

1
2
3
4
5
6
7

8

9

URI Templates for Marlin

10

11

Version 1.0

12

Final

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

Source

Marlin Developer Community

Date

Sept 10, 2007

33

34 **Notice**

35 THIS DOCUMENT IS PROVIDED "AS IS" WITH NO REPRESENTATION OR
36 WARRANTY, EXPRESS OR IMPLIED, CONCERNING THE
37 COMPLETENESS, ACCURACY, OR APPLICABILITY OF ANY
38 INFORMATION CONTAINED IN THIS DOCUMENT. THE MARLIN
39 DEVELOPER COMMUNITY ("MDC") ON BEHALF OF ITSELF AND ITS
40 PARTICIPANTS (COLLECTIVELY, THE "PARTIES") DISCLAIM ALL
41 LIABILITY OF ANY KIND WHATSOEVER, EXPRESS OR IMPLIED,
42 ARISING OR RESULTING FROM THE RELIANCE OR USE BY ANY PARTY
43 OF THIS DOCUMENT OR ANY INFORMATION CONTAINED HEREIN. THE
44 PARTIES COLLECTIVELY AND INDIVIDUALLY MAKE NO
45 REPRESENTATIONS CONCERNING THE APPLICABILITY OF ANY
46 PATENT, COPYRIGHT (OTHER THAN THE COPYRIGHT TO THE
47 DOCUMENT DESCRIBED BELOW) OR OTHER PROPRIETARY RIGHT OF
48 THIS DOCUMENT OR ITS USE, AND THE RECEIPT OR ANY USE OF THIS
49 DOCUMENT OR ITS CONTENTS DOES NOT IN ANY WAY CREATE BY
50 IMPLICATION, ESTOPPEL OR OTHERWISE, ANY LICENSE OR RIGHT TO
51 OR UNDER ANY PATENT, COPYRIGHT, TRADEMARK OR TRADE
52 SECRET RIGHTS WHICH ARE OR MAY BE ASSOCIATED WITH THE
53 IDEAS, TECHNIQUES, CONCEPTS OR EXPRESSIONS CONTAINED
54 HEREIN.

55 Use of this document is subject to the agreement executed between you and
56 the Parties, if any.

57 Any copyright notices shall not be removed, varied, or denigrated in any
58 manner.

59 Copyright © 2003 - 2009 by MDC, 415-112 North Mary Avenue #383 Sunnyvale, CA
60 94085, USA. All rights reserved. Third-party brands and names are the property of
61 their respective owners.

62 **Intellectual Property**

63 A commercial implementation of this specification requires a license from the Marlin
64 Trust Management Organization.

65 **Contact Information**

66 Feedback on this specification should be addressed to: [editor@marlin-
community.com](mailto:editor@marlin-
67 community.com)

68 Contact information for the Marlin Trust Management Organization can be found at:
69 <http://www.marlin-trust.com/>

70

71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87

Contents

1	Introduction	4
1.1	Syntax Notation	4
1.2	Conformance Conventions	4
1.3	References	4
1.3.1	Normative References	4
1.3.2	Informative References	4
1.4	Definitions	4
1.5	Acronyms	5
2	URI Template Syntax	6
3	Template Transform	7
4	Template Variables	8
4.1	Marlin Variables	8
5	Examples	9

88 1 Introduction

89 The purpose of URI templates is to be able to represent partially-formatted URIs that
90 a Marlin Device can transform into a final URI, such as a URL, by substituting parts
91 of the URI template with the value of variables for which the value may not be known
92 at the time the URI template was created or transmitted.

93 1.1 Syntax Notation

94 This specification uses the Augmented Backus-Naur Form (ABNF) notation of
95 [RFC2234], including the following core ABNF syntax rules defined by that
96 specification: ALPHA (letters), DIGIT (decimal digits) and CHAR (character).

97 1.2 Conformance Conventions

98 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
99 "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
100 specification are to be interpreted as described in IETF RFC 2119 [RFC2119].
101

102 These capitalized key words are used to unambiguously specify requirements and
103 behavior that affect the interoperability and security of implementations. When these
104 key words are not capitalized they are meant in their natural-language sense.
105

106 All Elements of this specification are considered **Normative** unless specifically
107 marked **Informative**. All Normative Elements are **Mandatory** to implement, except
108 where such an element is specifically marked **OPTIONAL**. Finally, where **Normative**
109 elements are described as **OPTIONAL**, they MAY be omitted from an
110 implementation, but when implemented, they MUST be implemented as described.

