

# Online-Dokumentation für Nutzer mit AsciiDoc und Antora

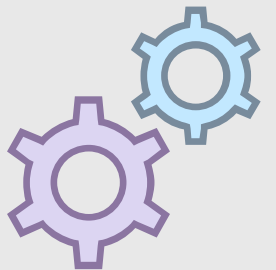
Alexander Schwartz, Principal IT Consultant  
FrosCon (Virtual), 2021-02-22



**Inhalte sind aktuell.**



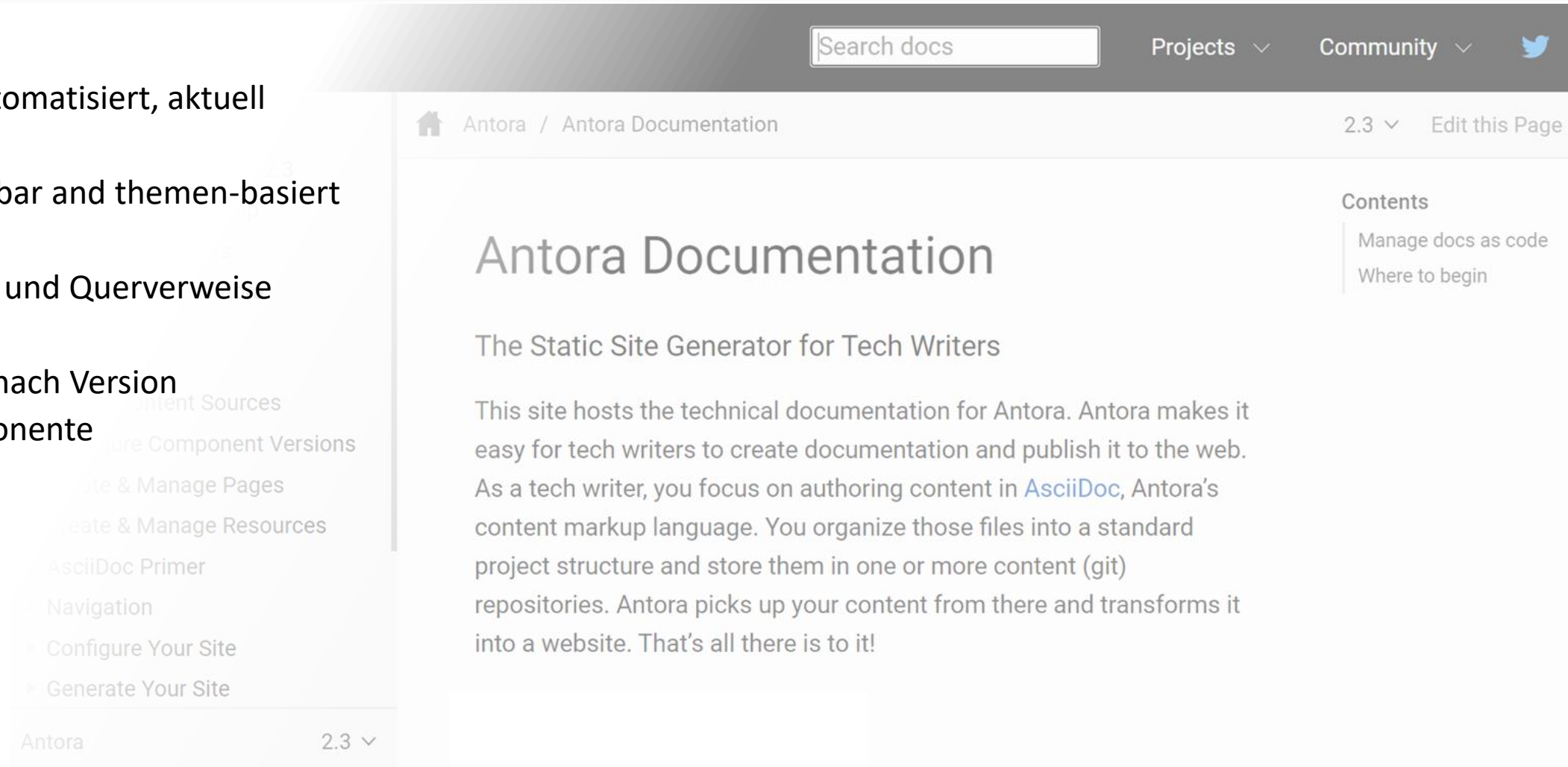
**Inhalte sind konsistent.**

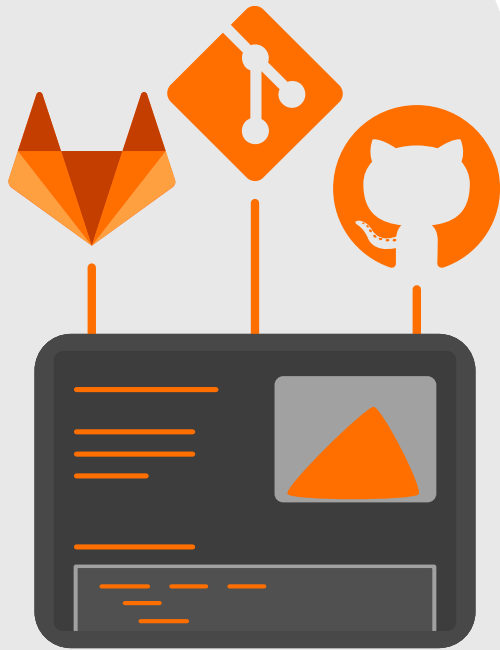


**Neue Funktionen und ihre Dokumentation sind gleichzeitig live.**

# Dokumentation unterstützt die Nutzer bei ihrer Arbeit

- Online, automatisiert, aktuell
- Durchsuchbar and themen-basiert
- Navigation und Querverweise
- Gruppiert nach Version und Komponente





- Strukturiert Dokumentation in Komponenten und Module
- Sammelt AsciiDoc-Inhalte aus verschiedenen Git-Repositories und Branches
- Konvertiert AsciiDoc Inhalte in HTML
- Verbindet das HTML mit dem UI Theme und erzeugt eine statische Web-Site

#### Hauptvorteile:

- Schnelle Konvertierung auf Basis von Node.js mit wenigen abhängigen Modulen
- Modular und erweiterbar

## AsciiDoc

- Schreiben ohne Ablenkung mit einfachem Text
- Großer Funktionsumfang für technische Dokumentation
- Seit über 15 Jahren verfügbar und weiterentwickelt
- Standardisierung in 2020 bei der Eclipse Foundation begonnen

