Writing documentation with Asciidoctor

Jérémie Bresson – 03.11.2015

AsciiDoc

AsciiDoc example

```
= Getting Started with Java
Author Name <author@example.org>

== Hello World example

Copy the *HelloWorld.java* file.

TIP: The application prints _Hello World!_ to the console.

* Compile this source to a class file using `javac`.
* Run the compiled class file using `java`.

If you need help with the Java syntax check the
link:http://docs.oracle.com/javase/8/docs/[Java 8 Javadoc].
```

AsciiDoc example

Getting Started with Java

Hello World example

Copy the **HelloWorld.java** file.

The application prints *Hello World!* to the console.

- Compile this source to a class file using `javac`.
- Run the compiled class file using `java`.

If you need help with the Java syntax check the Java 8 Javadoc.

AsciiDoc example

```
= Getting Started with Java
Author Name <author@example.org>

== Hello World example

Copy the *HelloWorld.java* file.

TIP: The application prints _Hello World!_ to the console.

* Compile this source to a class file using `javac`.
* Run the compiled class file using `java`.

If you need help with the Java syntax check the
link:http://docs.oracle.com/javase/8/docs/[Java 8 Javadoc].
```

Headings

```
= Getting Started with Java
Author Name <author@exampl

== Hello World example

Copy the *HelloWorld.java* file.

TIP: The application prints _Hello World!_ to the console.

* Compile this source to a class file using `javac`.
* Run the compiled class file using `java`.

If you need help with the Java syntax check the
link:http://docs.oracle.com/javase/8/docs/[Java 8 Javadoc].
```

Document Title

Section Level 1

Lists

```
= Getting Started with Java
Author Name <author@example.org>

== Hello World example

Copy the *HelloWorld.java* file.

TIP: The application prints _Hello World!_ to the console.

* Compi                                    s file using `javac`.
* Run t                                    sing `java`.

If you need help with the Java syntax check the
link:http://docs.oracle.com/javase/8/docs/[Java 8 Javadoc].
```

Unordered list items

Inline formatting

```
= Getting Started with Java
Author Name <author@example.org>

== Hello World example

Copy the *HelloWorld.java* file.

TIP: The application prints _Hello World!_ to the console.

* Compile this source to a class file using `javac`.
* Run the compiled class file using `java`.

If you need help with the Java syntax check the
link:http://docs.oracle.com/javase/8/docs/[Java 8 Javadoc].
```

bold

italic

code

Links

```
= Getting Started with Java
Author Name <author@example.org>

== Hello World example

Copy the *HelloWorld.java* file.

TIP: The application prints _Hello World!_ to the console.

* Compile this source to a class file using `javac`.
* Run the compiled class file using `java`.

If you need help, check the
link:http://docs.oracle.com/javase/8/docs/[Java 8 Javadoc].
```

link with title

Admonition

```
= Getting Started with Java
Author Name <author@example.org>

== Hello World example

Copy the *HelloWorld.java* file.

TIP: T  blication prints _Hello World!_ to the console.

* Compile this source to a class file using `javac`.
* Run the compiled class file using `java`.

If you need help with the Java syntax check the
link:http://docs.oracle.com/javase/8/docs/[Java 8 Javadoc].
```

