The delivery office <i>yourdeliveryoffice</i> is unknown in register	Die zugestellte Nummer der Ausgleichskasse ist nicht im Register vorhanden.
business	WARN	No announcement in register for this vn <i>yournavs</i>	Im Register ist keine Meldung zu der mitgelieferten AHV-Nummer vorhanden.
technical	ERROR	System error, please contact the technical support with this message: <i>the message</i>	Das System steht nicht zur Verfügung.

3.5.2 Aktion getRegisterStatus

Typ	Stufe	Nachricht	Ursache
technical	FATAL	Fehlermeldungen der Infrastruktur Sedex, siehe unter 3.6.	Probleme bei der Authentifizierung mittels Sedex EAWS-Service
business	ERROR	The delivery office <i>yourdeliveryoffice</i> is unknown in register.	Die zugestellte Nummer der Ausgleichskasse ist nicht im Register

			vorhanden.
business	ERROR	Criteria 'inConflict=Yes' and 'canceled=Yes' cannot be combined. There would be no result	Eine Anfrage die nur aus annullierten und in Konflikt Meldungen besteht ist unmöglich.Keine Resultate werden produziert
Business	ERROR	Criteria 'InProgress=Yes' and 'canceled=Yes' cannot be combined. There would be no result	Eine Anfrage die nur aus annullierten und laufenden Meldungen besteht ist unmöglich.Keine Resultate werden produziert
Business	ERROR	The parameter 'referenceMessageld' must be null for an action = 1	Das Element referenceMessageld muss nicht bei Aktion 1 belegt sein
Business	ERROR	You can't schedule an execution in the past. Received date : <i>yourdate</i>	Ein Auszug kann nicht für einen Tag in der Vergangenheit bestellt werden
Business	ERROR	You can't schedule an execution so far in the future. Maximum number of days in the future : 60	Ein Auszug kann nicht 60 Tagen im voraus bestellt werden
business	ERROR	An Execution for referenceMessageld: <i>yourmessageld</i> was already scheduled. Update instead.	Es existiert bereits ein Auszug aus dem Register zu der angegebenen messageld.
business	WARN	Already exists an execution for the office <i>yourdeliveryoffice</i> for the date <i>yourdate</i> . Use update instead.	Es ist bereits ein Auszug des Registers für diese Kasse zu dem gleichen Datum geplant.
business	ERROR	Impossible to schedule an execution for the register status batch. It may have too many executions for the date: <i>yourdate</i> , try it again with an other date.	Es ist nicht möglich, ein Auszug aus dem Register zum gewünschten Datum zu erstellen, weil es bereits zu viele Anfragen gibt. Der Anzahl der Anfragen ist begrenzt
Business	ERROR	The parameter 'referenceMessageld' can't be null	Das Element referenceMessageld muss bei den Aktionen 3, 4 und 5 belegt sein
business	ERROR	No execution has been found for referenceMessageld: <i>yourreferencemessageid</i> .	Es existiert keine Auszug zu den angegebenen Referenz-Kennung.
Business	ERROR	Execution with 'referenceMessageld' = <i>yourreferencemessageid</i> can not be modified, it's already in a terminal state : <i>thestate</i>	Ein Auszug der bereits behandelt ist kann nicht mehr geändert werden
business	ERROR	Unknown or unimplemented action: <i>youraction</i>	Es wurde ein undefinierter Aktions-Code geschickt.
technical	ERROR	System error, please contact the technical support with this message: <i>the message</i>	Das System steht nicht zur Verfügung.

3.6 Fehler-Codes des sedex Authentifizierungs-Service:

errorCode	Name	Beschreibung
0	OK	Operation erfolgreich abgeschlossen.
<i>Message Type Errors:</i>		
10	messageTypeld invalid	Die Id bezieht sich nicht auf einen existierenden MessageType.
<i>Sender Errors:</i>		
20	missing sender identification	Es wurde weder eine sedexId noch ein Zertifikat zur Identifikation eines Teilnehmers übermittelt.
21	sender sedexId invalid	Die Id bezieht sich nicht auf einen existierenden Teilnehmer.
22	sender certificate invalid	Das Zertifikat kennzeichnet keinen existierenden Teilnehmer.
23	sender certificate format error	Das Zertifikat entspricht nicht dem erwarteten Format.

<i>Recipient Errors:</i>		
30	missing recipient identification	Es wurde weder eine sedexId noch ein Zertifikat zur Identifikation eines Teilnehmers übermittelt.
31	recipient sedexId invalid	Die Id bezieht sich nicht auf einen existierenden Participant.
32	recipient certificate invalid	Das Zertifikat kennzeichnet keinen existierenden Teilnehmer.
33	recipient certificate format error	Das Zertifikat entspricht nicht dem erwarteten Format.
<i>Other Errors:</i>		
99	internal sedex error	Ein Verarbeitungsfehler innerhalb des Sedex-Systems ist aufgetreten.

4 Test Client

Die Informatikabteilung der ZAS stellt ein Java Hilfs-Projekt zur Verfügung, welches eine beispielhafte Client-Anwendung demonstriert. Das Projekt ist mit dem Build-Management-Tool Maven eingerichtet und besteht aus einer Klasse mit der main() Methode.

4.1 Verwendete Frameworks

Die verwendeten Frameworks sind:

1. Apache CXF: Framework, dass mit Hilfe von Web Service Standards das Erstellen einer Webservice Anwendung unterstützt.
2. JAXB: Java Architektur zur XML-Datenbindung.
3. Maven: Build-Management-Tool

4.2 Voraussetzungen

Um das Hilfs-Projekt verwenden zu können, müssen folgende Komponenten installiert sein:

- Java 1.6
- Eclipse 3.5, 3.6 oder 3.7
- m2eclipse (update site <http://download.eclipse.org/technology/m2e/releases/>)
- m2e connectors builderhelper

4.2.1 Installation des m2e connectors builderhelper

Diese Komponente verbindet Maven mit den neusten Versionen von m2eclipse. Im Besonderen mit dieser Version, die hier abgerufen werden kann <http://download.eclipse.org/technology/m2e/releases/>.

Wenn die Installation von m2eclipse abgeschlossen ist, bitte Window > Preferences > Maven > Discovery > open catalog auswählen. Von den zur Verfügung stehenden Konnektoren muss « builderhelper » installiert werden.

