

INTERNETMARKE

Webservice 1C4A

Kurzanleitung

Einstufung des Dokuments lt. SHB IT:

<**offen** | nur für den internen Gebrauch | vertraulich | streng vertraulich>

Version: 1.0 / Stand 08.03.2018

Diese Kurzanleitung wurde mit großer Sorgfalt erstellt und steht Ihnen unentgeltlich als Hilfe für die Entwicklung von Schnittstellen zu unseren Systemen zur Verfügung. Eine Haftung oder Garantie für die Aktualität, Richtigkeit und Vollständigkeit der zur Verfügung gestellten Informationen wird aber nicht übernommen.

Copyright © Deutsche Post AG 2017

Deutsche Post AG	INTERNETMARKE Webservice 1C4A für Drittanbieter	Seite 2 / 19
Kurzanleitung		

Inhaltsverzeichnis

1	Über dieses Dokument	3
1.1	Zweck	3
1.2	Zielgruppe	3
1.3	Serviceversion	3
1.4	Mitgeltende Dokumente	3
1.5	Weiterführende Hilfe	3
2	Funktionsweise	4
2.1	Benutzer- und Portokassenmanagement	4
2.2	Internetmarkenerzeugung	4
2.3	Informationsbereitstellung	5
3	Voraussetzungen für die Kommunikation des Drittanbietersystems.....	6
3.1	Bedingungen für den SOAP-Request	6
3.2	Form der Zieladresse.....	6
3.3	Aufbau des SOAP-Requests	6
3.4	Aufbau des Headers	6
3.5	Prüfsummenberechnung	7
4	Aktionen	9
4.1	AuthenticateUser	9
4.2	CheckoutShoppingCartPDF	10
5	Anhang	13
5.1	Programmbeispiel für die Prüfsummenberechnung in Java	13
5.2	Programmbeispiel für die Prüfsummenberechnung in C#	16
5.3	Glossar	19

Deutsche Post AG	INTERNETMARKE Webservice 1C4A für Drittanbieter	Seite 3 / 19
Kurzanleitung		

1 Über dieses Dokument

1.1 Zweck

Dieses Dokument vermittelt die Grundlagen für das Einbinden des OneClickForApplication-Webservices (1C4A) bei Drittanbietern. Anhand des Minimalbeispiels im Kapitel 4 *Aktionen* ab Seite 9 kann der 1C4A-Webservice getestet werden.

1.2 Zielgruppe

Dieses Dokument richtet sich an die Entwickler bei Drittanbietern, welche 1C4A in ihre eigenen Applikationen einbinden möchten.

1.3 Serviceversion

Dieses Dokument bezieht sich auf die Serviceversion 3 des 1C4A-Webservices.

Hinweis: In diesem Dokument werden die technischen Bezeichner für die jeweiligen Aktionen verwendet.

1.4 Mitgeltende Dokumente

Dokument	Ablage
Webservice 1C4A_Informationspaket für Partner	ZIP-Datei: DP-Webservice 1C4A Partneranbindung INTERNETMARKE
Technische WEB Service Definition (WSDL)	https://internetmarke.deutschepost.de/OneClickForAppV3?wsdl
Produkt-Webservice (ZIP-Datei mit allen Inhalten zum ProdWS)	ZIP-Datei: DP-Webservice 1C4A Partneranbindung INTERNETMARKE

1.5 Weiterführende Hilfe

Die ausführliche Darstellung der 1C4A-Schnittstelle enthält das Dokument „INTERNET-MARKE – Technische Servicebeschreibung“. Sind Fragen in diesen beiden Dokumenten nicht geklärt, wenden Sie sich per E-Mail an den Support (pcf-1click@deutschepost.de).

Deutsche Post AG	INTERNETMARKE Webservice 1C4A für Drittanbieter	Seite 4 / 19
Kurzanleitung		

2 Funktionsweise

Der 1C4A-Webservice dient dem Vertrieb von Internetmarken, die in der Anwendung INTERNETMARKE der Deutsche Post AG erzeugt werden. Die Erzeugung der INTERNET-MARKE erfolgt über eine individuelle, dem Kunden zugeordnete Portokasse.