Asciidoctor

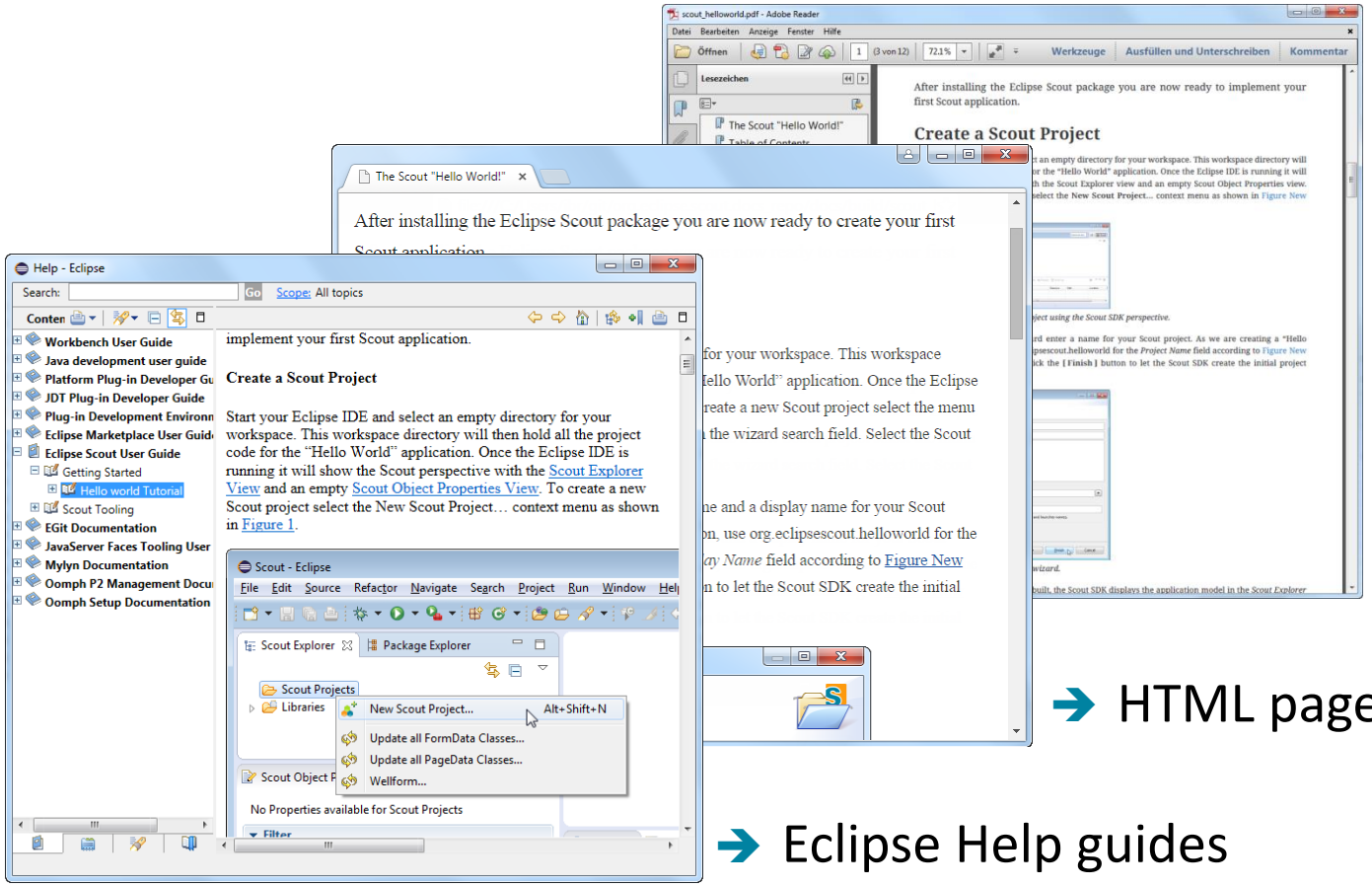
Asciidoctor

- Asciidoctor is **publishing toolchain** for the AsciiDoc writing format.

- Written in **Ruby** (and transpiled to **JavaScript**)
  (can be used with JRuby in the **JVM**)

- **Open Source** (MIT Licence)

Multiple Outputs



➔ PDF

➔ HTML pages

➔ Eclipse Help guides

«Why is asciidoctor a great tool?»

Source code listings

A code snippet in your documentation

```
== Usage of Mylyn wikitext
With the method `toHtml(String)` in <<lst-wikitext1>> you
can convert a text from Mediawiki Language to HTML.

[[lst-wikitext1, Listing 1]]
[source,java]
----
public static String toHtml(String mediawikiText) {
  StringWriter writer = new StringWriter();
  MarkupParser parser = new MarkupParser(new MediaWikiLangua
  parser.parse(mediawikiText);
  return writer.toString();
}
----
```

Callout

```
== Usage of Mylyn wikitext
With the method `toHtml(String)` in <<lst-wikitext1>> you
can convert a text from Mediawiki Language to HTML.

[[lst-wikitext1, Listing 1]]
[source,java]
----
public static String toHtml(String mediawikiText) {
  StringWriter writer = new StringWriter();
  MarkupParser parser = new MarkupParser(new MediaWikiLangua
  parser.parse(mediawikiText); // <1>
  return writer.toString();
}
----
<1> Main call of the Mylyn Wikitext framework
```

Callout

Usage of Mylyn wikitext

With the method `toHtml(String)` in Listing 1 you can convert a text from Mediawiki Language to HTML.

```
public static String toHtml(String mediawikiText) {
  StringWriter writer = new StringWriter();
  MarkupParser parser = new MarkupParser(new MediaWikiLanguage(), new HtmlDocumentBuilder(writer
  parser.parse(mediawikiText); (1)
  return writer.toString();
}
```

(1) Main call of the Mylyn Wikitext framework

Callout – Copy paste friendly

Usage of Mylyn wikitext

With the method `toHtml(String)` in Listing 1 you can convert a text from Mediawiki Language to HTML.

```
public static String toHtml(String mediawikiText) {
  StringWriter writer = new StringWriter();
  MarkupParser parser = new MarkupParser(new MediaWikiLanguage(), new HtmlDocumentBuilder(writer
  parser.parse(mediawikiText); (1)
  return writer.toString();
}
```

