



Service Component Architecture HTTP Binding Specification Version 1.1

Working Draft 01

22nd October 2008

Specification URIs:

This Version:

- <http://docs.oasis-open.org/sca-bindings/sca-binding-http-1.1-spec-wd01.html>
- <http://docs.oasis-open.org/sca-bindings/sca-binding-http-1.1-spec-wd01.doc>
- <http://docs.oasis-open.org/sca-bindings/sca-binding-http-1.1-spec-wd01.pdf> (Authoritative)

Previous Version:

Latest Version:

- <http://docs.oasis-open.org/sca-bindings/sca-binding-http-1.1-spec.html>
- <http://docs.oasis-open.org/sca-bindings/sca-binding-http-1.1-spec.doc>
- <http://docs.oasis-open.org/sca-bindings/sca-binding-http-1.1-spec.pdf> (Authoritative)

Latest Approved Version:

Technical Committee:

OASIS Service Component Architecture / Bindings (SCA-Bindings) TC

Chair(s):

Simon Holdsworth, IBM

Editor(s):

Simon Holdsworth, IBM
Khanderao Kand, Oracle
Anish Karmarkar, Oracle
Sanjay Patil, SAP
Piotr Przybylski, IBM

Related work:

This specification is related to:

- Service Component Architecture Assembly Model Specification Version 1.1
- Service Component Architecture Policy Framework Specification Version 1.1

Declared XML Namespace(s):

<http://docs.oasis-open.org/ns/opencsa/sca/200712>

Abstract:

This document defines the concept and behavior of an HTTP-based binding.

The binding specified in this document applies to an SCA composite's services and references. The binding is especially well suited for use by services and references of composites that are directly deployed, as opposed to composites that are used as implementations of higher-level components. Services and references of deployed composites become system-level services and references, which are intended to be used by non-SCA clients.

The HTTP binding provides HTTP-specific details of the connection to the required HTTP endpoint and controls formation of HTTP protocol messages. It supports the use of HTTP(s) type destinations.

Status:

This document was last revised or approved by the OASIS Service Component Architecture / Bindings (SCA-Bindings) TC on the above date. The level of approval is also listed above. Check the "Latest Version" or "Latest Approved Version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at <http://www.oasis-open.org/committees/sca-bindings/>.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (<http://www.oasis-open.org/committees/sca-bindings/ipr.php>).

The non-normative errata page for this specification is located at <http://www.oasis-open.org/committees/sca-bindings/>.

Notices

Copyright © OASIS® 2008. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The names "OASIS", [insert specific trademarked names and abbreviations here] are trademarks of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

Table of Contents

1	Introduction	5
1.1	Terminology	5
1.2	Normative References	5
1.3	Non-Normative References	5
2	Operation Selection and Data Binding	Error! Bookmark not defined.
3	HTTP Binding Schema	6
4	Default Operation Selection and Data Binding behavior	9
4.1	Default Operation Selection	9
4.2	Default Data Binding	9
5	Policy	Error! Bookmark not defined.
6	Examples	11
6.1	Minimal Binding Example	Error! Bookmark not defined.
6.2	URI Binding Example	11
6.3	Policy Set Example	Error! Bookmark not defined.
7	Conformance	12
A.	HTTP Binding Schema	13
B.	Acknowledgements	15
C.	Non-Normative Text	16
D.	Revision History	17

1 Introduction

This document defines the concept and behavior of the HTTP [HTTP] binding. The binding specified in this document applies to an SCA composite's services and references. The binding is especially well suited for use by services and references of composites that are directly deployed, as opposed to composites that are used as implementations of higher-level components. Services and references of deployed composites become system-level services and references, which are intended to be used by non-SCA clients. The HTTP binding provides HTTP-specific details of the connection to the required HTTP endpoint and controls formation of HTTP protocol messages. It supports the use of HTTP(s) type destinations.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119.

This specification uses predefined namespace prefixes throughout; they are given in the following list. Note that the choice of any namespace prefix is arbitrary and not semantically significant.

Table 1-1 Prefixes and Namespaces used in this specification

Prefix	Namespace	Notes
xs	"http://www.w3.org/2001/XMLSchema"	Defined by XML Schema 1.0 specification
sca	"http://docs.oasis-open.org/ns/opencsa/sca/200712"	Defined by the SCA specifications

1.2 Normative References

- [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
- [HTTP] HTTP specification (1.0, 1.1?) TBC
<http://www.w3.org/Protocols/rfc2616/rfc2616.html>
- [WSDL] E. Christensen et al, *Web Service Description Language (WSDL) 1.1*, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>, W3C Note, March 15 2001.
R. Chinnici et al, *Web Service Description Language (WSDL) Version 2.0 Part 1: Core Language*, <http://www.w3.org/TR/2007/REC-wsdl20-20070626/>, W3C Recommendation, June 26 2007.