Der 1C4A-Webservice besteht aus drei Blöcken von Aktionen:

- Benutzer- und Portokassenmanagement,
- Internetmarkenerzeugung und
- Informationsbereitstellung.

2.1 Benutzer- und Portokassenmanagement

In dieser Kurzanleitung wird aus dem Block „Benutzer- und Portokassenmanagement“ nur die Aktion „AuthenticateUser“ beschrieben. Das Referenzdokument „Technische Servicebeschreibung“ enthält die übrigen Aktionen mit ihren entsprechenden Erläuterungen.

Aktion	Technischer Bezeichner	Kurzbeschreibung
Benutzer authentifizieren	AuthenticateUser	Diese Aktion authentifiziert den Kunden bzw. dessen Portokasse und autorisiert nachfolgende Abfragen zur Nutzung der mit der E-Mail-Adresse verknüpften Portokasse.

2.2 Internetmarkenerzeugung

In dieser Kurzanleitung wird bei der Internetmarkenerzeugung ausschließlich auf die Variante im PDF-Format eingegangen. Sie können neben dem PDF-Format auch PNG als Dateiformat für die Internetmarkenerzeugung verwenden. Eine kurze Beschreibung finden Sie in der nachstehenden Tabelle. Die detaillierte Beschreibung finden Sie im Referenzdokument „INTERNETMARKE – Technische Servicebeschreibung“.

Aktion	Technischer Bezeichner	Kurzbeschreibung
PDF-Internetmarken erstellen	CheckoutShoppingCartPDF	Die Aktion validiert den Warenkorb und die Bezahlung. Die Aktion bucht die Kosten für die Transaktion von der Portokasse ab und liefert als Response die URL auf eine PDF-Datei, welche die Internetmarken enthält.

Deutsche Post AG	INTERNETMARKE Webservice 1C4A für Drittanbieter	Seite 5 / 19
Kurzanleitung		

Aktion	Technischer Bezeichner	Kurzbeschreibung
PNG-Internetmarke erstellen	CheckoutShoppingCartPNG	Die Aktion ist vergleichbar mit der Aktion CheckoutShoppingCartPDF, liefert jedoch anstelle eines Links auf ein PDF einen Link auf eine ZIP-Datei mit Darstellungen der gekauften Frankierungen im PNG-Format.

2.3 Informationsbereitstellung

Die Aktion der nachfolgenden Tabelle ist optional. Weitere optionale Aktionen finden Sie in der Referenzdokumentation „INTERNETMARKE – Technische Servicebeschreibung“.

Aktion	Technischer Bezeichner	Kurzbeschreibung
Abfrage der gültigen Druckformate	retrievePageFormats	Die Aktion dient der Abfrage nach gültigen Druckformaten, die in der INTERNETMARKE angelegt wurden.

Deutsche Post AG	INTERNETMARKE Webservice 1C4A für Drittanbieter	Seite 6 / 19
Kurzanleitung		

3 Voraussetzungen für die Kommunikation des Drittanbietersystems

3.1 Bedingungen für den SOAP-Request

Die 1C4A-Schnittstelle erwartet einen SOAP-Request auf Basis des Transportprotokolls HTTPS. Bei allen Anfragen muss

- auf Transportebene die SSL-Verschlüsselung aktiviert sein und
- ein drittanbieterspezifischer SOAP-Header samt Signatur mitgegeben werden.

3.2 Form der Zieladresse

Produktiv wird der Webservice über folgende URL angesprochen.

<https://internetmarke.deutschepost.de/OneClickForAppV3>

3.3 Aufbau des SOAP-Requests

Pro Aktion sind optionale und notwendige Felder bei der Anfrage (Request) vorgegeben, die im SOAP-Body angegeben werden müssen. Ob die Anfrage strukturell korrekt ist und alle Pflichtelemente vorhanden sind, wird bei jedem Request mittels SchemaValidation geprüft.

