Gradle Guides (https://guides.gradle.org)

Writing Gradle Guides

build passing (https://travis-ci.org/gradle-guides/writing-getting-started-guides)

Table of Contents

This guide walks you through the process of writing a new guide for inclusion at the Gradle Guides site (https://guides.gradle.org/). These guides can take different forms—such as short, step-by-step tutorials or longer, more discussion-oriented pieces—but the process for writing and contributing them is the same.

What you’ll need

- A Git (https://git-scm.org) client (command line is fine, or use one of the many excellent GUI clients)
- A GitHub (https://github.com) account
- A Java Runtime Environment (http://www.oracle.com/technetwork/java/javase/downloads/index.html), version 1.7 or better
- A text editor
- A basic familiarity with Asciidoc (http://asciidoctor.org/docs/user-manual/)

Create a GitHub issue

The first step in creating a new Gradle guide is to submit a proposal to the gradle/guides (https://github.com/gradle/guides) GitHub repository. This is so that we, the Gradle Guides team, can work with you to determine the most appropriate subject, form and scope for the guide.

To submit your proposal:

1. Create a new issue (https://help.github.com/articles/creating-an-issue/) at gradle/guides (https://github.com/gradle/guides/issues) with a title that matches the suggested topic for the new guide. For example, "Publishing to private repositories" or "Creating fat JARs". The title should, in general, be a description of a task that a Gradle user might want to undertake.
2. Add a basic outline for the proposed guide, i.e. section headings
3. Engage with the team through this issue to nail down the final form of the guide

You don’t have to specify which form the guide should take in the initial proposal, but if you do, please express your preference as one of the following:

- Short, step-by-step introduction (getting-started guide)
- Long-form tutorial
- In-depth discussion about a single topic (topical guide)

Once the proposal has been accepted, the Gradle Guides team will create a new GitHub repository (https://github.com/gradle/guides/blob/master/README.md) for your guide. You won’t be able to commit directly to that repository. Instead, you will use the standard fork and pull request workflow described here.

Fork and clone the repository

When your guide’s repository is ready, you’ll find a link to it in the issue created by you in the previous step. You should then:

1. Fork the repository (https://help.github.com/articles/fork-a-repo/)
2. Clone your fork onto your local machine
3. Set an `upstream` remote that points to the original repo (See this help page (https://help.github.com/articles/configuring-a-remote-for-a-fork/) for details)

That's it! You're now ready to work on the content for your new guide using the infrastructure provided by the project you just cloned.

Render and view your guide locally

The project provides everything you need to generate the HTML version of your guide and view the result. Just run the following commands in the guide's root directory:

```
$ ./gradlew build                   1
$ open build/html5/index.html       2
```

1 Run Gradle (via the Gradle Wrapper) to generate HTML from your guide's Asciidoc sources

2 Open the generated HTML file in your system's default browser

> The `open` command is only needed on Mac OS X and other Linux flavors. On Windows, the path to the HTML file in a command prompt should automatically open it in a browser

At this point, you should have an HTML page that has a title and some section headings, but nothing else. It's your job to fill in the gaps! But before you start working on that content, it's worth having a basic understanding of the project structure.

Get familiar with your guide's structure and contents

The structure of a Gradle guide repository is very simple, consisting of just two basic components:

1. An Asciidoc source file called `README.adoc`
2. A Gradle build script to process it into HTML (which later will publish it to the GitHub Pages site where the guides are hosted)

The file you'll interact with most as a guide author is `README.adoc`, but it's useful to know what the other files and directories are for as well. The following listing walks through each of the top-level directory entries:

```
.
├── README.adoc           1
├── build/                2
├── build.gradle          3
├── gradle/               4
├── gradlew               5
└── gradlew.bat           6
```

1 Asciidoc source for the guide

2 Output directory for the build, where generated HTML for the guide can be found

3 Project build script, containing tasks for Asciidoc processing and GitHub Pages publication

4 Gradle Wrapper storage directory

5 Gradle Wrapper Unix shell script

6 Gradle Wrapper Windows batch file

Now you're ready to work on your new guide's content. You will edit the `README.adoc` file. The Gradle build will rename the result to `index.html`.

> Edit `README.adoc`, but preview `index.html` in the `build/html5` directory

Write your guide

Asciidoc is a pure ASCII text format, so you can use any text editor or IDE to work on `README.adoc`. Many of them have support for Asciidoc syntax highlighting, which makes things easier. Once you've decided on an editor, just open the file and get writing!

As you work on the main content of the guide, we recommend that you read the associated style guide (https://github.com/gradle-guides/style-guide/blob/master/README.adoc) and browse through a few of the existing guides (https://guides.gradle.org/) if you haven't done so already.

> *New to Asciidoc?*
>
> Refer to the Asciidoctor User Manual (http://asciidoctor.org/docs/user-manual/) while writing and you'll master the basics in no time. There is also a quick reference for syntax (http://asciidoctor.org/docs/asciidoc-syntax-quick-reference/) that's particularly useful when you just want to know the syntax for something specific.

When you want to preview your changes, simply regenerate your guide’s HTML with the following build command:

```
$ ./gradlew build
```

(Again, the "dot-slash" syntax is used on Unix. On Windows, just type `gradlew` or even `gradlew.bat`.)

Then open or refresh `build/html5/index.html` in your browser. It’s as simple as that.

Send a pull request

Writing is usually an iterative process. You write a draft, it gets reviewed, you do a second draft, and so on. That’s why we recommend that you submit your guide for review as soon as you want feedback. To do so, initiate a pull request (https://help.github.com/articles/creating-a-pull-request-from-a-fork/) when you’re ready.

Your guide enters the review process at this point. Each step of a review involves either feedback (as comments) or edits (as commits) depending on the current stage of the process. That’s why it’s important for you to specify what type of review you want. Should the Gradle Guides team treat the current version of your guide as a first draft or something that’s ready to be published? Do you simply want feedback on the structure and overall tone, or do you want the team to check for and fix spelling and grammar issues? You don’t have to be too specific and the team will ask for clarification if necessary.

Whenever you have a new draft ready, push the changes to your fork and add a comment to the pull request saying what type of review you would like. The pull request will update automatically and the review will continue until both sides are happy with the result.

You’ll find your guide on the Gradle Guides site (https://guides.gradle.org/) soon after the pull request is accepted and merged. Well done and thank you for the valuable contribution!

Summary

This guide documented the process to:

- Submit a proposal for a new Gradle guide
- Fork and clone the new guide’s repository
- Work on the guide’s content according to style guidelines
- Initiate a review of your guide

Now that you have some experience with it, we hope that you're inspired to contribute more guides in the future!

Help improve this guide

Have feedback or a question? Found a typo? Like all Gradle guides, help is just a GitHub issue away. Please add an issue or pull request to gradle-guides/gs-writing-gs-guides (https://github.com/gradle-guides/gs-writing-gs-guides/) and we'll get back to you.

Last updated 2017-02-17 11:27:56 UTC