Software Architecture Description

Version 1.0, 2023-09-04

AsciiDoc Syntax Examples

AsciiDoc sits right in between Markdown and LatTeX:

- it is human-readable, like Markdown;

- it can be rendered on GitHub, like Markdown;

- it is more versatile than (vanilla) Markdown and is still simpler (but less powerful) than LaTeX;

- it is more PDF-aware than Markdown, e.g., with article and book document types.

For such reasons, AsciiDoc is a great language for technical writing. It works well for books (see the Pro Git book), and is great for project documentation (see the Spring framework repository).

> **TIP** It should be also great to write your architecture document!

So, here is a quick guide for some handy basic syntax. However, if this is not enough the official AsciiDoc documentation offers much more!

Section

Subsection

Subsubsection

Text Decoration

This is a paragraph, with **bold text**, *italicized text*, `monospaced text`, and even <mark>highlighted text</mark>.

Lists

Bullets

- Item 1

- Item 2

Numbered

1. Item 1

2. Item 2

Definition

Term 1   Definition 1

Term 2   Definition 2

Quoting

> This is a quote.
>
> — Author, Source

Admonitions

NOTE This is a note.

WARNING This is a warning.

Code Blocks

```
print("Hello, World!")
```

Tables

Table 1. This is the table caption

Header 1	Header 2
Cell 1-1	Cell 1-2
Cell 2-1	Cell 2-2

Images

The root path to all images (even from within section file) is defined in the All images

Figure 1. This is the figure caption

Links, References, Bibliography and Footnotes

Cross-referencing looks like this:

- for sections - Section 1.
- for tables and figures - Table 1, Figure 1

Create **external links** (e.g., URLs) with an autolink (http://example.com) or a text replacement.

Bibliography must be defined in a separate file (`references.adoc`). Once defined, you cite them like this [1]. For the numbering to work, make sure to give both label and number:

```
- [[[smith23,1]]] Smith, John....
```

Footnotes can be done lie this.[1] But there is a limitation for PDFs, they currently only appear as endnotes (i.e., together with the bibliography). So you may want to avoid them and just create a bibliographic entry.

Comments

```
// This is a single-line comment.

////
This is a block comment.
////
```

Authors

Acronym	Name	E- Mail	
jd	John Doe	john.doe@email.org	

Revision History

Date	Revision	Author	Description	
2023-09-03	1.0	jd	Initial Draft	

Table of Contents

Glossary

Term 1
> Description

Term 2
> Description

1. System Context

Description of the position of the envisaged system in its environment.

2. Business Information

2.1. Business Vision

Description of the envisaged business opportunities and unique selling points.

Advantages and disadvantages with respect to the current situation.

2.1.1. Advantages and Disadvantages

2.1.2. Business Rationale

2.2. Product/Service Description

And its evolution over time.

2.3. Target Audience

2.4. Business/Domain Model

2.5. Roadmaps

2.6. Financial Model

2.7. Competitors

3. Requirements

3.1. Architectural Vision

3.2. Stakeholders and their Concerns

Customer, users, implementers, configurators, maintainers, resellers, etc.

3.3. Stories and Use-cases

3.4. Functional Requirements

3.5. Commercial Non-Functional Requirements

Product appearance, prizing, etc.

3.6. Technical Non-Functional Requirements

Dependability (timeliness, performance, reliability & safety, availability, security), adaptability, scalability, maintainability, interoperability, etc.

3.7. Evolution Requirements

Typical change-cases with respect to the environment, the features and the technology of the system.

3.8. Risk Assessment

4. Analysis

4.1. Assumptions

4.2. Technology Roadmaps

4.3. Design Alternatives

Important design alternatives, their advantages/disadvantages, their feasibility.

The major design decisions must be documented together with their rationale.

5. System Architecture

5.1. Initial Models

Several important alternatives should be considered.

Design rationale and summary of the set of alternatives that will be worked out subsequently.

5.2. Elaborated Model

5.3. Verification

Verification of important requirements like costs, end-to-end performance in terms of responsiveness and system resource usage, reliability, etc.

6. Hardware Architecture

High-level, reverse-engineered description of the hardware platform and its application interfaces.

7. Software Architecture

7.1. Software Architecture Design

Attribute-Driven design.

7.2. Architectural Views

E.g., 4+1, including patterns and tactics

7.3. Components

Including their functionality, interfaces and interactions.

7.4. Decision Views

8. Architecture Evaluation

8.1. Requirements Verification

8.2. ATAM/SAAM/etc.

Including conclusions/improvements. Use scenarios from Chapter 3 (Requirements) when applying ATAM.

9. System Evolution

Architecture Decision Records

Title

Change Log

Date	Status
2023-09-04	proposed
2023-09-04	superseded by [adr3-new]

What is the status, such as proposed, accepted, rejected, deprecated, superseded, etc.?

Context

- Use case/Story:

What is the **issue** that we're seeing that is motivating this decision or change?

Considerations

What are the alternatives? What are the forces?

Decision

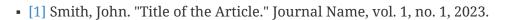What is the change that we're proposing and/or doing?

Consequences

What becomes easier or more difficult to do because of this change?

Related Decisions

What other decisions are affected by or influence this decision? Elaborate.

References

- [1] Smith, John. "Title of the Article." Journal Name, vol. 1, no. 1, 2023.
- [2] Johnson, Emily. "Another Title." Another Journal, vol. 2, no. 1, 2023.

[1] This is the footnote.