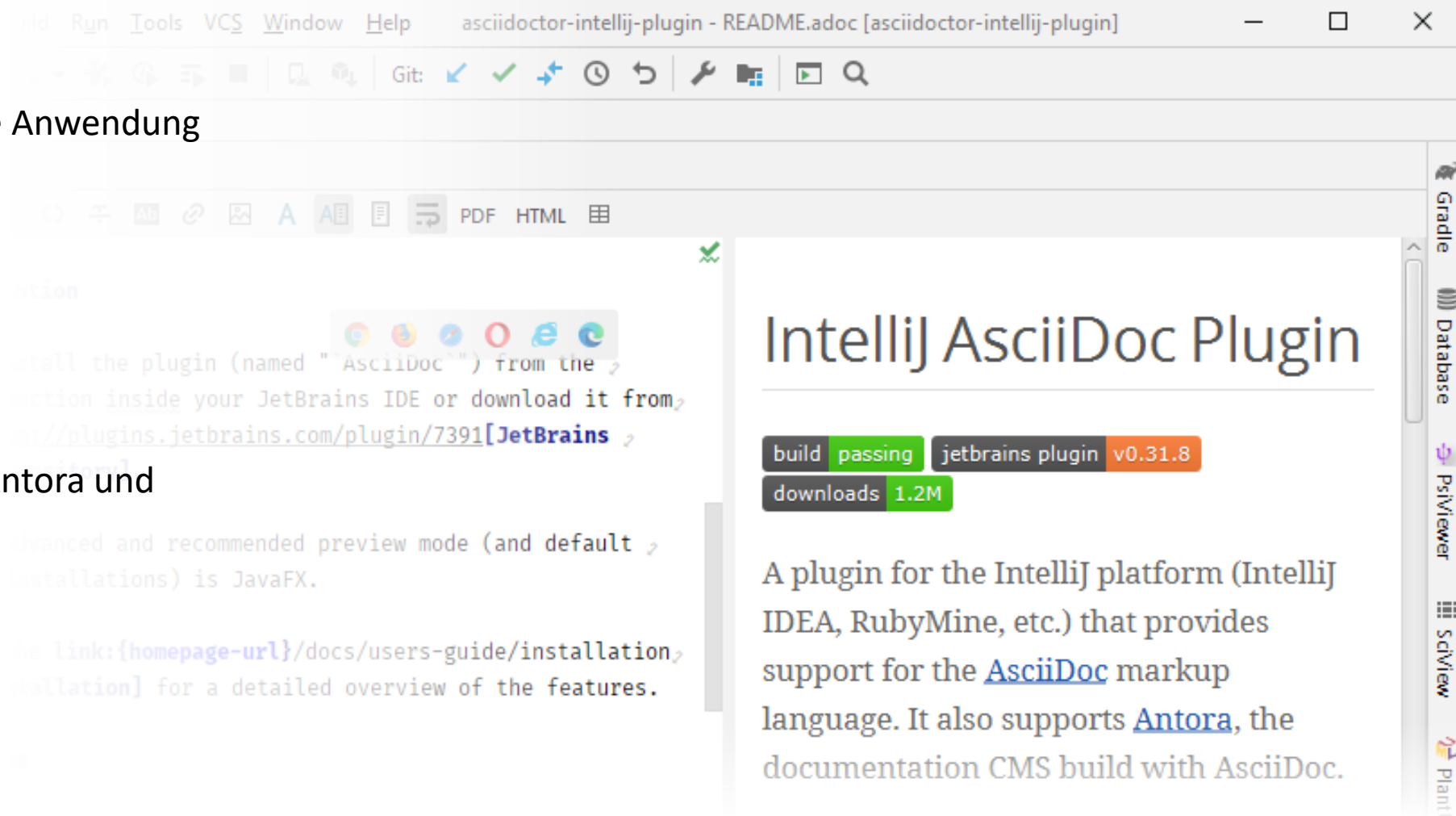
## AsciiDoctor

- Werkzeug, das aus den AsciiDoc-Textdateien HTML und viele andere Formate erstellt
- Open Source
- Läuft in Java, Ruby und JavaScript Laufzeitumgebungen
- Genutzt von Antora für die AsciiDoc-zu-HTML Konvertierung



AsciiDoctor  
<https://asciidoctor.org/>

- Fokussiert bleiben und die Anwendung nicht wechseln
- Zusammenarbeit über die Versionsverwaltung
- IntelliJ AsciiDoc Plugin bietet Unterstützung für Antora und AsciiDoc



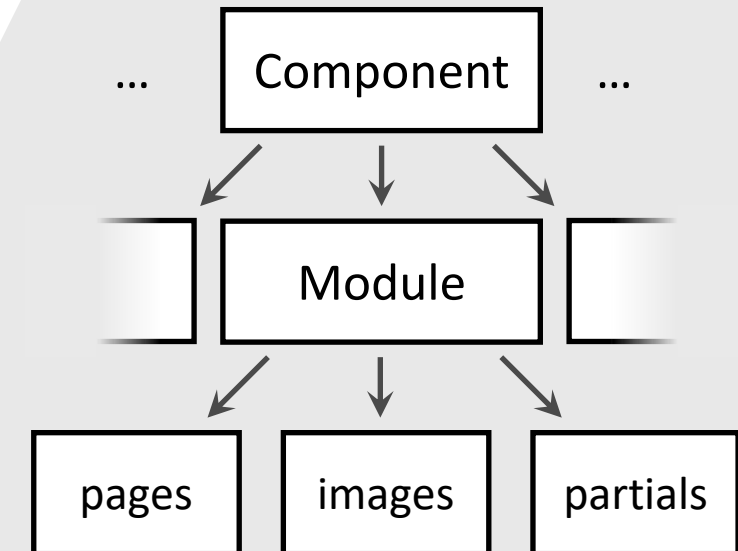
List of editors:

<https://asciidoctor.org/docs/editing-asciidoc-with-live-preview/>

- Mehrere unabhängige Components
- Components definiert über Name and Version
- Jede Component kann mehrere Module haben
- Jedes Modul hat vordefinierte Gruppen (*families*) für Seiten (*pages*), Bilder (*images*), Beispiele (*examples*), Anhänge (*attachments*) und Abschnitte, die an mehreren Stellen eingebunden werden können (*partials*)

## Erweitere Funktion:

Komponenten können über mehrere Ordner, Branches und Repositories verteilt werden.