1.3 Non-Normative References

TBD TBD

29

2 HTTP Binding Schema

30

The HTTP binding element is defined by the following schema.

31

```

<binding.http uri="xs:anyURI"?
  readTimeout="int"?
  numberOfRetries="int"?
  methods="list of string"?
  ...>

  <wireFormat/>?

  <operationSelector/>?

  <response>
    <wireFormat>?
  </response>?

  <pingableMethod name="string" statusCode="int">*

  <proxySettings host="string"?
    port="int"?
    type="http or https"?
    nonProxyHost="string"?>
    <credentials alias="string"/>?
  </proxySettings>*

  <sslSettings alias="string"/>?

  <authentication type="basic or digest or ..."?>
    <credentials alias="string"/>?
  </authentication>?

  <headers HTTPVersion="1.0 or 1.1"?
    HTTPContentEncoding="string"?
    HTTPTransferEncoding="string"?
    HTTPMediaType="string"?
    HTTPCharset="string"?
    HTTPMethod="string"?
    ...>
    <property name="NMTOKEN" type="NMTOKEN">*
  </headers>?

  <operationProperties name="string"
    selectedOperation="string"?
    methods="list of string"?
    contextPath="string"?>
    <property name="NMTOKEN" type="NMTOKEN">*
    <headers HTTPVersion="1.0 or 1.1"?
      HTTPContentEncoding="string"?
      HTTPTransferEncoding="string"?
      HTTPMediaType="string"?
      HTTPCharset="string"?
      HTTPMethod="string"?
      ...>
      <property name="NMTOKEN" type="NMTOKEN">*
    </headers>?
  </operationProperties>*
</binding.http>

```

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

Comment [sajh1]: Given that many of the elements/attributes are specific to use for a service or reference, it may make sense to split this into two elements, although there's no precedent for that in other bindings.

87 The **binding.http** element has the following attributes:

- 88 • **/binding.http** – This is the generic HTTP binding type. The type is extensible so that HTTP binding
89 implementers can add additional attributes and elements although such extensions are not
90 guaranteed to be portable across runtimes.
- 91 • **/binding.http@uri** – (from binding) for a reference, identifies the URL of the target service. For a
92 service, identifies the base context path.
- 93 • **/binding.http@readTimeout** – The amount of time that should be allowed for a response to be
94 received after sending a request.
- 95 • **/binding.http@numberOfRetries** – the number of times the request should be retried.
- 96 • **/binding.http@methods** – for a service, the list of methods supported for all operations, unless
97 overridden for a specific operation.
- 98 • **/binding.http/wireFormat** - identifies the wire format used by requests and responses sent or
99 received by this binding.
- 100 • **/binding.http/operationSelector** – identifies the operation selector used when receiving requests for
101 a service. If specified for a reference this provides the default operation selector for callbacks if not
102 specified via a callback service element.
- 103 • **/binding.http/response** – defines the resources used for handling response messages (receiving
104 responses for a reference, and sending responses from a service).
- 105 • **/binding.http/response/wireFormat** – identifies the wire format used by responses sent or received
106 by this binding. This value overrides the **wireFormat** specified at the binding level.
- 107 • **/binding.http/pingableMethod** – for a service, identifies a method that is pingable.
- 108 • **/binding.http/pingableMethod/@name** – the name of the pingable method.
- 109 • **/binding.http/pingableMethod/@statusCode** – the HTTP status code that is returned.
- 110 • **/binding.http/proxySettings** – for a reference, identifies the settings of an HTTP proxy to be used.
- 111 • **/binding.http/proxySettings/@host** – the host name of the proxy.
- 112 • **/binding.http/proxySettings/@port** – the port number of the proxy.
- 113 • **/binding.http/proxySettings/@type** – the type of the proxy, http or https.
- 114 • **/binding.http/proxySettings/@nonProxyHost** – Host name for which the proxy should not be used.
- 115 • **/binding.http/proxySettings/credentials** – Credentials to be used with the proxy.
- 116 • **/binding.http/proxySettings/credentials/@alias** – The alias of the credential settings to be used.
- 117 • **/binding.http/sslSettings** – for a reference, identifies the SSL settings to use to connect to the target
118 service.
- 119 • **/binding.http/sslSettings/@alias** – The alias of the SSL settings to be used.
- 120 • **/binding.http/authentication** – for a reference, identifies the authentication to use to connect to the
121 target service.
- 122 • **/binding.http/authentication/@type** – The type of authentication to use.
- 123 • **/binding.http/authentication/credentials** – Credentials to be used to authenticate to the target
124 service.
- 125 • **/binding.http/authentication/credentials/@alias** – The alias of the credential settings to be used.
- 126 • **/binding.http/headers** – this element allows HTTP headers to be set to the given values for all
127 operations. These values apply to requests from a reference and responses from a service.
- 128 • **/binding.http/headers/@HTTPVersion, @HTTPMethod, @HTTPContentEncoding,**
129 **@HTTPTransferEncoding, @HTTPMediaType, @HTTPCharSet** – specifies the value to use for the
130 given HTTP header property. **@HTTPVersion** and **@HTTPMethod** only apply to requests sent from
131 references.
- 132 • **/binding.http/headers/property** – specifies the value to use for the named HTTP header.