(1) Main call of the Mylyn Wikitext framework

Real source code in your documentation

- ➔ code snippets corresponding to source code under control
  - Always up-to-date
  - Ease maintenance

```
== Usage of Mylyn wikitext
With the method `toHtml(String)` in <<lst-wikitext1>> you
can convert a text from Mediawiki Language to HTML.

[[lst-wikitext1, Listing 1]]
[source,java]
----
include::src/main/java/MediawikiTest.java[tags=method]
----
<1> Main call of the Mylyn Wikitext framework
```

Real source code in your documentation

```
public class MediawikiTest {
  //tag::method[]
  public static String toHtml(String mediawikiText) {
      StringWriter writer = new StringWriter();
      MarkupParser parser = new MarkupParser(new MediaWikiLanguage(
      parser.parse(mediawikiText); // <1>
      return writer.toString();
  }
  //end::method[]
  @Test
  public void test() {
    StringBuilder sb = new StringBuilder();
    sb.append("= Heading 1 =\n");
    sb.append("\n");
    sb.append("Hello World!\n");
```

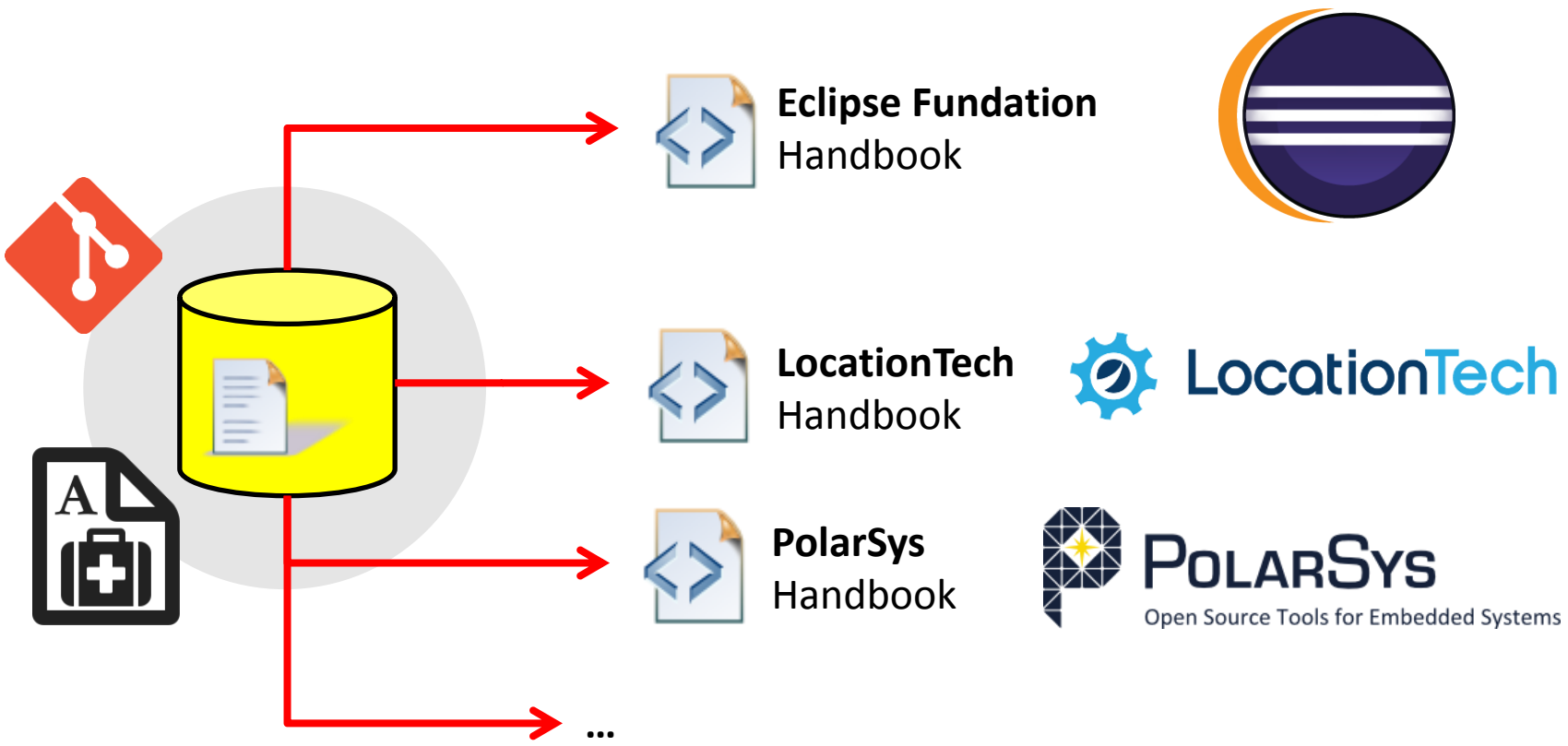
Single source your documents

«AsciiDoc lets me do variable substitution and conditional inclusion»