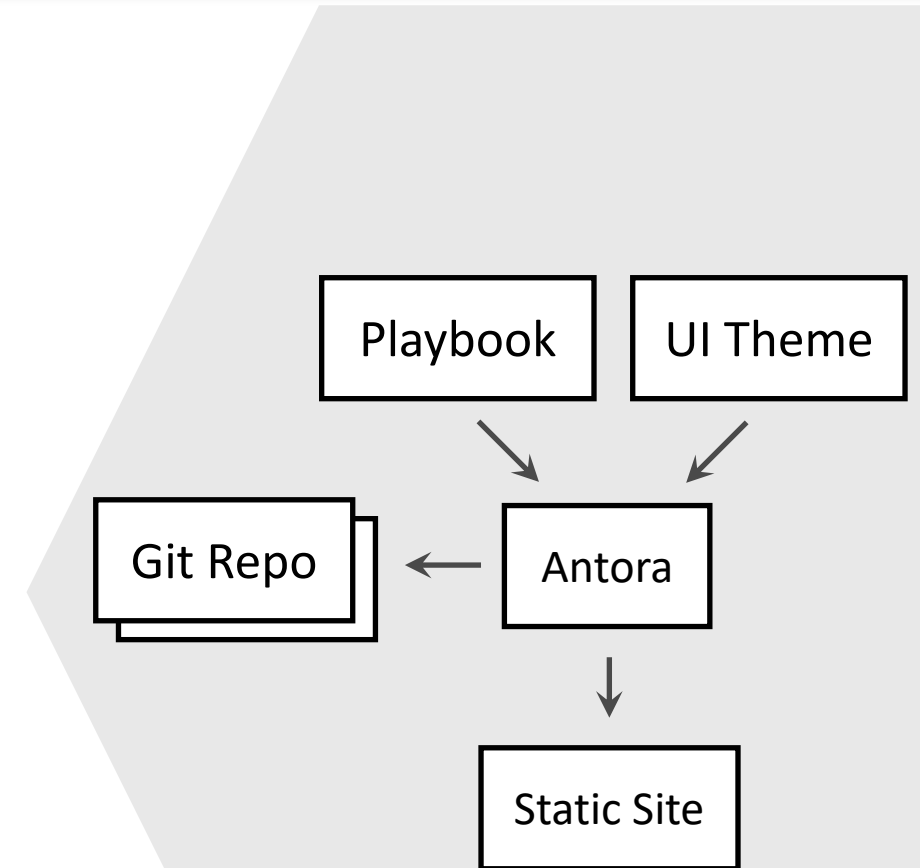


## Was die Nutzer machen:

1. Antora starten

## Was Antora macht:

1. Antora Playbook laden mit der Liste der Repositories und Branches
2. Git Repository clonen
3. AsciiDoc in HTML wandeln
4. Mit UI-Theme und UI-Anpassungen verbinden
5. Optional: Suchindex erstellen
6. Im Ausgabeverzeichnis eine statische Website erstellen





Rolle	Professionelle Fähigkeiten	Technische Fähigkeiten
Autor	Strukturieren und Schreiben	<ul style="list-style-type: none"><li>• Git (für die Zusammenarbeit mit anderen Autoren)</li><li>• AsciiDoc (um Inhalte zu erstellen)</li></ul>

Rolle	Professionelle Fähigkeiten	Technische Fähigkeiten
Autor	Strukturieren und Schreiben	<ul style="list-style-type: none"><li>• Git (für die Zusammenarbeit mit anderen Autoren)</li><li>• AsciiDoc (um Inhalte zu erstellen)</li></ul>
Docu Ops	Automatisierung und Infrastruktur	<ul style="list-style-type: none"><li>• Git (für die erstmalige Einrichtung)</li><li>• YAML (für die Konfiguration)</li><li>• package.json (für Versionsmanagement)</li><li>• Continuous Integration (für Automatisierung)</li></ul>

Rolle	Professionelle Fähigkeiten	Technische Fähigkeiten
Autor	Strukturieren und Schreiben	<ul style="list-style-type: none"><li>• Git (für die Zusammenarbeit mit anderen Autoren)</li><li>• AsciiDoc (um Inhalte zu erstellen)</li></ul>
Docu Ops	Automatisierung und Infrastruktur	<ul style="list-style-type: none"><li>• Git (für die erstmalige Einrichtung)</li><li>• YAML (für die Konfiguration)</li><li>• package.json (für Versionsmanagement)</li><li>• Continuous Integration (für Automatisierung)</li></ul>
Web Developer	Web Site Erstellung	<ul style="list-style-type: none"><li>• HTML/CSS (um das Antora Theme anzupassen)</li></ul>

