Amadeus Web Services

Security Credentials & Soap 4.0 Headers for USA Travel Services, LLC

1ASIWOTANA4

**Index**

Soap 4.0 Handbook 3

Security Credentials 3

Nonce 3

Current Date and Time 4

Hashed Password 4

Soap 4.0 with Stateless Header 5

Soap 4.0 with Stateful Header 6

First Request 6

Following Request 6

Final Request 7

# Soap 4.0 Handbook

|  |  |
| --- | --- |
| Soap 4.0 Handbook – Extranet  Technical doc  WBS  Framework documentation  Framework Documentation  SOAP header 4.0: Embedded Security  Handbook |  |
|  |  |

#### Extranet URL

https://webservices.amadeus.com/extranet/kdbViewDocument.do?externalId=wikidoc\_web\_services\_embedded\_security\_implementation\_guide&docStatus=Published&mpId=fla\_\_1\_\_technical

# Security Credentials

**OTA4**:

WSAP - 1ASIWOTANA4

End Point - <https://nodeD1.test.webservices.amadeus.com/1ASIWOTANA4>

Username – WSNA4OTA

Clear Password - uYTzW

POS\_Type = 1

PseudoCityCode – MIA1S38BL

AgentDutyCode – SU

RequestorType – U

### Nonce

Nonce is a random number used to encode the password and avoid replicate attacks

**Nonce** in Base 64 = 'PZgFvh5439plJpKpIyf5ucmXhNU='

#### Script

import org.apache.ws.security.util.Base64;

import java.security.SecureRandom;

def generate\_nonce() {

def random = SecureRandom.getInstance("SHA1PRNG");

random.setSeed(System.currentTimeMillis());

def nonceValue = new byte[16];

random.nextBytes(nonceValue);

return Base64.encode(nonceValue);

}

return generate\_nonce()

### Current Date and Time

CurrentGMTTimestamp is a current date and time ('YYYY-MM-DDTHH:MM:SSZ') in GMT

**CurrentGMTTimestamp** = ‘2013-01-11T09:41:03Z’

#### Script

import java.text.SimpleDateFormat;

def generate\_created() {

def format = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'");

def cal = Calendar.instance;

format.setTimeZone(TimeZone.getTimeZone("Zulu"));

return format.format(cal.time);

}

return generate\_created()

### Hashed Password

HashedPassword is a combination of Nonce, current date & time in GMT and Clear Password

HashedPassword = Base64 (SHA-1 (Nonce + CurrentGMTTimestamp +SHA-1(<clearpassword>)))

**HashedPassword** = 'PZgFvh5439plJpKpIyf5ucmXhNU=’ + ‘2013-01-11T09:41:03Z’ + ‘ xvmMB’

#### Script

import org.apache.ws.security.util.Base64;

import java.security.MessageDigest;

import java.security.SecureRandom;

import java.text.SimpleDateFormat;

def password\_encrypt(algorithm) {

def sha = MessageDigest.getInstance(algorithm);

sha.reset();

def password = testRunner.testCase.getPropertyValue("OverridePassword");

if (password == null || password.length() == 0) {

if (testRunner.testCase.testSuite.project.hasProperty("password")) {

password = testRunner.testCase.testSuite.project.getPropertyValue("password");

def b1 = password.getBytes("UTF-8");

sha.update(b1);

}

else {

testRunner.fail("LSS Password is not defined at project level")

}

}

return sha.digest();

}

def password\_digest(password, nonce, created) {

def passwdDigest = null;

def b1 = nonce != null ? Base64.decode(nonce) : new byte[0];

def b2 = created != null ? created.getBytes("UTF-8") : new byte[0];

def b3 = password; def b4 = new byte[b1.length + b2.length + b3.length];

def offset = 0;

System.arraycopy(b1, 0, b4, offset, b1.length);

offset += b1.length;

System.arraycopy(b2, 0, b4, offset, b2.length);

offset += b2.length;

System.arraycopy(b3, 0, b4, offset, b3.length);

def sha = MessageDigest.getInstance("SHA-1");

sha.reset();

sha.update(b4);

passwdDigest = Base64.encode(sha.digest());

return passwdDigest;

}

return password\_digest(password\_encrypt("SHA-1"),

testRunner.testCase.getPropertyValue("Nonce"),

testRunner.testCase.getPropertyValue("Created"));

## Soap 4.0 with Stateless Header

<soapenv:Header>

<add:MessageID xmlns:add="http://www.w3.org/2005/08/addressing">33f4b248-0d57-b439-52c6-aa85f7cd15db</add:MessageID>

<add:Action xmlns:add="http://www.w3.org/2005/08/addressing">http://webservices.amadeus.com/**FMPTBQ\_13\_1\_1A**</add:Action>