Wayne Beaton, Eclipse Foundation

https://waynebeaton.wordpress.com/2015/08/10/new-eclipse-project-handbook/

One source, multiple output



Variable substitution

```
:forgeName: Eclipse
:bugzillaUrl: https://bugs.eclipse.org/bugs

Eclipse Project Handbook
========================

include::chapters/resources.adoc[]
```

eclipse.adoc

```
Open source projects at the Eclipse Foundation are required
to make use of certain Eclipse Foundation services:

* All project issues must be tracked in the
{bugzillaUrl}[{forgeName} Bugzilla]instance;
```

resources.adoc

Variable substitution

```
:forgeName: LocationTech
:bugzillaUrl: https://www.locationtech.org/bugs

LocationTech Project Handbook
=============================

include::chapters/resources.adoc[]
```

locationtech.adoc


```
Open source projects at the Eclipse Foundation are required
to make use of certain Eclipse Foundation services:

* All project issues must be tracked in the
{bugzillaUrl}[{forgeName} Bugzilla]instance;
```

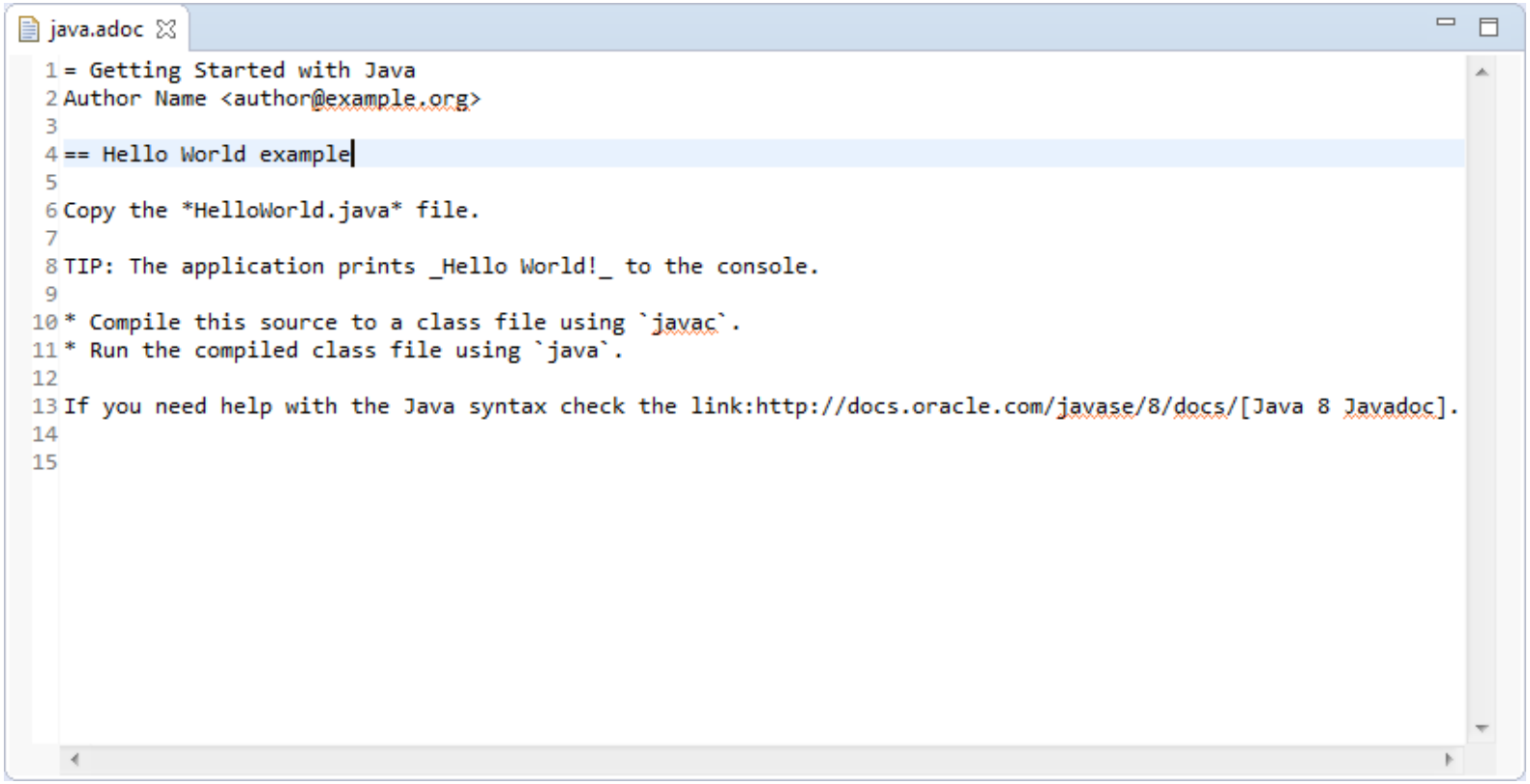
resources.adoc

Conditional inclusion

```
ifeval::["{forgeName}"=="Eclipse"]
_The Eclipse Foundation strongly encourages all projects to
create an
maintain and http://marketplace.eclipse.org[Eclipse
Marketplace]
presence._
endif::[]
```