Comment [sajh2]: Do we want to say anything at all about the format of the URI wrt. parameterization?

Comment [SAJH3]: This is somewhat JEE-centric; perhaps these should be removed and replaced by intents?

- 133 • ***/binding.http.operationProperties*** – specifies various properties that are specific to the processing
134 of a particular operation.
- 135 • ***/binding.http.operationProperties/@name*** – The name of the operation in the interface.
- 136 • ***/binding.http.operationProperties/@selectedOperation*** – The value generated by the
137 ***operationSelector*** that corresponds to the operation in the service or reference interface identified
138 by the ***operationProperties/@name*** attribute. If this attribute is omitted then the value defaults to
139 the value of the ***operationProperties/@name*** attribute. The value of this attribute MUST be unique
140 across the containing ***binding.http*** element.
- 141 • ***/binding.http.operationProperties/@methods*** – for a service, the list of methods supported for this
142 operation. Overrides the value specified for the binding.
- 143 • ***/binding.http.operationProperties/@contextPath*** – for a service, context path for this specific
144 operation.
- 145 • ***/binding.http.operationProperties/property*** – specifies properties specific to this operation. These
146 properties are intended to be used to parameterize the ***wireFormat*** identified for the binding for a
147 particular operation. The SCA runtime SHOULD make the ***operationProperties*** element
148 corresponding to the ***selectedOperation*** available to the ***wireFormat*** implementation.
- 149 • ***/binding.http.operationProperties/headers*** – this element allows HTTP headers to be set to the
150 given values for all operations. These values apply to requests from a reference and responses from
151 a service.
- 152 • ***/binding.http.operationProperties/headers/@HTTPVersion, @HTTPMethod,***
153 ***@HTTPContentEncoding, @HTTPTransferEncoding, @HTTPMediaType, @HTTPCharSet*** –
154 specifies the value to use for the given HTTP header property. ***@HTTPVersion*** and ***@HTTPMethod***
155 only apply to requests sent from references.
- 156 • ***/binding.http.operationProperties/headers/property*** – specifies the value to use for the named
157 HTTP header.
- 158 • ***/binding.http/@{any}*** - this is an extensibility mechanism to allow extensibility via attributes.
- 159 • ***/binding.http/any*** – this is an extensibility mechanism to allow extensibility via elements.

160 2.1 Intents listed in the bindingType

161 This specification places no requirements on the intents that must be listed as either ***@alwaysProvides***
162 or ***@mayProvides*** in the ***bindingType*** for ***binding.http***.

163 2.2 Intents and binding configuration

164 The SCA runtime MUST raise an error if the HTTP binding is configured with a policy intent(s) that
165 conflicts with a binding instance's configuration.

166 3 Default Operation Selection and Data Binding 167 behavior

168 In general HTTP-based applications deal with URL paths and HTTP methods. There is not always a
169 direct relationship between these and an operation defined in a WSDL portType [WSDL]. Messages
170 have a content type which corresponds in some way to the schema of an input or output message of an
171 operation in the interface of a service or reference, however additional information is required in order for
172 an SCA runtime to know how to identify the operation and understand the format of messages.

173 The process of identifying the operation to be invoked is operation selection; the information that
174 describes the contents of messages is a wire format. The binding element as described in the SCA
175 Assembly specification [ref needed] provides the means to identify specific operation selection via the
176 **operationSelector** element and the format of messages received and to be sent using the **wireFormat**
177 element. The HTTP binding defines a default operation selector and wire format and the corresponding
178 **operationSelector** and **wireFormat** elements; SCA providers may provide additional such elements.