Alle fachlichen Fehlertypen und deren fachliche Fehlercodes sind in der WSDL beschrieben. Alle Fehler werden in einem SOAP-Fault-Element eingebettet zurückgegeben, wobei die fachlichen Fehler als Typen in der WSDL vorab definiert sind.

Technische Fehler werden als Java-Laufzeitfehler an den Servicenehmer zurückgemeldet.

Der Servicenehmer kann sich darauf verlassen, dass immer die in der WSDL fest definierten Fehlercodes zurückgegeben werden, wenn der fachliche Fehlertyp ein ID-Element beinhaltet. Bei bestimmten Fehlertypen, die nur eine einzige fachliche Fehlersituation widerspiegeln, wird auf das ID-Element verzichtet.

Hinweis: Der Servicenehmer sollte vermeiden, Prüfungen auf Textinhalte von Message-Elementen in fachlichen Fehlertypen vorzunehmen, da diese ohne Ankündigung geändert werden können.

Bevor ein SOAP-Request zur Verarbeitung an die Business-Logik der 1C4A-Schnittstelle weitergeleitet wird, erfolgt eine Validierung gegen das Schema der WSDL. Der Schema-Validator prüft, ob der Request strukturell dem Schema entspricht und ob alle Pflichtelemente vorhanden sind. Bei Nichterfüllung wird eine SOAP-Fault SchemaValidationException zurückgegeben.

3.4 Aufbau des Headers

Bei jedem SOAP-Request erwartet der 1C4A-Webservice den folgenden Header.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v3="http://oneclickforapp.dpag.de/V3 " >
  <soapenv:Header>
    <v3:PARTNER_ID>DPPAR</v3:PARTNER_ID>
    <v3:REQUEST_TIMESTAMP>28012014-142729</v3:REQUEST_TIMESTAMP>
```

Deutsche Post AG	INTERNETMARKE Webservice 1C4A für Drittanbieter	Seite 7 / 19
Kurzanleitung		

```

<v3:KEY_PHASE>1</v3:KEY_PHASE>
<v3:PARTNER_SIGNATURE>58578270</v3:PARTNER_SIGNATURE>
</soapenv:Header>
<soapenv:Body>
<-- Nutzinformationen zugunsten der Verständlichkeit entfernt -->
</soapenv:Body>
</soapenv:Envelope>

```

Die nachfolgenden Header-Elemente müssen in jedem Request gesendet werden. Die Reihenfolge der Header-Elemente ist beliebig.

Parameter	Beschreibung	Typ	Länge	Werte aus dem Beispiel
PARTNER_ID	Das Versandpartnerkennzeichen muss in der Applikation konfiguriert werden, andernfalls wird der Request abgelehnt.	String	5	DPPAR
REQUEST_TIMESTAMP	Dieser Parameter enthält den Absendezeitpunkt des Requests mit Datum und Uhrzeit. im Format DDMMYYYY-HHMMSS	String	15	28012014-142729
KEY_PHASE	Der Parameter enthält die vereinbarte Version des Geheimnisses.	Integer	3	1
PARTNER_SIGNATURE	Der Parameter enthält die Signatur des Requests, die über die definierten Felder berechnet wird.	String	8	58578270

3.5 Prüfsummenberechnung

3.5.1 Manuelle Prüfsummenberechnung

Die Signatur wird als MD5-Hash berechnet, basierend auf den Inhalten der folgenden Felder.

- PARTNER_ID
- REQUEST_TIMESTAMP
- KEY_PHASE
- SIGNATURSCHLÜSSEL
Länge 32 Zeichen; Der Inhalt ist statisch und wird dem Partner der Deutsche Post AG mitgeteilt.

Deutsche Post AG	INTERNETMARKE Webservice 1C4A für Drittanbieter	Seite 8 / 19
Kurzanleitung		

Für die Prüfsummenbildung werden die Inhalte der o. g. Felder aneinandergehängt. Alle Feldinhalte werden rechts und links um Blanks bereinigt und mit dem Trenner „:“ versehen konkateniert.

Das Ergebnis wird als Eingabe für MD5 verwendet und ist ein Hashwert mit der Länge von 16 Byte. Die Bytes werden in ihrer hexadezimalen Darstellung (mit a–f in Kleinbuchstaben) in einen Text umgewandelt. Hexadezimale Darstellungen von Bytes, die nur über eine Stelle verfügen (0–f) werden um eine führende Null erweitert (00–0f).

Vom daraus resultierenden 32 Zeichen langen Text werden die ersten 8 Zeichen von links betrachtet. Diese werden in dem Feld PARTNER_SIGNATURE abgelegt.

Auf der Empfängerseite sind dieselben Schritte zur Verifizierung von PARTNER_SIGNATURE durchzuführen.

3.5.2 Automatisierte Prüfsummenberechnung

Mit dem nachfolgenden Unixkommando, getestet auf RHEL 6 in einer Bash-Shell, können Sie die Prüfsummenberechnung durchführen. Die orange markierten Kommandoteile ersetzen Sie durch Ihre individuellen Kundendaten.