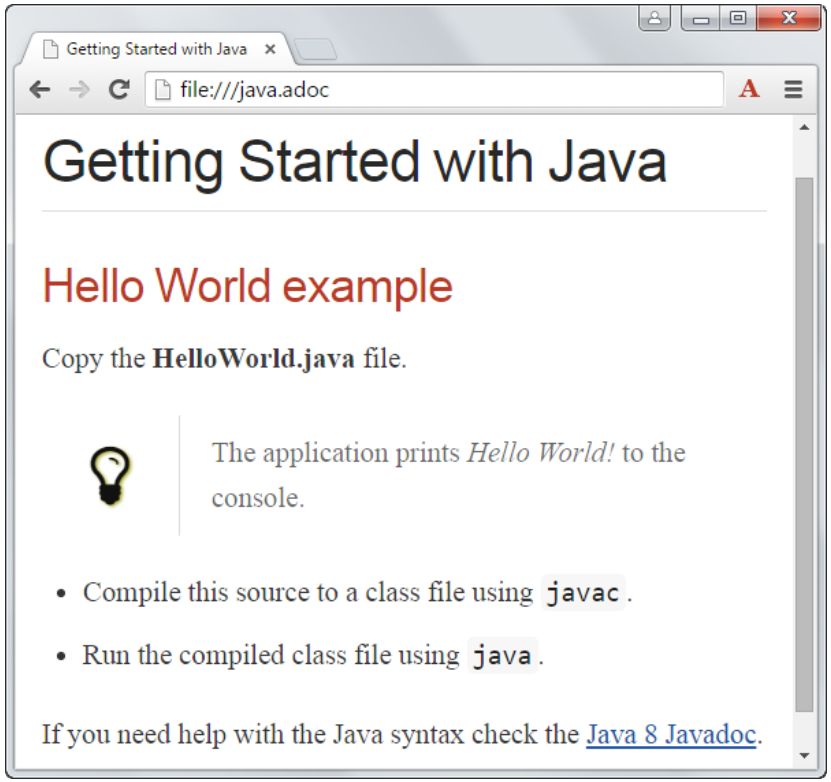
Simple Tools (no setup)