179 No standard means is provided for linking the **wireFormat** or **operationSelector** elements with the
180 runtime components that implement their behavior.

181 This section describes the default operation selector and wire format for an HTTP binding.

182 3.1 Default Operation Selector

183 When receiving a request at a service the selected operation name is determined as follows:

- 184 • If there is only one operation on the service's interface, then that operation is assumed as the
185 selected operation name.
- 186 • Otherwise, if the HTTP header "**scaOperationName**" is present, then its value is used as the
187 selected operation name.
- 188 • Otherwise, a combination of the URL's context path and HTTP method is used as the selected
189 operation name, i.e. HTTP Method + "@" + URL context path.

190 The selected operation name is then be mapped to an operation in the service's interface via a matching
191 **operationProperties** element in the HTTP binding. If there is no matching element, the operation name
192 is assumed to be the same as the selected operation name.

193 The use of this operation selector can be explicitly specified in a **binding.http** using the
194 **<operationSelector.httpdefault>** element; if no **operationSelector** element is specified then SCA
195 runtime MUST use this as the default.

196 An SCA runtime **SHOULD** provide the means for identifying alternative operation selectors via additional
197 **<operationSelector>** elements.

198 3.2 Default Wire Format

199 The default wire format behavior maps between a message on the wire and the object(s) expected by the
200 component implementation.

201 The message body is mapped to the parameters or return value of the target operation as follows:

- 202 • If the operation has a single parameter or return value that is a byte array, then the whole message is
203 passed as is, and MUST have a content type of "application/octet-stream".
- 204 • Otherwise, the message content type MUST be "text/xml".
- 205 • If there is a single parameter, or for the return value, the text XML payload is the XML serialization of
206 that parameter according to the WSDL schema for the message.
- 207 • If there are multiple parameters, then they are encoded in XML using the document wrapped style,
208 according to the WSDL schema for the message.

Comment [sajh4]: Should we support MIME messages in the default wire format?

Also can the default wire format make use of additional content types?

Should we say anything about URL parameterization here?

209 • When sending request messages, if there is a single parameter and the interface includes more than
210 one operation, the SCA runtime MUST set the HTTP header "**scaOperationName**" to the name of
211 the operation being invoked.

212 The use of this wire format can be explicitly specified in a **binding.http** using the
213 **<wireFormat.httpdefault>** element; if no **wireFormat** element is specified then SCA runtimes MUST use
214 this as the default.

215 An SCA runtime SHOULD provide the means for identifying alternative wire formats to support any other
216 type of HTTP message via additional **<wireFormat>** elements.

217

218 4 Examples

219 The following snippets show the sca.composite file for the MyValueComposite file containing the service
220 element for the MyValueService and a reference element for the StockQuoteService. Both the service
221 and the reference use a HTTP binding.

222 4.1 URI Binding Example

223 The following example shows the HTTP binding using the URI attribute for a service to specify the context
224 path, and for a reference to specify the location of the target service:

```
225 <?xml version="1.0" encoding="ASCII"?>  
226 <composite xmlns="http://www.oesa.org/xmlns/sca/1.0"  
227           name="MyValueComposite">  
228  
229     <service name="MyValueService">  
230       <interface.java interface="services.myvalue.MyValueService"/>  
231       <binding.http uri="/myValueService"/>  
232     </service>  
233  
234     <reference name="StockQuoteService">  
235       <interface.java interface="services.stockquote.StockQuoteService"/>  
236       <binding.http uri="http://www.example.org/StockQuoteService"/>  
237     </reference>  
238 </composite>
```

239

240 TBD: Need more examples of using various binding elements/attributes

241 5 Conformance

242 Any SCA runtime that claims to support this binding MUST abide by the requirements of this specification.

243 The XML schema available at the namespace URI, defined by this specification, is considered to be
244 authoritative and takes precedence over the XML Schema defined in the appendix of this document.

