Example Document

Living Document, 4 February 2020

This version:
https://example.com/where/is/this/document/published/

Issue Tracking:
GitHub

Editors:
DASH Industry Forum

Table of Contents

1. Example content§

The text of the document is mostly authored using Markdown syntax. You can use *italicized text*, **bold text**, hyperlinks and `inline code blocks`.

There are unordered lists:

- Pollen
- Honey
- Bees
- Work

And there are ordered lists:

1. Aardvark
2. Abacus
3. Academic
    1. Subitems work in lists, too
    2. The list numbering is automatic

Block formatting can be useful for quotes and excerpts:

> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus tellus dolor, porttitor ut elit sed, vestibulum maximus velit. Maecenas at sollicitudin neque. Sed eu risus ullamcorper:
>
> - porttitor lacus at
> - convallis nunc
> - suspendisse id dolor urna
> - quervos murat
> - curabitur in eros diam

In quote blocks, you can still use all regular formatting. To disable formatting, use code blocks (see below).

There is a special syntax for "key-value" lists:

key
: value

another key
: another value

1.1. Special formatting of informative examples and notes

> Note: if a paragraph starts with "Note: " it gets special highlighting and block formatting. These paragraphs are considered informative.

> EXAMPLE 1
> Some paragraphs might be marked as informative examples.

1.2. Including code/XML snippets

There is a special syntax for code blocks. This disables markup processing:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Example website</title>
</head>

<body id="home">
  <h1>HTML5!</h1>
</body>
</html>
```

1.3. Including math§

When a \ne 0 there are two solutions to ax^2 + bx + c = 0 and they are x = {-b \pm \sqrt{b^2-4ac} \over 2a}.

1.4. References§

There is a shortcut syntax for cross references to chapters in the same document: § 1 Example content

There is a shortcut syntax for referencing well-known documents (RFCs etc) that are published on SpecRef. For example, [rfc2324] is an important one.

> Note: You can also change the reference text and just call it the coffee pot RFC.

There is a slightly different shortcut syntax for normative references [DASH-SystemIDs].

2. Some things require HTML§

Tables are defined as HTML and should be followed by `<figcaption>` and together enclosed by `<figure>`.

Usage	Algorithm
Content Key wrapping	AES256-CBC, PKCS #7 padding
Encrypted key MAC	HMAC-SHA512

Figure 1 *Some cryptographic algorithms.*

The `data` and `pre` classes enable some default styling for tables. Pick whichever you prefer. The above table uses `data`.

Images are also inserted as HTML.

Figure 2 *Just a random static example image.*

3. Automatic diagram generation§

Diagrams can be automatically generated from text files. See content of `Diagrams/` subdirectory for diagram source code.

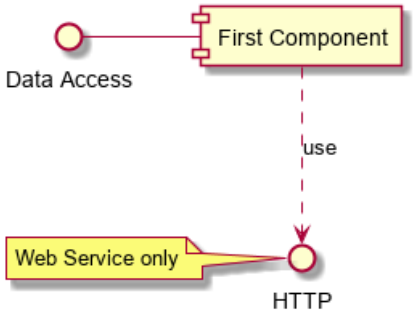

Figure 3 *Example for PlantUML component diagram.*

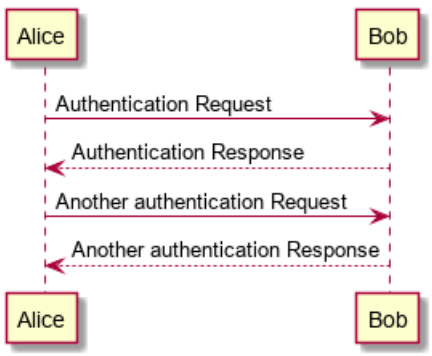

Figure 4 *Example for PlantUML sequence diagram.*

4. Manual diagrams§

Diagrams can also be managed manually, treated as static images. Often these are yEd diagrams (.graphml files) that are manually exported to PNG.

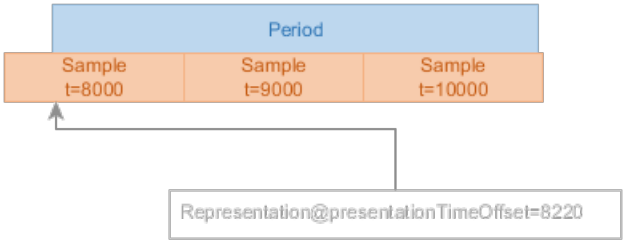

Figure 5 *Example yEd diagram.*

5. Defining terms§

The **Bikeshed document compiler** has a special syntax for various types of term/element definitions. This syntax enables easy cross-refrencing and building of the terminology index.

Terms can be defined either inline (as Bikeshed document compiler above) or in a key-value list:

foo
: bar

baz
: woo

Using term reference syntax will link back to the definition of the term: foo or baz.

> Note: every term must be referenced and every reference must point to a valid term. Terms with 0 references will result in a build error, just the same as broken references.

Use a pipe character to specify custom text for the generated link (e.g. for grammatical purposes):

Two bazes are better than three foos!

6. Defining data structures§

If your document defines data structures or languages, you will generally want to use the HTML/XML reference syntax of Bikeshed.

Consider the following XML structure consisting of `<bookstore>` and `<book>` elements:

```
<bookstore name="Ye Olde Booke Shoppe">
  <book title="Machine Learning for Machines" />
  <book title="List of letters in the English alphabet, 2nd ed" />
</bookstore>
```

The data structures in this snippet can be defined as the examples below illustrate. This type of definition allows easy referencing of elements and their children (e.g. `title`).

> Note: the data structure syntax shown here is not ideal but it is the closest we can get to a general-purpose Bikeshed data structure syntax that still enables automatic references. Notably, we cannot easily differentiate between XML element children and attributes.

6.1. `bookstore` element§

The root element of the bookstore document format.

`name` (required, xs:string)
: The human readable name of the bookstore.

`book` (0...N, `<book>`)
: Any number of books made available by the bookstore.

6.2. `book` element§

Defines one book that is published in a bookstore.

`title` (required, xs:string)
: The human readable title of the book.

7. Remember, this is Bikeshed not Markdown!§

Many editors have "Markdown preview" functions that will not be a 100% match to what will really be generated from the source code of this document. Do not be surprised if there are formatting differences.

Conformance§

Conformance requirements are expressed with a combination of descriptive assertions and RFC 2119 terminology. The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in the normative parts of this document are to be interpreted as described in RFC 2119. However, for readability, these words do not appear in all uppercase letters in this specification.

All of the text of this specification is normative except sections explicitly marked as non-normative, examples, and notes. [RFC2119]

Examples in this specification are introduced with the words “for example” or are set apart from the normative text with `class="example"`, like this:

EXAMPLE 2
This is an example of an informative example.

Informative notes begin with the word “Note” and are set apart from the normative text with `class="note"`, like this:

Note, this is an informative note.

Index§

Terms defined by this specification§

name

title

References§

Normative References§

[DASH-SystemIDs]

DASH-IF registry of DRM System IDs.. URL: https://dashif.org/identifiers/protection/

[RFC2119]

S. Bradner. Key words for use in RFCs to Indicate Requirement Levels. March 1997. Best Current Practice. URL: https://tools.ietf.org/html/rfc2119

Informative References§

[RFC2324]

L. Masinter. Hyper Text Coffee Pot Control Protocol (HTCPCP/1.0). 1 April 1998. Informational. URL: https://tools.ietf.org/html/rfc2324

↑