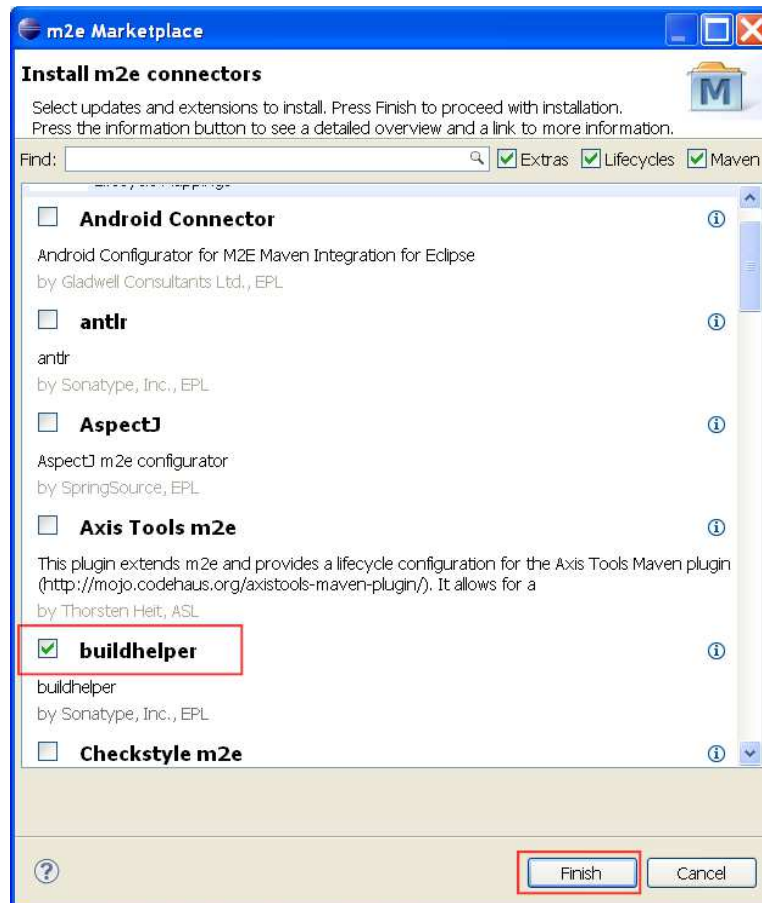


Abbildung 6: Installation des m2e Konnektors

4.3 Installation

Die Installation findet in vier Teilschritten statt:

1. Herunterladen der komprimierten Datei, welche das Projekt beinhaltet.
2. Erstellen eines leeren Maven-Projektes.
3. Projekt in Eclipse importieren.
4. Erzeugen des Source Codes: Klasse generieren und kompilieren

4.3.1 Herunterladen der Datei

Auf der Internet-Informationssseite des EO-Registers, kann die Zip-Datei « rapg-ws-4-2.zip » heruntergeladen werden.

4.3.2 Erstellen eines leeren Maven-Projektes

Eclipse mit einem neuen Arbeitsbereich öffnen, z.B. C:\workspace\rapg-ws-client und anschliessend mit « File -> New -> Project » ein neues Maven-Projekt erstellen.

Im entsprechenden Menu « Maven -> MavenProject » und danach « Next > » auswählen.

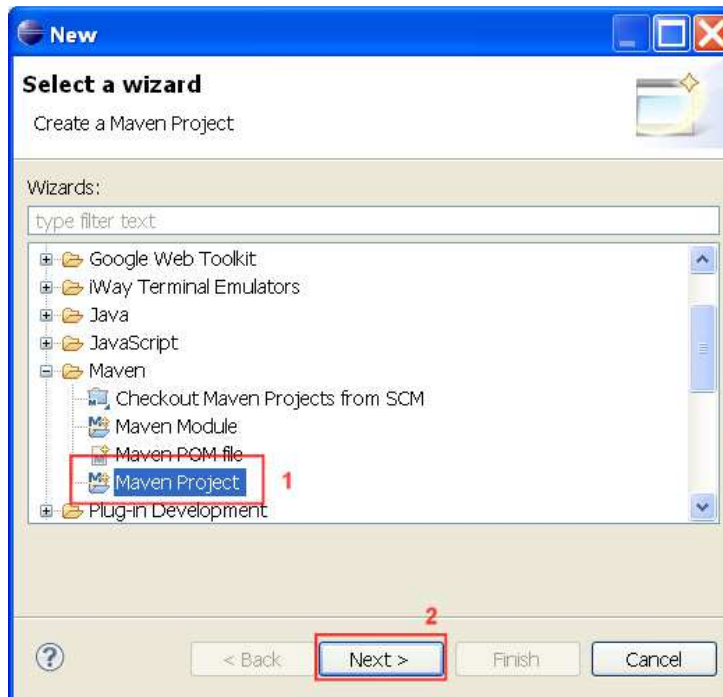


Abbildung 7: Création d'un projet maven vide – Schritt 1

Auswahl von « Create a simple project (skip archetype selection) » und « use default Workspace location », danach weiter mit « Next > ».

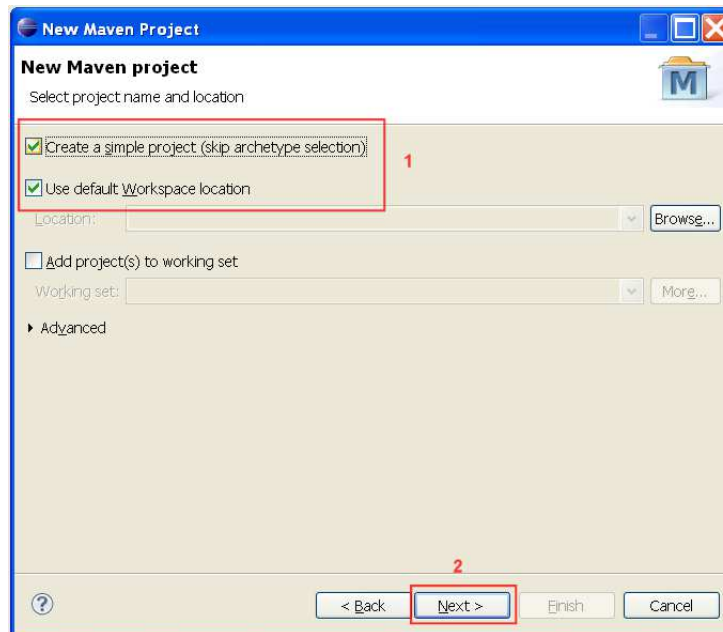


Abbildung 8: Création d'un projet maven vide – Schritt 2

Im nächsten Schritt, muss eine « Group Id » und eine « Artifact Id » eingegeben werden, danach mit « Finish » beenden.