111 1.3 References

112 1.3.1 Normative References

- [MRL-BB] Marlin Engineering Work Group
Marlin Broadband Delivery Specification
Version 1.2
- [RFC2234] Crocker, D. and P. Overell, "Augmented BNF for Syntax
Specifications: ABNF", [RFC 2234](#), November 1997
- [CORE] Marlin Engineering Work Group
Marlin Core System Specification
Version 1.3
- [RFC2119] Key words for use in RFCs to Indicate Requirement Levels.
Internet Engineering Task Force, 1997
- [RFC3986] IETF RFC 3986
Uniform Resource Identifiers (URI): Generic Syntax
- [RFC3629] F. Yergeau, "UTF-8, a transformation format of ISO 10646",
November 2003

113 1.3.2 Informative References

- [Marlin] Marlin
<http://www.marlin-community.com/>

114 1.4 Definitions

Percent-Encoding, Escaping of certain characters as specified in section 2.1 of
percent-encoded [RFC3986]

115
116

117 **1.5 Acronyms**

URI	Uniform Resource Identifier
UTF	Unicode Transformation Format

118

119 2 URI Template Syntax

120 A URI template is a UTF-8 encoded string [RFC3629] that contains zero or more
121 placeholder substrings. The syntax for a URI template is:

122

```
123 uri-template = 1*VCHAR
```

124

125 The URI template MAY contain one or more variable placeholders substrings. A
126 placeholder substring represents a single template variable. A placeholder substring
127 has the following syntax:

128

```
129 placeholder = "{" var-name "}"
```

```
130 var-name = [var-modifier] var-namespace ":" var-local-name
```

```
131 var-modifier = "~"
```

```
132 var-namespace = ALPHA
```

```
133 var-local-name = 1*( ALPHA / DIGIT / "%" / ":" / "@" / "-" / "." /  
134 "_" / "~" / "!" / "$" / "&" / "*" / "+" / "=" )
```

135

136

137 The template string MUST NOT contain any "{" or "}" characters outside the
138 placeholder substrings except in they percent-encoded form.

139 **3 Template Transform**

140 The template transform takes a template string as input and produces a URI string as
141 output. The output is obtained by replacing all placeholder substrings with the string
142 value of that placeholder's variable (var-name). Variable names are case sensitive.

143 The template **MUST** be constructed in such a way that the URI string produced is a
144 valid URI as defined in [RFC3986]. This implies that all reserved characters in the
145 values of the variable **MUST** be percent-encoded as specified in section 2.1 of
146 [RFC3986].

147

148 The value of a variable that is not understood by the entity that is performing the
149 transform depends on the presence of the optional variable modifier: if the modifier
150 “~” is present, the value **SHALL** be the fixed string “~”; if it is not present the value
151 **SHALL** be an empty string.

152 4 Template Variables

153 4.1 Marlin Variables

154 The variable namespace for Marlin variables is “m”.

155 The general syntax for a Marlin variable is:

156

```
157 marlin-var = match * ( "&" match )
158 match = attribute-name "=" attribute-value
159 attribute-name = "type" / URI
160 attribute-value = 1*VCHAR
```

161

162 attribute-name: either the fixed string “type” or the name of an Octopus Node
163 attribute as defined in section 3.2 of Marlin Core [CORE].

164

165 attribute-value: a string.

166

167 Since the “=” and “&” characters are used as delimiters, any occurrence of those
168 characters in attribute-name or attribute-value MUST be percent-encoded as
169 specified in section 2.1 of [RFC3986].

170

171 The value of the variable is the unique identifier (uid) of an Octopus node selected by
172 the client application.

173 The node selected by the application MUST satisfy all the match filters indicated in
174 the variable name using the ‘match’ syntax. Each match filter indicates an attribute
175 name (attribute-name) and an attribute value (attribute-value). A node satisfies the
176 match filter if and only if it has an attribute whose name is the filter’s attribute-name
177 with a value equal to the filter’s attribute-value. When attribute-name is ‘type’, the
178 node attribute to match is the node attribute named
179 ‘urn:marlin:core:node:attribute:type’.

180 Since the attribute names and attribute values to match may contain percent-
181 encoded characters, the attribute-name and attribute-value subcomponent strings
182 MUST be percent-decoded before matching.

183

184 If the application finds more than one node that can be selected, it MAY use other
185 criteria, such as user preferences, to select one of them.

186 The application MAY use the empty string for the value of this variable if it is not able
187 to select a suitable node.

188 **5 Examples**

Input	http://www.bok.net/music/get-token?user-node={m:type=user}&cid=8967F56D
Output	http://www.bok.net/music/get-token?user-node=urn:marlin:organization:boknet:user:006789F5&cid=8967F56D

189

Input	http://www.bok.net/music/get-token?user-node={m:type=user&urn:marlin:organization:test:MyAttribute=urn:marlin:acme:myServiceId}&cid=8967F56D
Output	http://www.bok.net/music/get-token?user-node=urn:marlin:organization:acme:myNode:006789F5&cid=8967F56D

190

Input	http://www.bok.net/test?z={z:known_var}&foo=3
Output	http://www.bok.net/test?z=some_value&foo=3

191

Input	http://www.bok.net/test?z={z:unknown_var}&foo=3
Output	http://www.bok.net/test?z=&foo=3

192

Input	http://www.bok.net/test?z={~z:unknown_var}&foo=3
Output	http://www.bok.net/test?z=~&foo=3

193