245 Within this specification, the following conformance targets are used:

- 246 • XML document elements and attributes, including binding.http and its children, and bindingType
- 247 • The SCA runtime – this refers to the implementation that provides the functionality to support the SCA
248 specifications, including that specific to the HTTP binding as well as other SCA capabilities
- 249 • WSDL documents

A. HTTP Binding Schema

251 Note: needs to be validated.

```

252 <?xml version="1.0" encoding="UTF-8"?>
253 <!-- (c) Copyright OASIS 2008 -->
254 <schema xmlns="http://www.w3.org/2001/XMLSchema"
255       targetNamespace="http://docs.oasis-open.org/ns/opencsa/sca/200712"
256       xmlns:sca="http://docs.oasis-open.org/ns/opencsa/sca/200712"
257       elementFormDefault="qualified">
258
259   <include schemaLocation="sca-core.xsd"/>
260
261   <complexType name="HTTPBinding">
262     <complexContent>
263       <extension base="sca:Binding">
264         <sequence>
265           <element name="response" type="sca:HTTPResponse" minOccurs="0"/>
266           <element name="pingableMethod" type="sca:HTTPPingableMethod"
267                 minOccurs="0"/>
268           <element name="proxySettings" type="sca:HTTPProxySettings"
269                 minOccurs="0"/>
270           <element name="sslSettings" type="sca:HTTPSSLSettings"
271                 minOccurs="0"/>
272           <element name="authentication" type="sca:HTTPAuthentication"
273                 minOccurs="0"/>
274           <element name="headers" type="sca:HTTPHeaders" minOccurs="0"/>
275           <element name="operationProperties"
276                 type="sca:HTTPOperationProperties"
277                 minOccurs="0" maxOccurs="unbounded"/>
278           <any namespace="##other" processContents="lax"
279                 minOccurs="0" maxOccurs="unbounded"/>
280         </sequence>
281         <attribute name="readTimeout" type="int"/>
282         <attribute name="numberOfRetries" type="int"/>
283         <attribute name="methods" type="string"/>
284         <anyAttribute/>
285       </extension>
286     </complexContent>
287   </complexType>
288
289   <complexType name="HTTPResponse">
290     <sequence>
291       <element name="wireFormat" type="sca:WireFormat" minOccurs="0"/>
292     </sequence>
293   </complexType>
294
295   <complexType name="HTTPProxySettings">
296     <sequence>
297       <element name="credentials" minOccurs="0">
298         <attribute name="alias" type="string"/>
299       </element>
300       <attribute name="host" type="string"/>
301       <attribute name="port" type="int"/>
302       <attribute name="type">
303         <simpleType>
304           <restriction base="string">
305             <enumeration value="http"/>
306             <enumeration value="https"/>
307           </restriction>
308         </simpleType>
309       </attribute>

```

```

310     <attribute name="nonProxyHost" type="string"/>
311   </sequence>
312 </complexType>
313
314 <complexType name="HTTPSSLSettings">
315   <attribute name="alias" type="string"/>
316 </complexType>
317
318 <complexType name="HTTPAuthentication">
319   <sequence>
320     <element name="credentials" minOccurs="0">
321       <attribute name="alias" type="string"/>
322     </element>
323     <attribute name="type" type="string"/>
324   </sequence>
325 </complexType>
326
327 <simpleType name="HTTPVersion">
328   <restriction base="string">
329     <enumeration value="1.1"/>
330     <enumeration value="1.0"/>
331   </restriction>
332 </simpleType>
333
334 <complexType name="HTTPHeaders">
335   <sequence>
336     <element name="property" type="string"
337       minOccurs="0" maxOccurs="unbounded"/>
338   </sequence>
339   <attribute name="HTTPVersion" type="HTTPVersion" default="1.1"/>
340   <attribute name="HTTPContentEncoding" type="string"/>
341   <attribute name="HTTPTransferEncoding" type="string"/>
342   <attribute name="HTTPMediaType" type="int"/>
343   <attribute name="HTTPCharset" type="string"/>
344   <attribute name="HTTPMethod" type="string"/>
345 </complexType>
346
347 <complexType name="HTTPOperationProperties">
348   <sequence>
349     <element name="property" type="string"
350       minOccurs="0" maxOccurs="unbounded"/>
351     <element name="headers" type="sca:HTTPHeaders"/>
352   </sequence>
353   <attribute name="name" type="string" use="required"/>
354   <attribute name="selectedOperation" type="string"/>
355 </complexType>
356
357 <element name="binding.http" type="sca:HTTPBinding"
358   substitutionGroup="sca:binding"/>
359 </schema>

```

360

361 **B. Acknowledgements**

362 The following individuals have participated in the creation of this specification and are gratefully
363 acknowledged:

364 **Participants:**

365 [Participant Name, Affiliation | Individual Member]

366 [Participant Name, Affiliation | Individual Member]

367 **C. Non-Normative Text**

368 **D. Revision History**

369 [optional; should not be included in OASIS Standards]

370

Revision	Date	Editor	Changes Made
wd01	2008-10-22	Simon Holdsworth	Initial draft

371