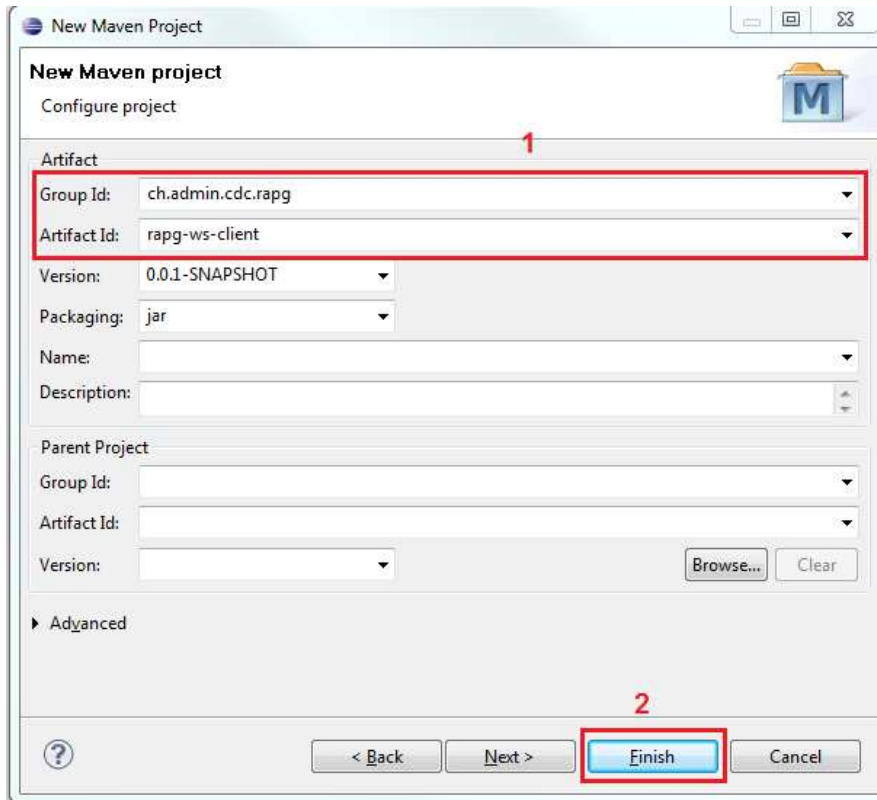


Abbildung 9: Création d'un projet maven vide – Schritt 3

4.3.3 Projekt in Eclipse importieren

Der Menüpunkt « File->Import... » öffnet den Import-Dialog.

Im Import-Dialog « General -> Archive file » und anschliessend « Next > » auswählen.

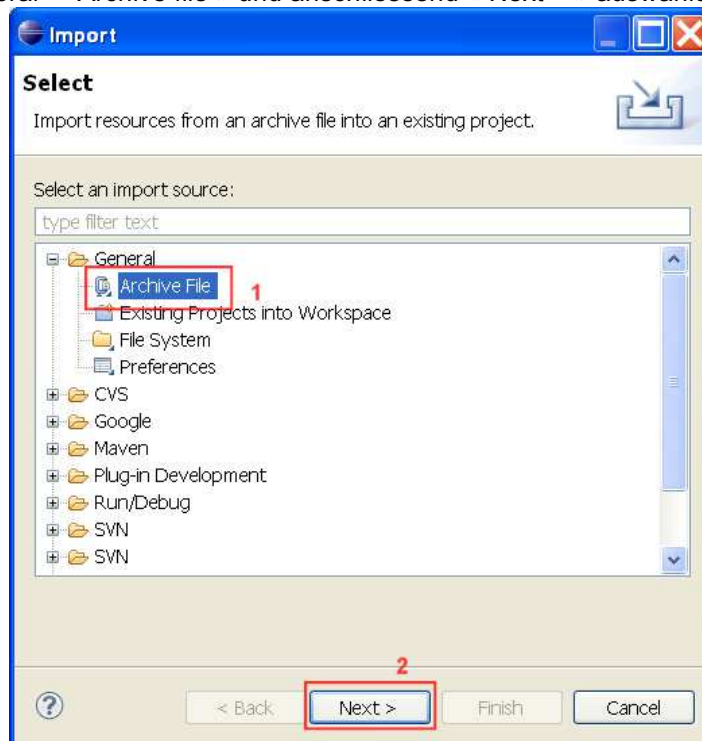


Abbildung 10: Importation du projet – Schritt 1

Das entsprechende Archiv auswählen und die Option « Overwrite existing resources without warning » durch Setzen des Häkchens aktivieren, anschliessend mit « Finish » beenden.

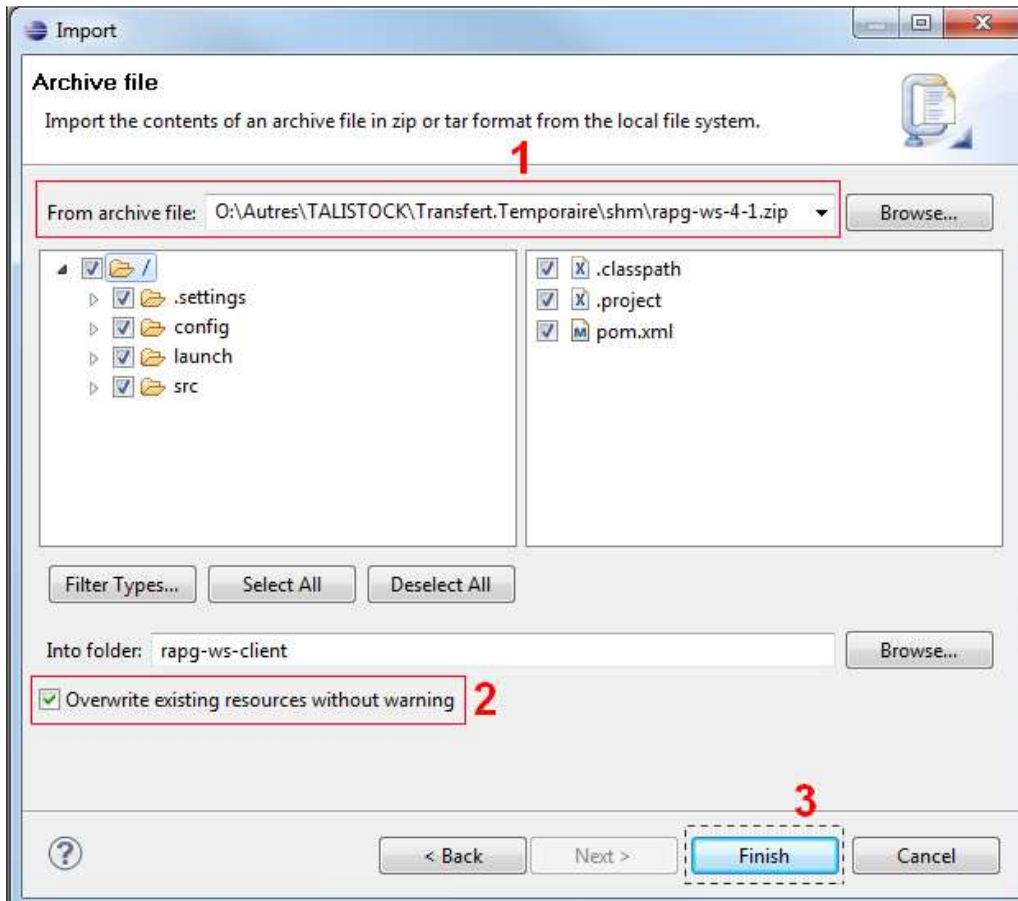


Abbildung 11: Importation du projet – Schritt 2

4.3.4 Erzeugen des Source Codes: Klasse generieren und kompilieren

Auswählen von « Run As > Maven generate-sources ».