Any text editor

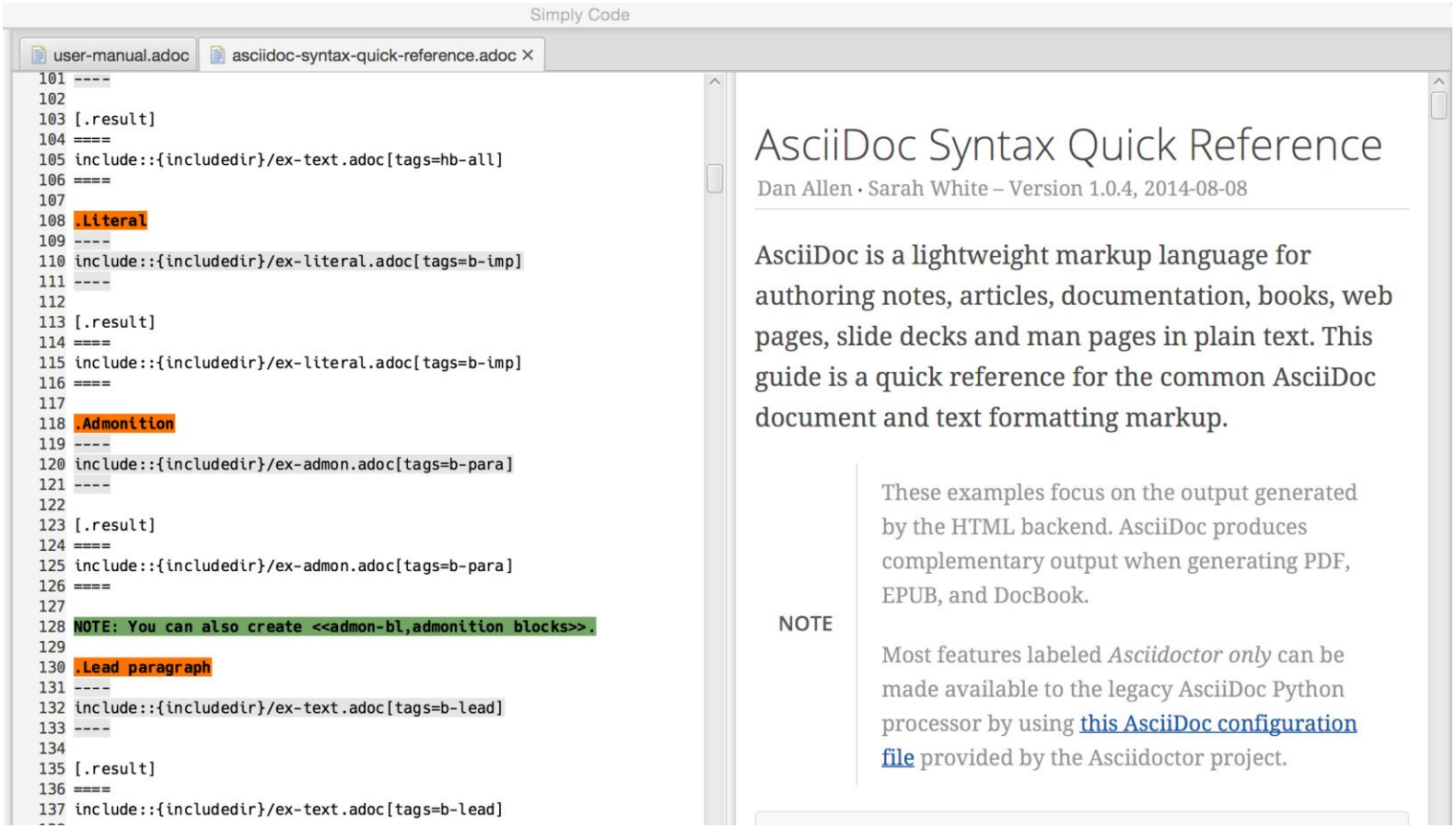


Live preview in Chrome

➔ Extension for Chrome: **Asciidoctor.js Live Preview**



JavaFX Codeeditors – Tom Schindl



Build

«Generating the documentation with Asciidoctor is always a straightforward experience»

Julien Ponge, Eclipse Golo project

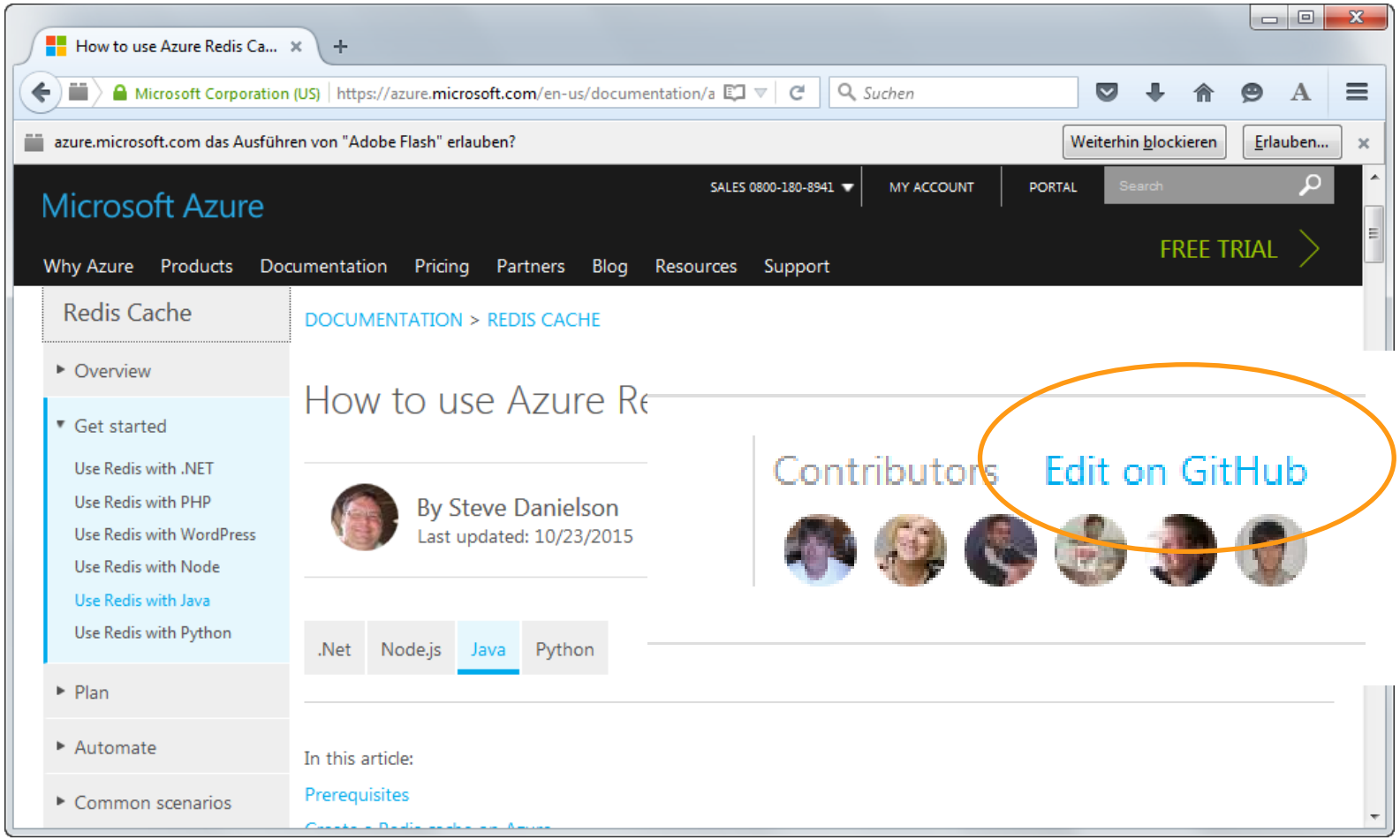
https://julien.ponge.org/blog/an-experiment-in-maven-to-gradle-migration/