<add:To xmlns:add="http://www.w3.org/2005/08/addressing">https://nodeD1.test.webservices.amadeus.com/**1ASIWOTANA4**</add:To>

<link:TransactionFlowLink xmlns:link="http://wsdl.amadeus.com/2010/06/ws/Link\_v1"/>

<oas:Security xmlns:oas="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">

<oas:UsernameToken oas1:Id="UsernameToken-1" xmlns:oas1="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

<oas:Username>**WSNA4OTA**</oas:Username>

<oas:Nonce EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">**Nonce**</oas:Nonce>

<oas:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordDigest">**HashedPassword**</oas:Password>

<oas1:Created>**CurrentGMTDateAndTime**</oas1:Created>

</oas:UsernameToken>

</oas:Security>

<AMA\_SecurityHostedUser xmlns="http://xml.amadeus.com/2010/06/Security\_v1">

<UserID AgentDutyCode="SU" RequestorType="U" PseudoCityCode="**MIA1S38BL**" POS\_Type="1"/>

</AMA\_SecurityHostedUser>

</soapenv:Header>

## Soap 4.0 with Stateful Header

### First Request

<soapenv:Header>

<add:MessageID xmlns:add="http://www.w3.org/2005/08/addressing">69582a2c-e955-eebb-f51d-d54ed1aa594b</add:MessageID>

<add:Action xmlns:add="http://www.w3.org/2005/08/addressing">http://webservices.amadeus.com/**FMPTBQ\_14\_3\_IA**</add:Action>

<add:To xmlns:add="http://www.w3.org/2005/08/addressing">https://nodeD1.test.webservices.amadeus.com/**1ASIWOTANA4**</add:To>

<link:TransactionFlowLink xmlns:link="http://wsdl.amadeus.com/2010/06/ws/Link\_v1"/>

<oas:Security xmlns:oas="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">

<oas:UsernameToken oas1:Id="UsernameToken-1" xmlns:oas1="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">

<oas:Username**>WSNA4OTA**</oas:Username>

<oas:Nonce EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary"> **Nonce**</oas:Nonce>

<oas:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordDigest"> **HashedPassword**</oas:Password>

<oas1:Created> **CurrentGMTDateAndTime**</oas1:Created>

</oas:UsernameToken>

</oas:Security>

<AMA\_SecurityHostedUser xmlns="http://xml.amadeus.com/2010/06/Security\_v1">

<UserID AgentDutyCode="SU" RequestorType="U" PseudoCityCode="**MIA1S38BL**" POS\_Type="1"/>

</AMA\_SecurityHostedUser>

<awsse:Session **TransactionStatusCode="Start"** xmlns:awsse="http://xml.amadeus.com/2010/06/Session\_v3"/>

</soapenv:Header>

### Following Request

<soapenv:Header>

<awsse:Session **TransactionStatusCode="InSeries"** xmlns:awsse="http://xml.amadeus.com/2010/06/Session\_v3">

<awsse:SessionId>**${SessionId}**</awsse:SessionId>

<awsse:SequenceNumber>**${SequenceNumber}**</awsse:SequenceNumber>

<awsse:SecurityToken>**${SecurityToken}**</awsse:SecurityToken>

</awsse:Session>

<add:MessageID xmlns:add="http://www.w3.org/2005/08/addressing">69582a2c-e955-eebb-f51d-d54ed1aa594b</add:MessageID>

<add:Action xmlns:add="http://www.w3.org/2005/08/addressing">http://webservices.amadeus.com/**ITAREQ\_05\_2\_1A**</add:Action>

<add:To xmlns:add="http://www.w3.org/2005/08/addressing">https://nodeD1.test.webservices.amadeus.com/**1ASIWOTANA4**</add:To>

</soapenv:Header>

### Final Request

<soapenv:Header>

<awsse:Session **TransactionStatusCode="End"** xmlns:awsse="http://xml.amadeus.com/2010/06/Session\_v3">

<awsse:SessionId>**${SessionId}**</awsse:SessionId>

<awsse:SequenceNumber>**${SequenceNumber}**</awsse:SequenceNumber>

<awsse:SecurityToken>**${SecurityToken}**</awsse:SecurityToken>

</awsse:Session>

<add:MessageID xmlns:add="http://www.w3.org/2005/08/addressing">69582a2c-e955-eebb-f51d-d54ed1aa594b</add:MessageID>

<add:Action xmlns:add="http://www.w3.org/2005/08/addressing">http://webservices.amadeus.com/ **VLSSOQ\_04\_1\_1A**</add:Action>

<add:To xmlns:add="http://www.w3.org/2005/08/addressing">https://nodeD1.test.webservices.amadeus.com/**1ASIWOTANA4**</add:To>

</soapenv:Header>