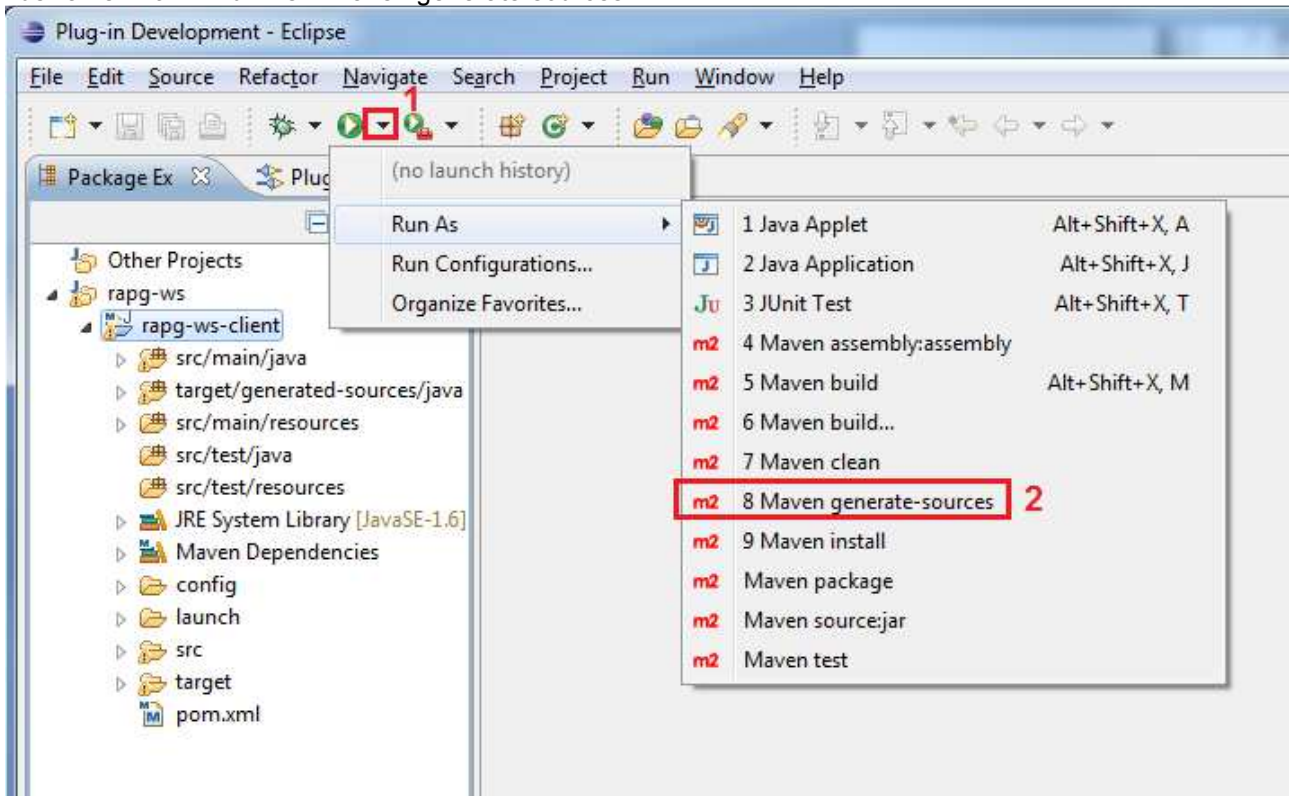


Abbildung 12: Erzeugen des Source Codes: Klasse generieren und kompilieren – Schritt 1

Anschließend den Source Code mittels « Project->Clean » und « Refresh » neu erzeugen.

Als nächstes die Konfiguration des Maven-Projektes auf den neuesten Stand bringen.

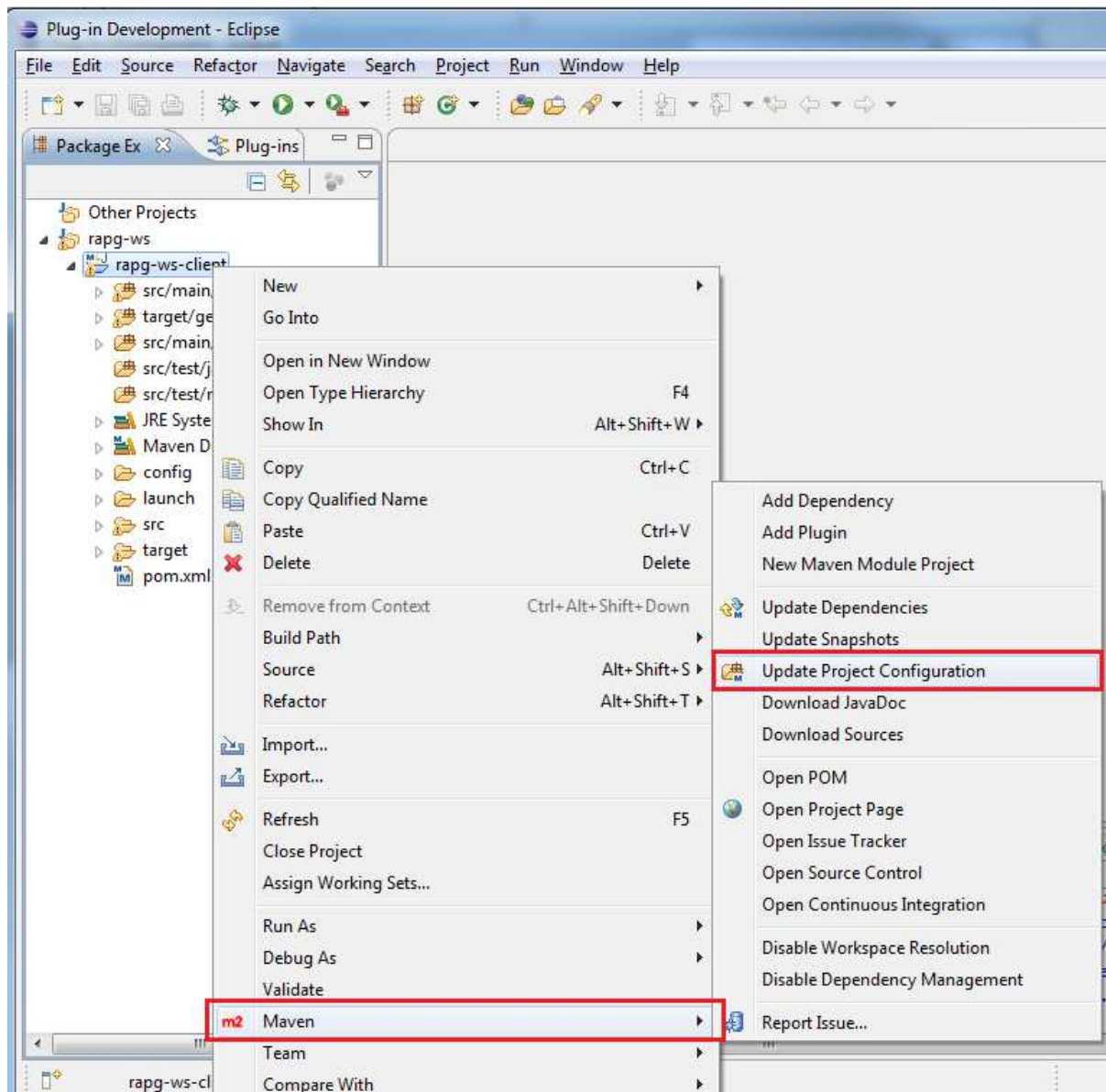


Abbildung 13: Aktualisierung der Konfiguration des Maven-Projektes – Schritt 2

4.4 Struktur des Projektes

Das Projekt besteht aus:

- src/main/java/ ch/admin/cdc/rapg/ws/client/CommandLauchner. « Main » Klasse von der aus der Webservice mit 6 oder 7 Übergabeparametern aufgerufen wird:
 - URL: URL des webservice, z.B. (für Abnahme-Umgebung) <https://www.rapg-a.zas.admin.ch/ApgCore/ws/consultation>
 - certificatFilename: Verzeichnispfad und vollständiger Name des vom BSV herausgegebenen p12 Zertifikates.
 - certificatPassword: Passwort des vom BSV herausgegebenen Zertifikates.
 - vn: 13-stellige Versicherungsnummer
 - officidentifier: Nummer der Ausgleichskasse, z.B. 106
 - branch: Nummer der Zweigstellen, z.B. 1
 - secure: muss immer « true » sein

- src/main/resources:
 - wsdl: Vertrag des Webservices
 - xsd: Schemata, die in der WSDL verwendet werden.
- target/generated-sources /java, répertoire de génération des classes.

4.5 Beschreibung der Klasse CommandLauchner

4.5.1 main

Hier findet eine Prüfung der Parameter vor dem Aufruf des Webservices statt.

4.6 Beschreibung der Klasse RAPGConsultationSSLSession

La Klasse RAPGConsultationSSLSession besteht hauptsächlich aus zwei Methoden:

- findAnnonces()
- getRAPGConsultationService()

Die Methoden sind untenstehend beschrieben.

4.6.1 findAnnonces

Diese Methode besteht aus fünf Einheiten:

1. Erstellen der Services durch Aufruf der Funktion getRAPGConsultationService().
2. Erstellen der Abfrage, Erstellen eines Objektes StandardConsultationInputType mit der Angabe der Versicherten- und der Kassen-Nummer.

```
// 2) Create the request StandardConsultationInputType
final RapgAnnoncesRequestType request = createRequest(nas, cafCode, agencyCode);
...
...
private static RapgAnnoncesRequestType createRequest(final Long nas, final int
cafCode, final int agencyCode) {
    final RapgAnnoncesRequestType request = new RapgAnnoncesRequestType();
    request.setVn(nas.longValue());
    final DeliveryOfficeType caisse = new DeliveryOfficeType();
    caisse.setOfficeIdentifrier(cafCode);
    caisse.setBranch(agencyCode);
    request.setDeliveryOffice(caisse);
    return request;
}
```

3. Aufruf des Webservice

```
//3) Call webservice
final RapgAnnoncesResponseType response =
rapgConsultation.findAnnonces(request);
```

4. Gebrauch der Antwort

```
//4) Use the RegisterStatus answer
for (final RegisterStatusRecordType registerStatus:
response.getMessage().getContent().getRegisterStatusRecords()) {
    //Do what you have to do with the message response

    // It's an exemple to get information
    final String officeNumber =
registerStatus.getDeliveryOffice().getOfficeIdentifrier()
+
(registerStatus.getDeliveryOffice().getBranch() > 0 ? "." +
registerStatus.getDeliveryOffice().getBranch(): ""); //$NON-NLS-1$ //$NON-NLS-2$
    System.out.println("Total APG amount " +
registerStatus.getTotalAPG() //$NON-NLS-1$
+ " delivery office " + officeNumber + "| number of
conflicts: " + (registerStatus.getSimpleIntraConflicts().size() +
registerStatus.getInterConflicts().size())); //$NON-NLS-1$ //$NON-NLS-2$
}
```

5. Gebrauch der Fehlerliste

```
//5) Use the Errors
//5) Use the Errors
```

```
if (response.getErrors() != null) {
    for (final ErreurMessageType error: response.getErrors()) {
        // Make what you need
        // It's an exemple to get error
        System.out.println("Error message: " + error.getMessage()); //$NON-
NLS-1$
    }
}
```

4.6.2 configureSSLOnTheClient

Diese Methode erlaubt es, das Zertifikat auszulesen und dem Objekt « HTTPConduit » hinzuzufügen und dieses dann in der Anfrage mitzusenden.

Die Eingabeparamter der Methode sind:

- Proxy des Webservice
- Name der vom BSV herausgegebenen Zertifikatsdatei inklusive des Verzeichnispfades.
- Passwortes des vom BSV herausgegebenen Zertifikats.

```
// Get httpConduit
final Client client = ClientProxy.getClient(proxy);
final HTTPConduit httpConduit = (HTTPConduit) client.getConduit();
KeyStore ks = null;

try {
    final TLSClientParameters tlsParams = new TLSClientParameters();
    tlsParams.setDisableCNCheck(true);

    // Read the certificate
    FileInputStream filePkcs12;
    ks = KeyStore.getInstance("PKCS12");
    filePkcs12 = new FileInputStream(certFileName);

    // For a better security you can encode your password and decode it here
    ks.load(filePkcs12, certPassword.toCharArray());

    try {
        filePkcs12.close();
    } catch (final IOException e) {
        System.err.println("Error on close " + certFileName + " file");
    }

    // Add certificate to the conduit
    final KeyManagerFactory keyFactory =
KeyManagerFactory.getInstance(KeyManagerFactory.getDefaultAlgorithm());
    keyFactory.init(ks, certPassword.toCharArray());
    final KeyManager[] km = keyFactory.getKeyManagers();
    tlsParams.setKeyManagers(km);

    httpConduit.setTlsClientParameters(tlsParams);

} catch (final FileNotFoundException e) {
    System.err.println("File " + certFileName + " doesn't exist");
} catch (final IOException ioe) {
    System.err.println("File " + certFileName + " doesn't exist. Cause by " +
ioe.getCause());
} catch (final KeyStoreException kse) {
    System.out.println("Security configuration failed with the following: " +
kse.getCause());
} catch (final NoSuchAlgorithmException nsa) {
    System.out.println("Security configuration failed with the following: " +
nsa.getCause());
} catch (final GeneralSecurityException gse) {
    System.out.println("Security configuration failed with the following: " +
```



```
gse.getCause());  
}
```

4.6.3 getRAPGConsultationService

Diese Methode erlaubt es, eine Instanz des Webservice zu erstellen.