```
echo -n "DPPAR::28012014-142729::1::Das_Ist_der_neue_Key" | md5sum | cut -b 1-8
```

Dabei sind:

DPPAR	die ID für den Drittanbieter
28012014-142729	das aktuelle Requestdatum mit Uhrzeit
1	die Versionsnummer des Geheimnisses
Das_Ist_der_neue_Key	das Geheimnis

Variabel sind hier das Requestdatum und die Uhrzeit.

Hinweis: Im Anhang befindet sich je ein Programmbeispiel zur Prüfsummenberechnung in Java und in C#.

Deutsche Post AG	INTERNETMARKE Webservice 1C4A für Drittanbieter	Seite 9 / 19
Kurzanleitung		

4 Aktionen

Dieses Kapitel enthält ein funktionierendes Minimalbeispiel mit den Aktionen „AuthenticateUser“ und „CheckoutShoppingCartPDF“. Die orange markierten Code-Teile ersetzen Sie durch Ihre individuellen Kundendaten. Die blau markierten Code-Teile werden von der Response zum nächsten Request übertragen.

4.1 AuthenticateUser

4.1.1 Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v3="http://oneclickforapp.dpag.de/V3">
  <soapenv:Header>
    <v3:PARTNER_ID>DPPAR</v3:PARTNER_ID>
    <v3:REQUEST_TIMESTAMP>28012014-142729</v3:REQUEST_TIMESTAMP>
    <v3:KEY_PHASE>1</v3:KEY_PHASE>
    <v3:PARTNER_SIGNATURE>58578270</v3:PARTNER_SIGNATURE>
  </soapenv:Header>
  <soapenv:Body>
    <AuthenticateUserRequest xmlns="http://oneclickforapp.dpag.de/V2">
      <username>max.mustermann@deutschepost.de</username>
      <password><![CDATA[vH$0kF*z40]]></password>
    </AuthenticateUserRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Parameter	Beschreibung
password	Passwort, das bei der Registrierung des INTERNETMARKE-Benutzers vergeben wurde
username	E-Mail-Adresse, mit welcher die Benutzerregistrierung erfolgte

Die übrigen Parameter sowie erlaubte Werte und Datentypen finden Sie in der Referenzdokumentation „INTERNETMARKE – Technische Servicebeschreibung“.

4.1.2 Response

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header/>
  <S:Body>
    <AuthenticateUserResponse xmlns="http://oneclickforapp.dpag.de/V3">
      <userToken>5N0i3VHvC9PukGUKSzTYsHL/JrMgG8+15BAY/8wgG5CSj6tzy1jJmg==</userToken>
      <walletBalance>67424</walletBalance>
      <showTermsAndConditions>>false</showTermsAndConditions>
    </AuthenticateUserResponse>
  </S:Body>
</S:Envelope>
```