Build system

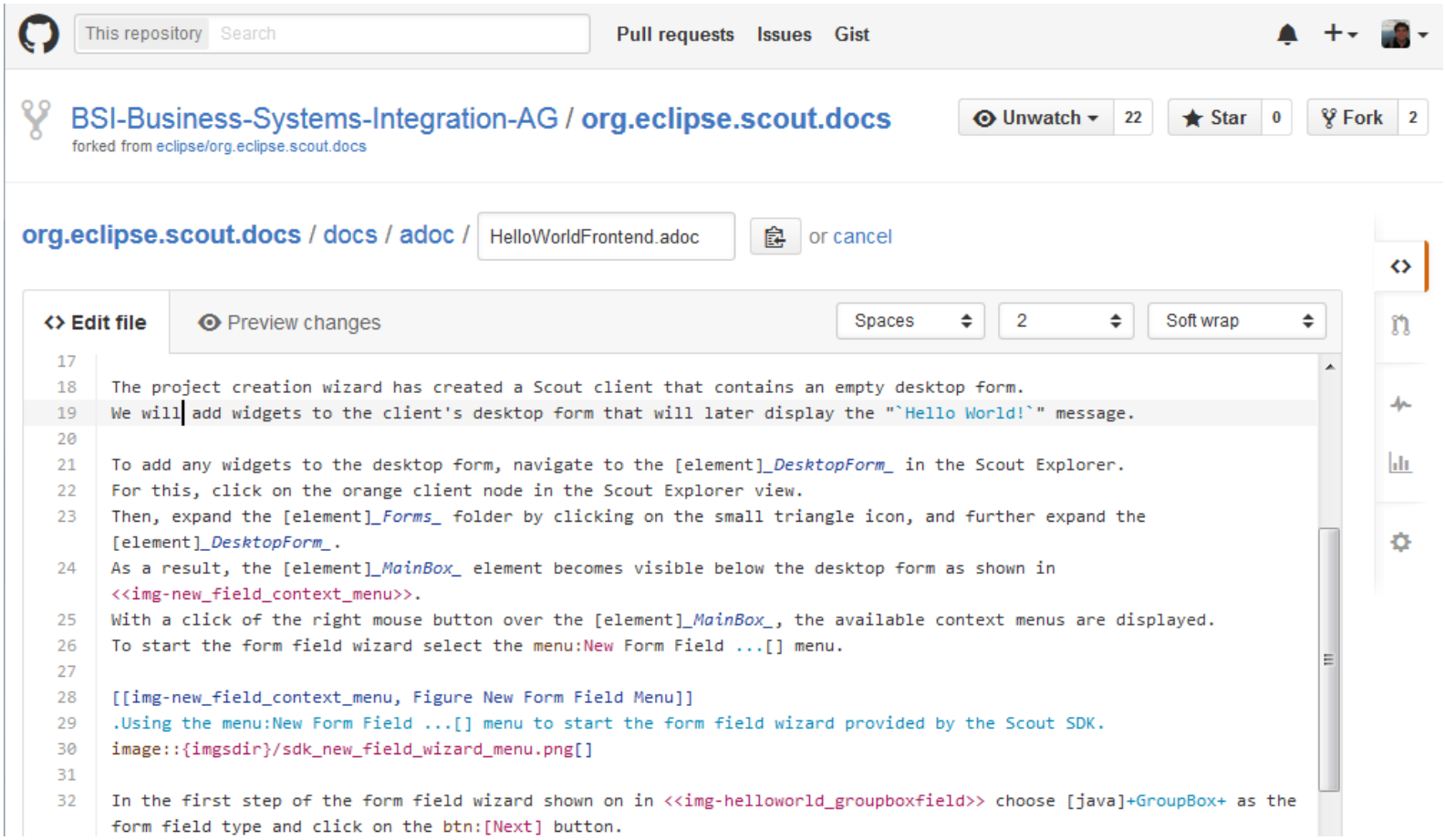
- Ruby, JavaScript, JVM environments
- Integration with a lot of build systems:
  - Maven
  - Gradle
- Dependencies available in several repositories
  - Maven central
  - JCenter
  - NPM
- You can build on every machine, without a complex setup