## Autor:

1. Erstellt erste Ordner-Struktur für erste Component
2. Erstellt *antora.yml* und vergibt Name und Version
3. Erstellt *index.adoc* im Verzeichnis *pages*
4. Erstellt Navigationseinträge in *nav.adoc* und fügt es zu *antora.yml* hinzu

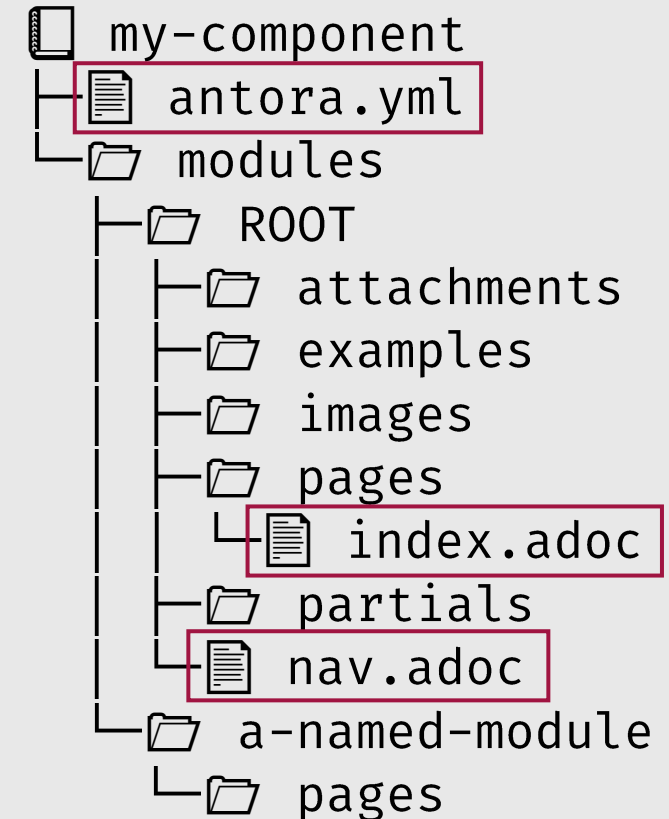
## Docu Ops:

1. Installiert Antora (z. B. npm CLI oder *package.json*)
2. Erstellt Playbook für den Autor für eine lokale Vorschau
3. Erstellt Playbook das die Website für das Internet erstellt