Deutsche Post AG	INTERNETMARKE Webservice 1C4A für Drittanbieter	Seite 10 / 19
Kurzanleitung		

Parameter	Beschreibung
showTermsAndConditions	Lizenzvereinbarung
userToken	Token zur Bestätigung der Authentifizierung und zur Autorisierung der Nutzung einer Portokasse für den Kauf
walletBallance	Portokassenstand in Eurocent nach dem Kauf bzw. zum Zeitpunkt der Authentifizierung

Die übrigen Parameter sowie erlaubte Werte und Datentypen finden Sie in der Referenzdokumentation „INTERNETMARKE – Technische Servicebeschreibung“.

4.2 CheckoutShoppingCartPDF

4.2.1 Request

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v3="http://oneclickforapp.dpag.de/V3">
  <soapenv:Header>
    <v3:PARTNER_ID>DPPAR</v3:PARTNER_ID>
    <v3:REQUEST_TIMESTAMP>28012014-142729</v3:REQUEST_TIMESTAMP>
    <v3:KEY_PHASE>1</v3:KEY_PHASE>
    <v3:PARTNER_SIGNATURE>58578270</v3:PARTNER_SIGNATURE>
  </soapenv:Header>
  <soapenv:Body>
    <v3:CheckoutShoppingCartPDFRequest>
      <v3:userToken>5N0i3VHvC9PukGUKSzTYsHL/JrMgG8+15BAY/8wgG5CSj6tzy1jJmg==</v3:userToken>
      <v3:pageFormatId>1</v3:pageFormatId>
      <v3:positions>
        <v3:productCode>1</v3:productCode>
        <v3:voucherLayout>AddressZone</v3:voucherLayout>
        <v3:position>
          <v3:labelX>1</v3:labelX>
          <v3:labelY>1</v3:labelY>
          <v3:page>1</v3:page>
        </v3:position>
      </v3:positions>
      <v3:total>70</v3:total>
    </v3:CheckoutShoppingCartPDFRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Parameter	Beschreibung
labelX	X-Position (Spalte) auf der Seite
labelY	Y-Position (Zeile) auf der Seite

Deutsche Post AG	INTERNETMARKE Webservice 1C4A für Drittanbieter	Seite 11 / 19
Kurzanleitung		

Parameter	Beschreibung
page	Nummer der Seite, auf welcher sich die Marke befindet
pageFormatId	ID des Druckformats, auf das der Ausdruck erfolgt
position	Positionierung des Frankiervermerks auf dem PDF-Dokument (Seite, Zeile und Spalte)
positions	Warenkorbpositionen
productCode	ID, die das Produkt innerhalb der Produkt- und Preisliste identifiziert
total	Gesamtwert des Warenkorbs in Eurocent
userToken	Token zur Bestätigung der Authentifizierung und zur Autorisierung der Nutzung einer Portokasse für den Kauf
voucherLayout	Vorgesehene Positionierung der Frankierung auf dem Brief (Adressfenster bzw. Frankierzone), bestimmt das Aussehen des Frankiervermerks

Die übrigen Parameter sowie erlaubte Werte und Datentypen finden Sie in der Referenzdokumentation „INTERNETMARKE – Technische Servicebeschreibung“.

4.2.2 Response

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header/>
  <S:Body>
    <CheckoutShoppingCartPDFResponse xmlns="http://oneclickforapp.dpag.de/V3">