Wichtig - in dieser Methode findet man:

1. Die Deklaration von `ENDPOINT_ADDRESS_PROPERTY`, welche es ermöglicht, die gewünschte URL zu bestimmen.
2. Die SSL Konfiguration des Clients, welche das Hinzufügen des Zertifikates im Objekt HTTPConduit erlaubt.

```
// Instantiate the service object generated from wsdl.  
final RapgConsultationService1 rapgConsultationService = new  
RapgConsultationService1();  
//Getting the port. Entry point for the Webservice.  
final RapgConsultation1 port =  
rapgConsultationService.getRapgConsultationPort1();  
  
// Set endpoint address (URL) of the webservice.  
final Map<String, Object> ctxt = ((BindingProvider) port).getRequestContext();  
ctxt.put(BindingProvider.ENDPOINT_ADDRESS_PROPERTY, urlRapgWS);  
  
configureSSLonTheClient(port, certFileName, certPassword);  
return port;
```

4.7 Ausführen der Klasse CommandLauchner

Das P12-Zertifikat muss zugänglich sein. Zum Beispiel kann es in der config-Datei gespeichert sein.

Öffnen des Eingabedialoges « Run configuration »:

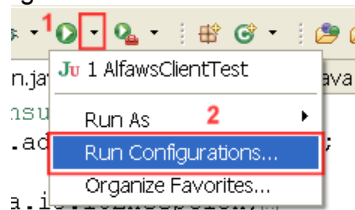


Abbildung 14 : Ausführen der Klasse CommandLauncher – Schritt 1

« Java application » auswählen und anschliessend mit « New launch configuration » weiter gehen.

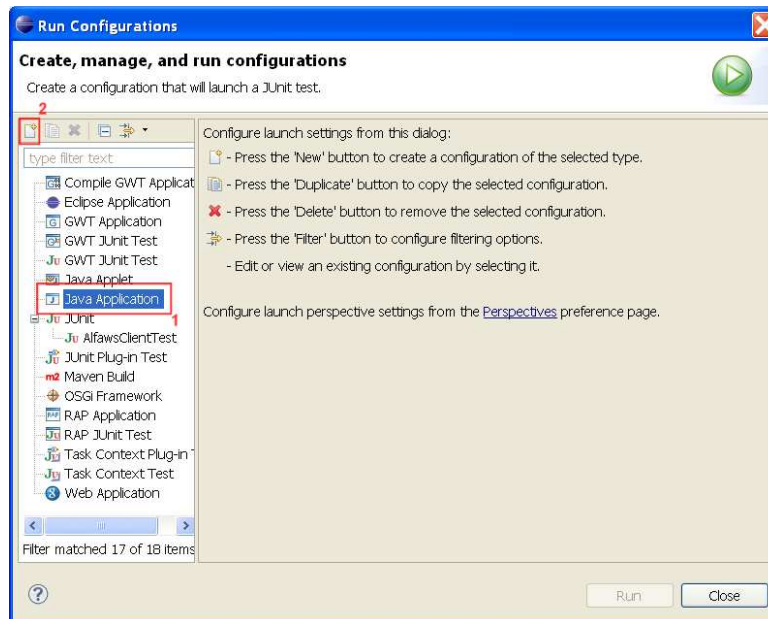


Abbildung 15 : Ausführen der Klasse CommandLauchner – Schritt 2

In dem sich öffnenden Fenster, den Namen (z.B. CommandLauchner) und die « Main class » ch.admin.cdc.rapg.ws.client.CommandLauchner auswählen.

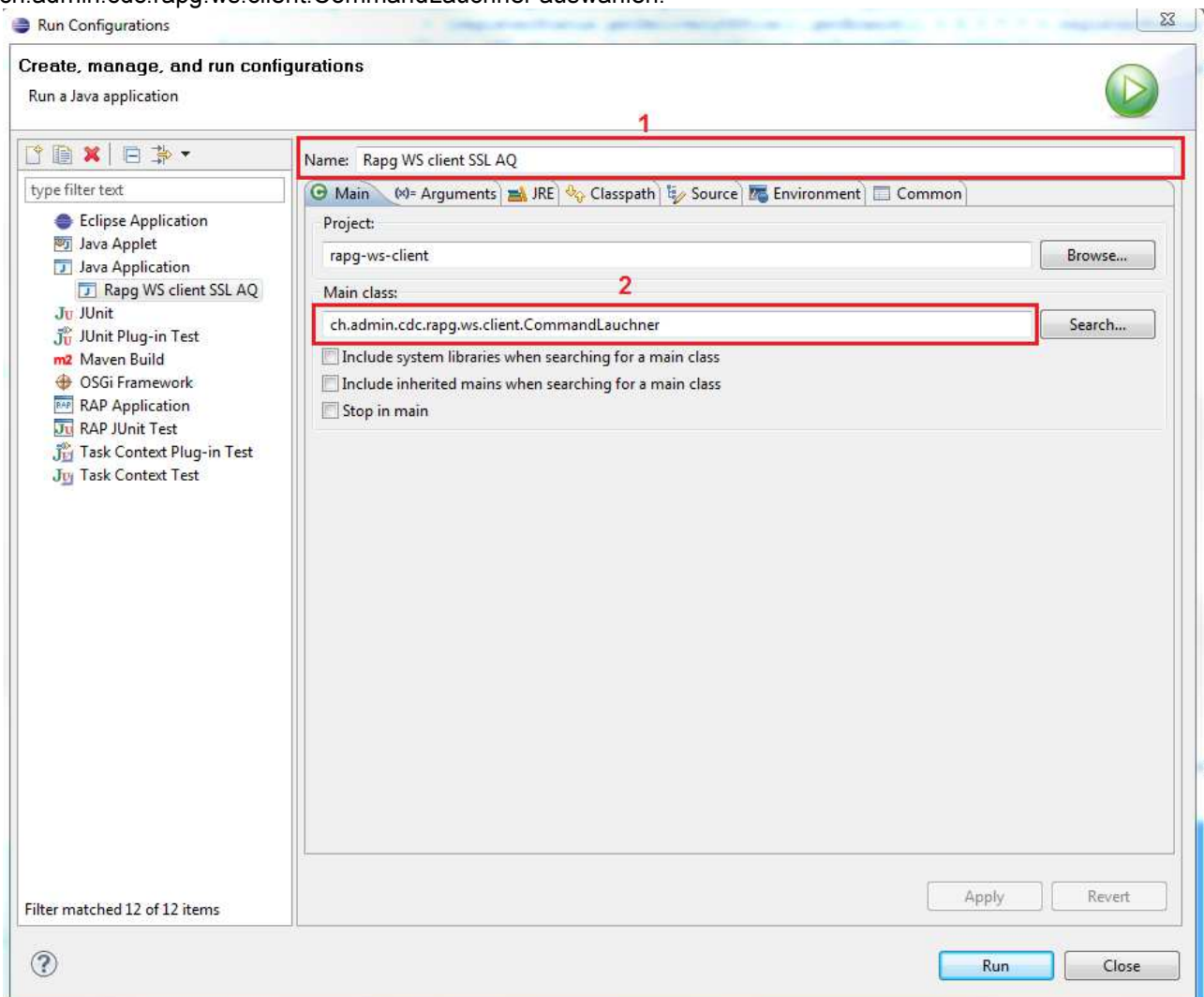


Abbildung 16 : Ausführen der Klasse CommandLauchner – Schritt 3

Projektname Register für Erwerbsausfallentschädigungen
Ergebnisname Benutzerhandbuch –WebService Befragung und Bestellung Auszug

Im Register « Arguments » müssen die Parameter URL, certificatFileName, certificatPassword vn und deliveryOffice [branch] true eingegeben werden.

