



eSIM Application Note

SOAP binding in SGP.02

28 May 2021



eSIM Application Note, 28 May 2021

SOAP binding in SGP.02.

Overview

The purpose of this Application Note is to provide clarification for SOAP binding mechanisms in accordance with SGP.02, to ensure interoperability and to simplify integration between servers, i.e. function requesters and function providers.

SOAP binding

SGP.02 v3.2 and v4.2 Annex A and Annex B describe the binding of RPS messages, composed of RPSHeader and RPSBody, into SOAP messages. Both RPS Header and RPS Body information is used to create the SOAP header and the SOAP body according to the rules defined in the specification.

RPS fields in RPSHeader

The RPS Header contains mainly information for the transport of the message, and is composed of the following fields: `rps3:SenderEntity`, `rps3:SenderName`, `rps3:ReceiverEntity`, `rps3:ResponseEndpoint`, `rps3:ContextId`, `rps3:TransactionId`, `rps3:MessageId`, `rps3:MessageType`, `rps3:RelatesTo`, `rps3:MessageDate`, `rps3:MnoId`, `rps3:ProfileType`.

NOTE: In this document, `rps` fields are named as defined in SGP.02 v4.2 i.e. "`rps3:<name>`", whereas in SGP.02 v3.2 they are defined with "`rps:<name>`".

It should be noted that `rps3:ResponseEndpoint`, `rps3:MessageId`, `rps3:RelatesTo`, are defined with the type `xs:anyURI` according to Extensible Markup Language (XML) 1.0, W3C Recommendation, referring the RFC 2396.

NOTE: RFC 3986 document obsoletes RFC 2396 for the definition of the URI generic syntax.

WSA fields

The SOAP header is defined according to the `wsa` namespace -WS-Addressing message elements. Fields used for binding are: `wsa:From`, `wsa:To`, `wsa:ReplyTo`, `wsa:Action`, `wsa:FaultTo`, `wsa:MessageId`, `wsaRelatesTo`.

Many fields are defined, according to Web Services Addressing 1.0, Core <http://www.w3.org/TR/ws-addr-core/> with the type `xs:anyURI`, e.g. `wsa:To`, `wsa:From`, `wsa:Address`, `wsa:MessageId` and `wsaRelatesTo`, but are specified to convey absolute IRIs as defined in RFC 3987.



SOAP Header

SGP.02 v3.2 and v4.x Annex B describe how these `wsa` fields are created from `rps3` fields, by providing key-value pairs separated by a question mark “?”.

The following example, extracted from SGP.02 illustrates the creation of such `wsa` fields from the following content of `rps3` RPS header:

```
<rps3:MessageId>//MySenderDomain/123</rps3:MessageId>
<rps3:TransactionId>MyTansactionID1</rps3:TransactionId>
<rps3:ContextId>MyContextID1</rps3:ContextId>
<rps3:MessageDate>2013-04-18T09:45:00Z</rps3:MessageDate>
<rps3:ProfileType>3G_16K</rps3:ProfileType>
```

Would be mapped into:

```
<wsa:MessageID>//MySenderDomain/123?TransactionId=MyTansactionID1?ContextId=MyContextID1?MessageDate=2013-04-18T09:45:00Z?ProfileType=3G_16K</wsa:MessageID>
```

Specific points of interest

The creation of some `wsa` fields in the SOAP header uses different `rps3` fields. In some cases, e.g. for the creation of `wsa:MessageId`, these fields are of the type `xs:anyURI` (i.e. `rps3:MessageId`). Therefore, there is an encapsulation of `xs:anyURI` into absolute IRIs encoded as `xs:anyURI`.

Different composition methods may be used to realize this encapsulation, resulting in URIs that are legitimate `xs:anyURI` values but ambiguous or invalid as WS-Addressing fields or SGP.02 fields. Therefore, function requesters and function providers should have identical rules to extract the needed information from the resulting `xs:anyURI` fields.

For example, the usage of specific URI parts in the `rps3:MessageId` (e.g. fragments) may result in `wsa:MessageID` which has `rps3` fields in either the fragment or in the query part of the resulting URI.

For example:

```
<rps3:MessageId>//MySenderDomain/123#fragment456</rps3:MessageId>
<rps3:TransactionId>MyTansactionID1</rps3:TransactionId>
<rps3:ContextId>MyContextID1</rps3:ContextId>
<rps3:MessageDate>2013-04-18T09:45:00Z</rps3:MessageDate>
<rps3:ProfileType>3G_16K</rps3:ProfileType>
```

May result in two possible mappings, both allowed per URI format but questionable as `wsa:MessageId` will be possibly misinterpreted.



- `<wsa:MessageID>//MySenderDomain/123#fragment456?TransactionId=MyTansactionID1?ContextId=MyContextID1?MessageDate=2013-04-18T09:45:00Z?ProfileType=3G_16K</wsa:MessageID>`
 - Despite being a valid URI, this field may not be correctly interpreted as a `wsa:MessageId` per SGP.02, as the mandatory query parameter “MessageDate” is not part of the URI query parameter, as it is swallowed in the fragment part of the URI.
- `<wsa:MessageID>//MySenderDomain/123?TransactionId=MyTansactionID1?ContextId=MyContextID1?MessageDate=2013-04-18T09:45:00Z?ProfileType=3G_16K#fragment456</wsa:MessageID>`
 - This is also a valid URI, however, this field is questionable as `wsa:MessageId` is ambiguous; despite RFC 3986 allows fragments, RFC 3987 does not allow fragments in the `absolute-IRI` syntax. Moreover, the `rps3:MessageId` is split in two separate parts which makes it ambiguous for both function provider and function requester to reconstruct it.

Problems could be caused by the presence of characters `'#','?'`, which delimit specific parts of a URI, in `rps3` fields that will be encapsulated in a URI.

Recommendations

To avoid ambiguity and interoperability problems due to different expectations of the positions of the `rps3` fields in the resulting `wsa` field it is recommended to

- 1) Avoid the usage of fragments in `rps3` fields, in particular in `rps3:MessageId`,
- 2) Avoid the usage of the characters `'#','?'` in the value of `rps3` fields,
- 3) Percent-escape these characters, as per RFC 3986 section 2.1, in the resulting `wsa` fields, if they are present in `rps3` fields.