<link>https://internetmarke.deutschepost.de/PcfExtensionWeb/document?keyphase=1&data=ORwfSjJTHkSUF4dj6mEfahg3St8HEBbf</link>
    <walletBallance>67354</walletBallance>
    <shoppingCart>
      <shopOrderId>718474626</shopOrderId>
      <voucherList>
        <voucher>
          <voucherId>A0011B1B9D0000008C71</voucherId>
        </voucher>
      </voucherList>
    </shoppingCart>
  </CheckoutShoppingCartPDFResponse>
</S:Body>
</S:Envelope>
```

Deutsche Post AG	INTERNETMARKE Webservice 1C4A für Drittanbieter	Seite 12 / 19
Kurzanleitung		

Parameter	Beschreibung
link	URL, die auf eine PDF-Datei verweist, welche die gekauften Frankiervermerke enthält
shopOrderId	Bestellnummer
shoppingCart	Warenkorb
voucherList	Liste der Frankier-IDs (voucherId)
walletBallance	Portokassenstand in Eurocent nach dem Kauf bzw. zum Zeitpunkt der Authentifizierung

Die übrigen Parameter sowie erlaubte Werte und Datentypen finden Sie in der Referenzdokumentation „INTERNETMARKE – Technische Servicebeschreibung“.

Deutsche Post AG	INTERNETMARKE Webservice 1C4A für Drittanbieter	Seite 13 / 19
Kurzanleitung		

5 Anhang

5.1 Programmbeispiel für die Prüfsummenberechnung in Java

```

/*
 * Copyright(c) 2000-2018 Deutsche Post DHL Group. All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions are met:
 *
 * 1. Redistributions of source code must retain the above copyright notice,
 *    this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright
 *    notice, this list of conditions and the following disclaimer in the
 *    documentation and/or other materials provided with the distribution.
 * 3. Neither the name of the copyright holder nor the names of its
 *    contributors may be used to endorse or promote products derived from
 *    this software without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE
 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
 * POSSIBILITY OF SUCH DAMAGE.
 */
import java.io.UnsupportedEncodingException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 * Deutsche Post DHL Group - Internetmarke.
 *
 * Example of an MD5 signing / hashing algorithm, to generate the hashes to
 * be enclosed in the Internetmarke interface SOAP headers.
 *
 * You may freely take the code for your Internetmarke handling stubs.
 */
public class MD5Sample {
    /** Logging output for the hashing algorithm. */
    private static final Logger logger = Logger.getAnonymousLogger();

```

Deutsche Post AG	INTERNETMARKE Webservice 1C4A für Drittanbieter	Seite 14 / 19
Kurzanleitung		

```

/** The identification of the MD5 hashing algorithm. */
public static final String MD5 = "MD5";
/** Encoding style for the DPDHL Internetmarke. */
public static final String ENCODING_STYLE = "UTF-8";
/** Separator character between the fields to encode / hash. */
public static final String FIELDS_SEPARATOR = "::*";
/** Maximum string size for the fields to be hashed. */
public static final int MAX_FIELD_STRING_SIZE = 256;
/** Maximum length of the hash string representation. */
public static final int MAX_HASH_STRING_SIZE = 8;

/**
 * Main function to start the example algorithm from the command line.
 * @param args Command line arguments.
 */
public static void main(String args[]) {

    DateFormat formatter = new SimpleDateFormat( "ddMMyyyy-HH:mm:ss" );
    String theDate = formatter.format( new Date() );

    // String theDate = "28012014-14:27:29"; // set a fixed date as an example

    // Example: To be replaced by the partner id provided by Deutsche Post
    String thePartnerID = "DPPAR";
    // Example: To be replaced by the secret provided by Deutsche Post
    String thePartnerSecret = "Das_Ist_der_neue_Key";
    // Example: To be replaced by the key phase provided by Deutsche Post
    String theKeyPhase = "1";

    MD5Sample md5Sample = new MD5Sample();

    // The main part, hashing of the given fields.
    byte[] hash = md5Sample.sign(new String[] {
        thePartnerID, theDate,theKeyPhase },
        thePartnerSecret);

    if (hash != null) {
        String theSignature = md5Sample.hashToString(hash);

        // Debugging output...
        logger.info("Hash signature (as string): " + theSignature);
    }
}

/**
 * Signs a given set of parameter fields.
 *
 * @param theFields Set of fields that are included in the signature and
 * are to be signed.
 * @param theSignatureKey Signature key, added to the list of fields.

```

Kurzanleitung