Beispiel: `https://www.rapg-a.zas.admin.ch/ApgCore/ws/consultation "C:\workspace\rapg-ws-client\rapg-ws-client\config\certificat.p12" certificatPassword 7568572432689 38 0 true`

Und VM argument

Für Prod :-Djavax.net.ssl.trustStore=\${workspace_loc:rapg-ws-client}/config/rapg.keystore-prod.jks

Für Abna :-Djavax.net.ssl.trustStore=\${workspace_loc:rapg-ws-client}/config/rapg.keystore-abna.jks

Auswahl von « Apply », danach weiter mit « Run ».

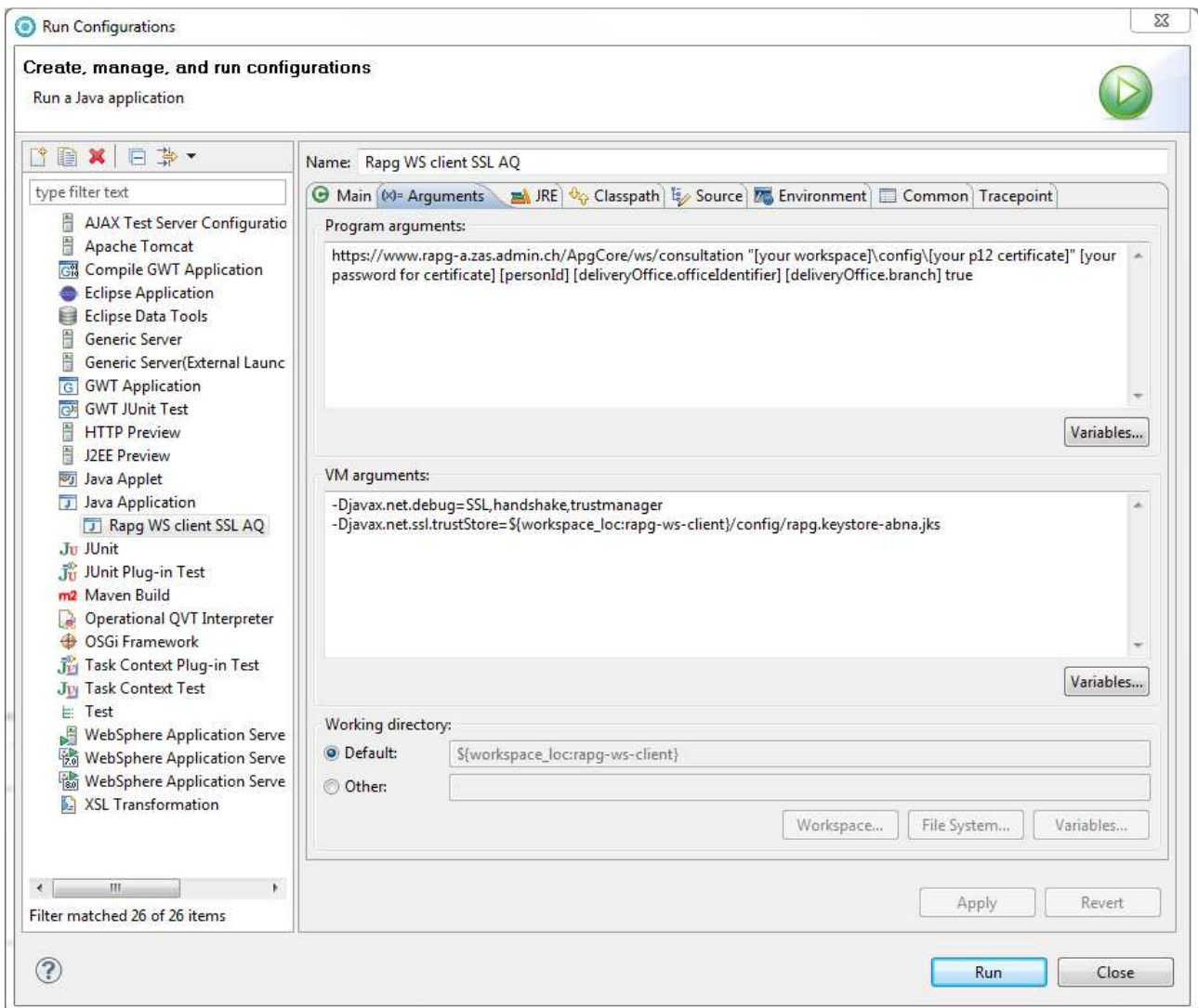


Abbildung 17 : Ausführen der Klasse CommandLauncher – Schritt 4

Falls alle Angaben richtig gemacht wurden, sollten Sie in der Konsole ähnliche Einträge wie in dem untenstehenden Beispiel finden.

```
25 oct. 2013 08:14:31
org.apache.cxf.service.factory.ReflectionServiceFactoryBean buildServiceFromWSDL
INFO: Creating Service
{http://www.zas.admin.ch/rapg/webservice/consultation/1}RapgConsultationService-
1 from WSDL: file:/C:/workspace/rapg-ws-client/rapg-ws-
client/src/main/resources/wsd/rapg-webservice-1-0.wsdl
Create request...
Calling RapgConsultationSSLSession.findAnnonces ...
```

```
Call successful, result: SUCCESS
Number of Records: 2
Total APG amount 6353.60 delivrrery office 38| number of conflicts: 0
Total APG amount 11368.00 delivrrery office 38| number of conflicts: 0
End of test.
```

Abbildung 18 : Antwort des CommandLaunchers

5 Beispiele

5.1 Anfragemeldung

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://www.zas.admin.ch/xmlns/rapg-ws/1" xmlns:ns1="http://www.eahv-iv.ch/xmlns/eahv-iv-2015-
common/4">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:rapgAnnoncesRequest>
      <ns:vn>7569999999999</ns:vn>
      <ns:deliveryOffice>
        <ns1:officeIdentifier>106</ns1:officeIdentifier>
        <ns1:branch>000</ns1:branch>
      </ns:deliveryOffice>
    </ns:rapgAnnoncesRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Abbildung 19: Exemple de message d'appel

