

Example Document

Living Document, 5 February 2019

This version:

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

Issue Tracking:

GitHub

Editors:

DASH Industry Forum

Table of Contents

1	Example content		
1.1	Special formatting of informative examples and notes		
1.2	Including code/XML snippets		
1.3	References		
2	Some things require HTML		
3	Automatic diagram generation		
4	Manual diagrams		
5	Defining terms		
6	Defining data structures		
6.1	bookstore element		
6.2	book element		
7	Remember, this is Bikeshed not Markdown!		
	Conformance		
	Index		
	Terms defined by this specification		
	References		
	Normative References		
	Informative References		

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
- 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. 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 <u>SpecRef</u>. 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.

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

Images are also inserted as HTML.



ДЕРЕВНЯ ЛЮБНИЦЫ (Любница)

Деревия получила своё название при русском царе Петре I, который здесь летом отдыхал и любовался красотой побережья. Наверно из-за прекрасного вида с пригорка Любница деревне и дали это название (любоваться).

дали это название (любоваться). Впервые деревня упоминалась в церковно-приходской книге города Пскова и его уезда в 1582 году. Возможно, что деревня существовала уже ранее - как причал. В 1749 году Любницы отмечается как само-стоятельное поселение. Жители занимались земледелием, рыбным промыслом, в основном ловили снетков, промыслом, в основном ловили слетков, супили их в больших оудиллыных лечах и продавали в Тарту и во Пскове. Купцы скупали рыбу у местных жителей и велли на продажу. Рыбой снабжали Печорский монастырь. В 19 вске образовалась крестьянская община и в 1882 году в общину входило 15 сегоских и русских деревень, в которых проживали 631 мужчин и 726 женщин. В настоящее время жители дотехници.

проживалиот мужчини гоженщин, В настоящее время жители деревни занимаются рыболовством и овоще-водством, особенно в большом коли-честве выращивают лук. Осенью блестят в поли статите выращивают лук. на полях длинные золотисто-желтые борозды лука. Уже убранный лук сушится под солицем на специальных нарах.

THE VILLAGE OF LÜÜBNITSA

The village got its name from tsar Peter I who sometimes took a holiday here to enjoy the beauty of the local beaches (the Russian word for enjoy is suggestive of the place-name Löübnitsa). It was probably because of the picturesque scene that this name was given to the village.

given to the village. The village was first mentioned in 1582 in the registry-book of the residents of the town and county of Pskov. The village may have existed as a harbour even earlier than that. In 1749 the village has been mentioned as a new settlement. The main activities of the settlers were cultivation of land, fishing and processing of fish, mostly sparing which was dried in large ovens and sold in Tartu and Pskov. The fish was bought by merchants on the spot and then exported. The inhabitants were obliged to eatch fish for the monsatery. In the 19th exported. The inhabitants were obliged to eatch fish for the monastery. In the 19th century Lüübnitsa made up a separate comminity. In 1882, according to the record, 631 of the inhabitants were men and 726 women. All in all there were 15 villages in the set of the 126 women. All mall there were 15 villages in the community, some of them being Russian villages. Even today the villager's, main activities are fishing and vegetable-growing. Onions are grown extensively and in autounin rows of C. den yellow onions in the vegetable garden and heaps of harvested onions in the yards can be sighted.

Figure 1 Just a random static example image.

Automatic diagram generation

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



Figure 2 Example for PlantUML component diagram.



Figure 3 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 4 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 cbookstore> and cbookstore> and cbookstore> and cbookstore>

```
<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. The root element of the bookstore document format.

name (required, xs:string)

The human readable name of the bookstore.

book (0...N, <u><book></u>)

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§

```
baz
Bikeshed document compiler
book
(element)
element-attr for bookstore
bookstore
```

```
foo
```

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]

1

 \rightarrow

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