



CHAPTER 7

SIP URI API

Lua 環境には、解析済み SIP URI を操作するための一連の API が用意されています。次の SIP URI API が使用できます。

- 「[applyNumberMask](#)」
- 「[getHost](#)」
- 「[getUser](#)」
- 「[encode](#)」

applyNumberMask

`applyNumberMask(mask, mask-char)`

この関数は、番号マスクを受け取り、指定された番号マスクを解析済み URI のユーザ部に適用してから、修正されたユーザを解析済み `sipUri` オブジェクトに格納します。修正された URI のユーザ部を表す文字列が返されます。

番号マスクの適用

マスク パラメータは、URI のユーザ部に適用される変換を定義します。大文字の「X」は、番号マスクのワイルドカード部分を指定します。たとえば、マスクが「+1888XXXXXXX」の場合、例のユーザ部「4441234」にマスクが適用されたたすると、得られる文字列は「+18884441234」です。

マスクされるユーザ部に見つかった文字数が、マスクのワイルドカード文字数よりも少ない場合、最も左にあるワイルドカード文字は「X」で残されます。前述のマスクを例のユーザ部「1234」に適用すると、結果の文字列は「+1888XXX1234」になります。マスクされるユーザ部に見つかった文字数が、マスクのワイルドカード文字数よりも多い場合、ユーザ部の最も左にある文字が切り捨てられます。たとえば、マスク「+1888XXXX」がユーザ部「4441234」に適用される場合、結果の文字列は「+18881234」になります。

例：P-Asserted-Id エンティティの URI のユーザ部を、番号マスクを適用した後にローカル変数に値として格納します。

スクリプト

```
M = {}  
function M.inbound_INVITE(msg)  
    local uriString = msg:getUri("P-Asserted-Identity")  
    if uriString  
    then  
        local uri = sipUtils.parseUri(uriString)  
        if uri
```

```

        then
            local user = uri:applyNumberMask("+1919476XXXX")
        end
    end
end
return M

```

メッセージ

```

INVITE sip:1234@10.10.10.1 SIP/2.0
.
P-Asserted-Identity: <sip:1234@10.10.10.1>
.

```

出力/結果

```
Local variable user is set to "+19194761234"
```

getHost

```
getHost()
```

この関数は、解析した sipUri オブジェクトのホスト部分を取り出して、それを文字列として呼び出し側に返します。

例：P-Asserted-Identity ヘッダー内の URI のホスト部分をローカル変数に値として格納します。

スクリプト

```

M = {}
function M.inbound_INVITE(msg)
    local uriString = msg:getUri("P-Asserted-Identity")
    if uriString
    then
        local uri = sipUtils.parseUri(uriString)
        if uri
        then
            local host = uri:getHost()
        end
    end
end
return M

```

メッセージ

```

INVITE sip:1234@10.10.10.1 SIP/2.0
.
P-Asserted-Identity: <sip:1234@10.10.10.1>
.

```

出力/結果

```
Local variable host is set to "10.10.10.1"
```

getUser

```
getUser()
```

この関数は、解析した sipUri オブジェクトのユーザ部分を取り出して、それを文字列として呼び出し側に返します。

例：P-Asserted-Identity ヘッダー内の URI のユーザ部分をローカル変数に値として格納します。

スクリプト

```
M = {}
function M.inbound_INVITE(msg)
    local uriString = msg:getUri("P-Asserted-Identity")
    if uriString
    then
        local uri = sipUtils.parseUri(uriString)
        if uri
        then
            local user = uri:getUser()
        end
    end
end
return M
```

メッセージ

```
IINVITE sip:1234@10.10.10.1 SIP/2.0
.
P-Asserted-Identity: <sip:1234@10.10.10.1>
.
```

出力/結果

```
Local variable user is set to "1234"
```

encode

```
encode()
```

この関数は、解析した sipUri オブジェクトを文字列にエンコードして、それを呼び出し側に返します。エンコードに先立って解析済み sipUri オブジェクトに加えられた変更はすべて、得られる文字列に反映されます。

例：P-Asserted-Identity ヘッダーから得た URI を解析し、番号マスクを適用してから、得られた URI をエンコードします。

スクリプト

```
M = {}
function M.inbound_INVITE(msg)
    local uriString = msg:getUri("P-Asserted-Identity")
    if uriString
    then
        local uri = sipUtils.parseUri(uriString)
        if uri
        then
            uri:applyNumberMask("+1919476XXXX")
            uriString = uri:encode()
        end
    end
end
```

```
    end  
end  
return M
```

メッセージ

```
INVITE sip:1234@10.10.10.1 SIP/2.0  
.  
P-Asserted-Identity: <sip:1234@10.10.10.1>
```

出力/結果

```
Local variable uriString is set to "<sip:+19194761234@10.10.10.1>"
```