5.2 Antwortmeldung

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns4:rapgAnnoncesResponse xmlns="http://www.eahv-iv.ch/xmlns/eahv-iv-2015-
000701/4" xmlns:ns2="http://www.eahv-iv.ch/xmlns/eahv-iv-2015-common/4"
xmlns:ns3="http://www.ofit.admin.ch/xmlns/commun-ws/2"
xmlns:ns4="http://www.zas.admin.ch/xmlns/rapg-ws/1"
xmlns:ns5="http://www.ech.ch/xmlns/eCH-0058/4" xmlns:ns6="http://www.eahv-
iv.ch/xmlns/eahv-iv-2015-000601/4" xmlns:ns7="http://www.ech.ch/xmlns/eCH-
0044/2" xmlns:ns8="http://www.eahv-iv.ch/xmlns/eahv-iv-2015-000301/4"
xmlns:ns9="http://www.eahv-iv.ch/xmlns/eahv-iv-2015-000401/4"
xmlns:ns10="http://www.eahv-iv.ch/xmlns/eahv-iv-2015-000101/4"
xmlns:ns11="http://www.eahv-iv.ch/xmlns/eahv-iv-2015-000501/4">
      <ns4:ack>
        <ns3:value>success</ns3:value>
      </ns4:ack>
      <ns6:message minorVersion="2">
        <ns6:header>
          <ns2:senderId>6-600000-2</ns2:senderId>
          <ns2:recipientId>106</ns2:recipientId>
          <ns2:messageId>0</ns2:messageId>
          <ns2:messageType>2015</ns2:messageType>
          <ns2:sendingApplication>
            <ns5:manufacturer>ZAS</ns5:manufacturer>
            <ns5:product>RAPG</ns5:product>
            <ns5:productVersion>1.5</ns5:productVersion>
          </ns2:sendingApplication>
          <ns2:messageDate>2013-10-25T08:24:21.133+02:00</ns2:messageDate>
          <ns2:eventDate>2013-10-25</ns2:eventDate>
          <ns2:testDeliveryFlag>true</ns2:testDeliveryFlag>
          <ns6:subMessageType>000601</ns6:subMessageType>
          <ns6:action>1</ns6:action>
        </ns6:message>
      </ns4:rapgAnnoncesResponse>
    </soap:Body>
  </soap:Envelope>
```

```
<ns6:responseExpected>>false</ns6:responseExpected>
</ns6:header>
<ns6:content>
  <ns6:commandDetail>
    <ns2:inProgress>All</ns2:inProgress>
    <ns2:inConflict>All</ns2:inConflict>
    <ns2:canceled>All</ns2:canceled>
    <ns2:history>All</ns2:history>
  </ns6:commandDetail>
  <ns6:registerStatusRecords>
    <ns6:deliveryOffice>
      <ns2:officeIdentifier>106</ns2:officeIdentifier>
      <ns2:branch>1</ns2:branch>
    </ns6:deliveryOffice>
<ns6:businessProcessId>106.001.xxxxxxxx</ns6:businessProcessId>
  <ns6:insurant>
    <ns7:vn>756999999999</ns7:vn>
    <ns7:officialName>Monsieur X</ns7:officialName>
    <ns7:firstName>Anonymous</ns7:firstName>
    <ns7:sex>1</ns7:sex>
    <ns7:dateOfBirth>
      <ns7:yearMonthDay>1982-10-02</ns7:yearMonthDay>
    </ns7:dateOfBirth>
  </ns6:insurant>
  <ns6:insurantMaritalStatus>1</ns6:insurantMaritalStatus>
  <ns6:numberOfChildren>0</ns6:numberOfChildren>
  <ns6:insurantDomicile>
    <ns2:canton>25</ns2:canton>
    <ns2:country>100</ns2:country>
  </ns6:insurantDomicile>
  <ns6:accountingMonth>2013-04</ns6:accountingMonth>
  <ns6:serviceType>41</ns6:serviceType>
  <ns6:controlNumber>11</ns6:controlNumber>
  <ns6:activityBeforeService>1</ns6:activityBeforeService>
  <ns6:averageDailyIncome>199.44</ns6:averageDailyIncome>
  <ns6:startOfPeriod>2013-02-04</ns6:startOfPeriod>
  <ns6:endOfPeriod>2013-02-28</ns6:endOfPeriod>
  <ns6:numberOfDays>25</ns6:numberOfDays>
  <ns6:basicDailyAmount>62.00</ns6:basicDailyAmount>
<ns6:dailyIndemnityGuaranteeAI>>false</ns6:dailyIndemnityGuaranteeAI>
  <ns6:allowanceFarm>>false</ns6:allowanceFarm>
  <ns6:allowanceCareExpenses>0.00</ns6:allowanceCareExpenses>
  <ns6:totalAPG>1550.00</ns6:totalAPG>
  <ns6:paymentMethod>2</ns6:paymentMethod>
  <ns6:history>>false</ns6:history>
  <ns6:creationDate>2013-05-02</ns6:creationDate>
  <ns6:breakRules>
    <ns2:breakRuleCode>502</ns2:breakRuleCode>
  </ns6:breakRules>
  <ns6:orphan>>false</ns6:orphan>
  <ns6:canceled>>false</ns6:canceled>
  <ns6:rawRecords>
    <ns6:deliveryOffice>
      <ns2:officeIdentifier>106</ns2:officeIdentifier>
      <ns2:branch>1</ns2:branch>
    </ns6:deliveryOffice>
<ns6:businessProcessId>106.001.xxxxxxxx</ns6:businessProcessId>
  <ns6:insurant>
    <ns7:vn>756999999999</ns7:vn>
```

```
        <ns7:officialName>Monsieur X</ns7:officialName>
        <ns7:firstName>Anonymous</ns7:firstName>
        <ns7:sex>1</ns7:sex>
        <ns7:dateOfBirth>
          <ns7:yearMonthDay>1982-10-02</ns7:yearMonthDay>
        </ns7:dateOfBirth>
      </ns6:insurant>
      <ns6:insurantMaritalStatus>1</ns6:insurantMaritalStatus>
      <ns6:numberOfChildren>0</ns6:numberOfChildren>
      <ns6:insurantDomicile>
        <ns2:canton>25</ns2:canton>
        <ns2:country>100</ns2:country>
      </ns6:insurantDomicile>
      <ns6:accountingMonth>2013-04</ns6:accountingMonth>
      <ns6:serviceType>41</ns6:serviceType>
      <ns6:controlNumber>11</ns6:controlNumber>
      <ns6:activityBeforeService>1</ns6:activityBeforeService>
      <ns6:averageDailyIncome>199.44</ns6:averageDailyIncome>
      <ns6:startOfPeriod>2013-02-04</ns6:startOfPeriod>
      <ns6:endOfPeriod>2013-02-28</ns6:endOfPeriod>
      <ns6:numberOfDays>25</ns6:numberOfDays>
      <ns6:basicDailyAmount>62.00</ns6:basicDailyAmount>
    <ns6:dailyIndemnityGuaranteeAI>false</ns6:dailyIndemnityGuaranteeAI>
      <ns6:allowanceFarm>false</ns6:allowanceFarm>
      <ns6:allowanceCareExpenses>0.00</ns6:allowanceCareExpenses>
      <ns6:totalAPG>1550.00</ns6:totalAPG>
      <ns6:paymentMethod>2</ns6:paymentMethod>
      <ns6:history>false</ns6:history>
      <ns6:creationDate>2013-05-02</ns6:creationDate>
      <ns6:action>1</ns6:action>
    </ns6:rawRecords>
  </ns6:registerStatusRecords>
</ns6:content>
</ns6:message>
</ns4:rapgAnnoncesResponse>
</soap:Body>
</soap:Envelope>
```