Edit on GitHub

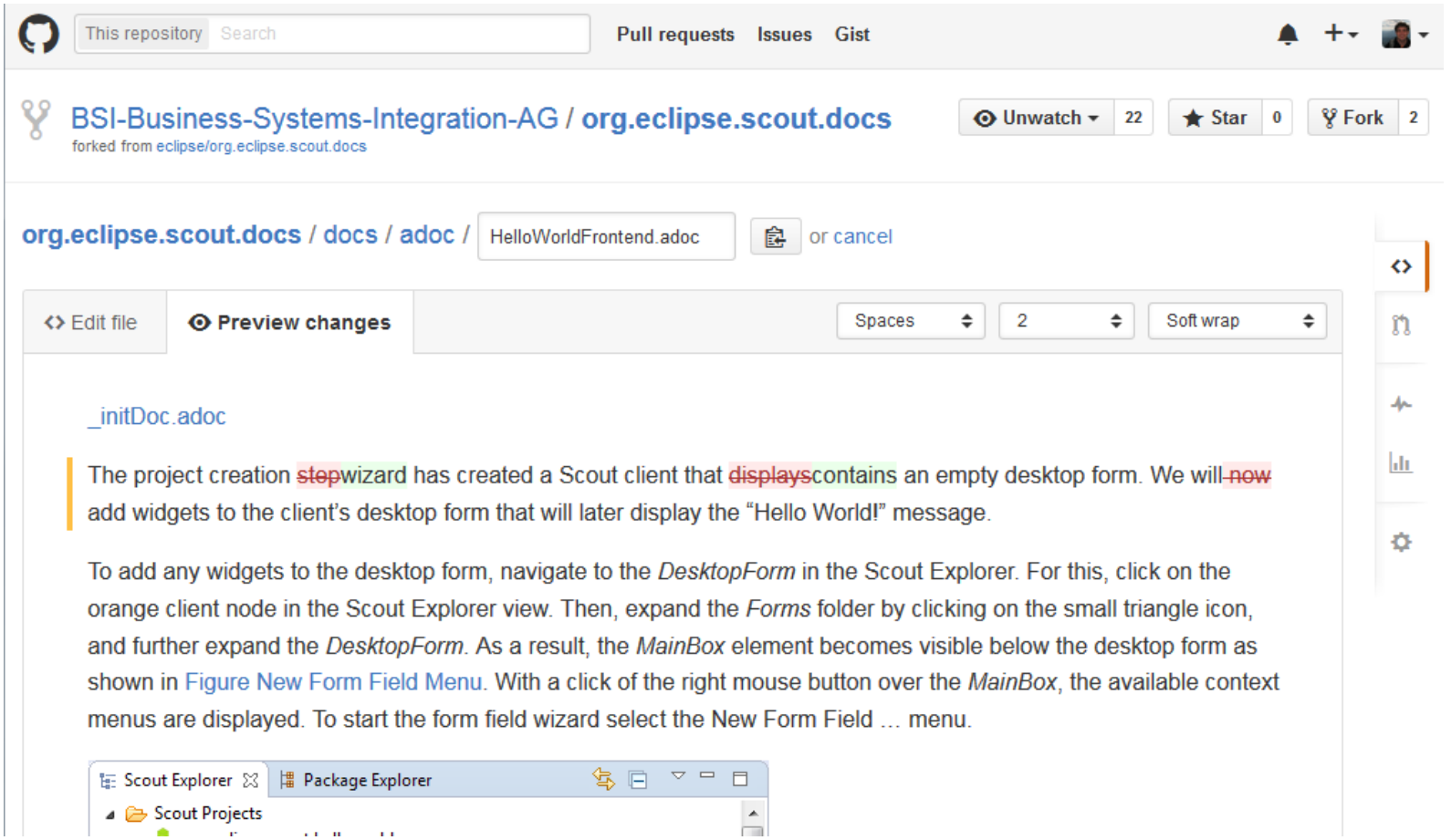
Edit on GitHub



GitHub Editor



GitHub Preview



Thanks

/@j2r2b
/+JeremieBresson
/jmini
jeremie.bresson@bsi-software.com