```
* @return byte[] Signature of the field set.
*/
public byte[] sign(String[] theFields, String theSignatureKey) {

    if (theFields != null && theFields.length > 0) {

        // Prepare a string buffer to assemble fields string
        StringBuffer fieldsString = new StringBuffer(MAX_FIELD_STRING_SIZE);

        // move all fields to be signed into string buffer with separators
        for (int i = 0; i < theFields.length; i++) {
            if (theFields[i] != null) {
                fieldsString.append(theFields[i].trim());
            }
            fieldsString.append(FIELDS_SEPARATOR);
        }
        fieldsString.append(theSignatureKey);

        // Debugging output....
        logger.info("String representation of fields: " + fieldsString.toString());

        try {

            // Get the hashing algorithm
            MessageDigest theDigest = MessageDigest.getInstance(MD5);
            theDigest.reset();

            // hash the fields string
            theDigest.update(fieldsString.toString().getBytes(ENCODING_STYLE));

            return theDigest.digest();

        } catch (NoSuchAlgorithmException nsae) {
            logger.log(Level.SEVERE, nsae.toString(), nsae);
        } catch (UnsupportedEncodingException uee) {
            logger.log(Level.SEVERE, uee.toString(), uee);
        }
    }

    // Some sort of error occurred, just return null
    return null;
}

/**
 * Converts the byte hash into a string representation
 * (human-readable), Base16 encoding truncated to 8 characters.
 * @param hash The hash (byte string) to be converted.
 * @return String String representation of the hash.
 */
public String hashToString(byte[] hash) {
```

Deutsche Post AG	INTERNETMARKE Webservice 1C4A für Drittanbieter	Seite 16 / 19
Kurzanleitung		

```

StringBuffer hashStringRepresentation = new StringBuffer();

for (int i = 0; i < hash.length; i++) {
    String hexByteString = Integer.toHexString(0xFF & hash[i]);
    if (hexByteString.length() == 1) {
        hashStringRepresentation.append("0");
    }
    hashStringRepresentation.append(hexByteString);
}

return hashStringRepresentation.substring(0, MAX_HASH_STRING_SIZE);
}

```

5.2 Programmbeispiel für die Prüfsummenberechnung in C#

```

/*
 * Copyright (C) 2000-2018 Deutsche Post DHL Group. All rights reserved.
 * This file and its contents are the sole property of Deutsche Post DHL Group.
 *
 * Redistribution and use in source and binary forms, with or without modification,
 * are permitted provided that the following conditions are met:
 *
 * 1. Redistributions of source code must retain the above copyright notice,
 * this list of conditions and the following disclaimer.
 * 2. Redistributions in binary form must reproduce the above copyright notice,
 * this list of conditions and the following disclaimer in the documentation
 * and/or other materials provided with the distribution.
 * 3. Neither the name of the copyright holder nor the names of its contributors
 * may be used to endorse or promote products derived from this software
 * without specific prior written permission.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
 * IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
 * INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING,
 * BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
 * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
 * LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE
 * OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
 * ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
 */
using System;
using System.Security.Cryptography;
using System.Text;

/*
 * Deutsche Post DHL Group - Internetmarke.
 *
 * Example of an MD5 signing / hashing algorithm, to generate the hashes to
 * be enclosed in the Internetmarke interface SOAP headers.
 */

```


Kurzanleitung

```
*
* You may freely use this snippet in your own programs for Internetmarke,
* according to the license regulations.
*/
namespace DPDHL
{
    /// <summary>
    /// Example snippet for the MD5 hash calculation for SOAP headers.
    /// </summary>
    class MD5Sample
    {
        // Length of the hash string for the SOAP header
        const int HASH_STRING_LENGTH = 8;
        const string SEPARATOR = "::";

        /// <summary>
        /// Starting method for the example program snippet.
        /// </summary>
        /// <param name="args">Command line parameters, not used here.</param>
        static void Main(string[] args)
        {
            MD5Sample md5Sample = new DPDHL.MD5Sample();

            const string partnerId = "DPPAR";
            const string signatureKey = "Das_Ist_der_neue_Key";

            string partnerSignature = md5Sample.sign(partnerId, signatureKey);