## Web Developer:

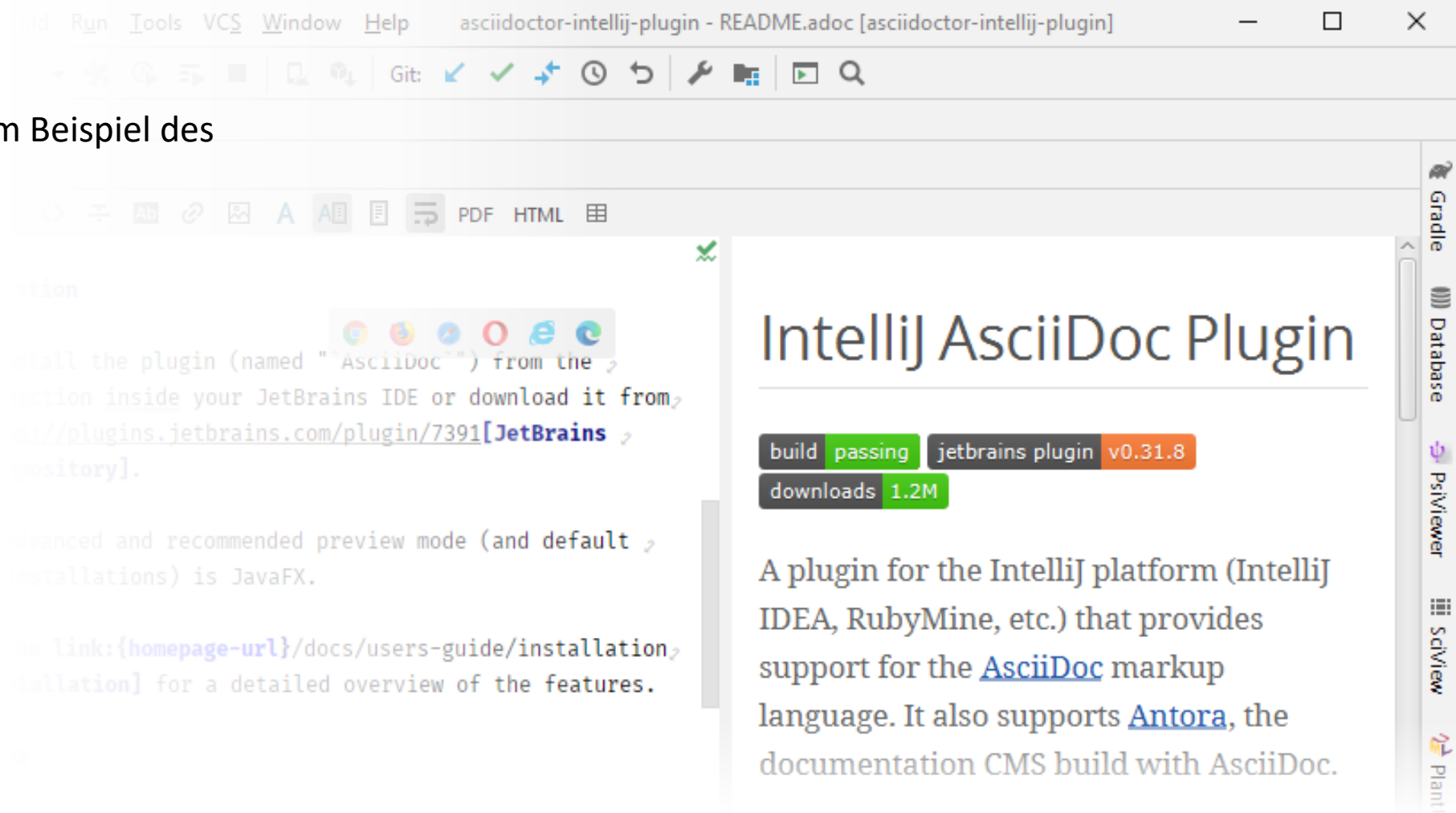
1. Kopiert das bestehende UI-Theme (oder erstellt ein eigenes)
2. Legt UI-Anpassungen im Ordner *supplemental\_ui* ab.

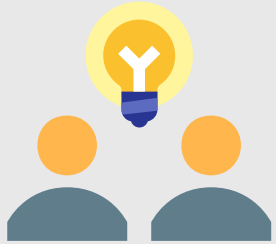
Quick Start Guide:  
<https://docs.antora.org/antora/2.3/install-and-run-quickstart/>



# Zeit für eine Live-Demo!

Wie alles zusammenspielt am Beispiel des IntelliJ AsciiDoc Plugins.





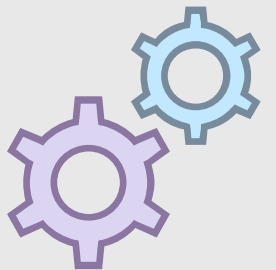
## **Inhalte sind aktuell**

durch Zusammenarbeit im Team über Versionskontrolle in Git.



## **Inhalte sind konsistent**

durch Komponentenversionen und de-duplizierung von Inhalten.



## **Neue Funktionen und ihre Dokumentation sind gleichzeitig live**

durch CI/CD Werkzeuge und automatisiertes Publizieren.

## **Asciidoctor**

<https://asciidoctor.org/>

## **IntelliJ AsciiDoc Plugin**

<https://intellij-asciidoc-plugin.ahus1.de/>

## **PlantUML**

<https://plantuml.com/>

<https://real-world-plantuml.com/>

## **Antora**

<https://antora.org/>

## **Videos, slides and examples**

<https://www.ahus1.de/post/asciidoctor-intro-and-deep-dive>

<https://www.ahus1.de/post/cdc-antora-live>

<https://www.ahus1.de/post/documentation-site-antora>

## **Asciidoctor Kroki extension**

<https://github.com/Mogztter/asciidoctor-kroki>



# Contact



Alexander Schwartz  
Principal IT Consultant

+49 171 5625767  
alexander.schwartz@msg.group



msg systems ag  
Robert-Bürkle-Straße 1  
85737 Ismaning  
Germany

value – inspired by people