            Console.WriteLine("Partner Signature: " + partnerSignature);
        }

        /// <summary>
        /// Generate an Internetmarke SOAP header signature, based on the given attributes.
        /// The signature (4 bytes hash in Base16 format) is returned in a string.
        /// </summary>
        /// <param name="partnerId">The PartnerID assigned to you.</param>
        /// <param name="signatureKey">The signature key assigned to you.</param>
        /// <returns>The signature, as 8 char hash in Base16 encoding.</returns>
        public string sign(string partnerId, string signatureKey)
        {
            return sign(partnerId, "1", signatureKey);
        }

        /// <summary>
        /// Generate an Internetmarke SOAP header signature, based on the given attributes.
        /// The signature (4 bytes hash in Base16 format) is returned in a string.
        /// </summary>
        /// <param name="partnerId">The PartnerID assigned to you.</param>
        /// <param name="keyPhase">The key phase to use.</param>
        /// <param name="signatureKey">The signature key assigned to you.</param>
        /// <returns>The signature, as 8 char hash in Base16 encoding.</returns>
    }
}
```

Kurzanleitung

```
public string sign(string partnerId, string keyPhase, string signatureKey)
{
    DateTime timestamp = DateTime.Now;
    // Day - Month - Year - Hour - Min - Sec
    string requestTimestamp = timestamp.ToString("ddMMyyyy-HHmss");
    // For debugging, set a fixed date time to verify hash
    requestTimestamp = "28012014-142729";

    string temp = partnerId + SEPARATOR +
        requestTimestamp + SEPARATOR +
        keyPhase + SEPARATOR +
        signatureKey;

    // A bit of debugging, to see what is going on...
    Console.WriteLine("Input string for MD5 hash: " + temp);

    try
    {
        MD5 md5 = new MD5CryptoServiceProvider();
        byte[] bsig = Encoding.ASCII.GetBytes(temp);

        // Do the MD5 hashing using a built-in lib
        byte[] md5ByteHash = md5.ComputeHash(bsig);

        // Now convert to string, as 4 Base16 encoded bytes (8 char)
        StringBuilder md5HashString = new StringBuilder(32);
        foreach (byte md5Byte in md5ByteHash)
        {
            md5HashString.Append(md5Byte.ToString("x2"));
        }

        return md5HashString.ToString().Substring(0, HASH_STRING_LENGTH);
    }
    catch (ArgumentException ex)
    {
        // Add logging or error handling here...
        Console.WriteLine("Can't generate signature! " + ex);
    }

    // Exit if hash could not be calculated.
    return null;
}
}
```

Deutsche Post AG	INTERNETMARKE Webservice 1C4A für Drittanbieter	Seite 19 / 19
Kurzanleitung		

5.3 Glossar

Begriff	Beschreibung
1C4A	OneClickForApplikation, Webservice der Deutsche Post AG zur Einbindung der INTERNETMARKE-Frankierung
Drittanbieter	Ein Drittanbieter bezeichnet im engeren Sinne die Marktplätze und Partner, welche die Applikation INTERNETMARKE über den hier beschriebenen Webservice eingebunden haben.
INTERNETMARKE	Die INTERNETMARKE ist eine Applikation der Deutsche Post AG zum Kauf von Internetmarken.
Internetmarken	Der Begriff Internetmarke wird im Kontext von Partnersystemen verwendet. Internetmarken sind Frankiervermerke, die vom System INTERNETMARKE erstellt wurden.
Kunde	Ein Benutzer, der Frankiervermerke kaufen möchte.
Marke	In diesem Dokument ist die Marke ein Synonym für eine Internetmarke.
Portokasse	Die PORTOKASSE ist ein Prepaid-Wallet mit dem zuvor aufgeladene Geldbeträge genutzt werden können, um INTERNETMARKEN oder andere Produkte der Deutsche Post AG zu kaufen
Versandliste	Dokumentation der Sendungen mit Zusatzleistungen National und International für den Absender