## Contributing to Apache Camel

There are many ways you can help make Camel a better piece of software - please dive in and help!

- Try surf the documentation - if somethings confusing or not clear, let us know.

- Download the code & try it out and see what you think.

- Browse the source code. Got an itch to scratch, want to tune some operation or add some feature?

- Want to do some hacking on Camel? Try surfing the our [issue tracker](https://issues.apache.org/jira/browse/CAMEL) for open issues or features that need to be implemented, take ownership of an issue and try fix it.

- If you are a new Camel rider and would like to help us, you can also find [some easy to resolve issues.](https://issues.apache.org/jira/issues/?filter=12348074)

- Leave a comment on the issue to let us know you are working on it and add yourself as a watcher to get informed about all modifications.

## Table of Contents

- [Getting in touch](#getting-in-touch)

- [Improving the documentation](#improving-the-documentation)

- [If you find a bug or problem](#if-you-find-a-bug-or-problem)

- [Working on the code](#working-on-the-code)

- [Running checkstyle](#running-checkstyle)

- [Verify Karaf features](#verify-karaf-features)

- [Apache Camel committers should work on the ASF git repo](#apache-camel-committers-should-work-on-the—asf-git-repo)

- [Creating patches](#creating-patches)

- [Pull request at Github](#pull-request-at-github)

- [Manual patch files](#manual-patch-files)

- [Submitting patches](#submitting-patches)

- [Using the issue tracker](#using-the-issue-tracker)

- [Becoming a committer](#becoming-a-committer)

- [More resources](#more-resources)

## Getting in touch

There are various ways of communicating with the Camel community.

- Join us on the [Discussion Forums](https://camel.465427.n5.nabble.com) and take part in any conversations

- Pop by on [Gitter](https://gitter.im/apache/apache-camel) and say hi

## Improving the documentation

Documentation is massively important to help users make the most of Apache Camel and its probably the area that needs the most help!

So if you are interested in helping the documentation effort; whether its just to fix a page here or there, correct a link or even write a tutorial or improve what documentation is already there please do dive in and help!

We moved the documentation into the code (AsciiDoc). We are not using the wiki system anymore.

To edit the documentation:

- It's easy as opening a Pull Request

- You'll find on each component under src/main/docs an .adoc file

- This file contains a static part and a dynamically generated part: the former can be edited directly in the .adoc file, while the latter needs your intervention on the javadoc

- Once you modify the javadoc, you'll need to rebuild the component and the .adoc will be automatically updated

- Create a commit and raise a Pull Request

- If you want to add more documentation check for .adoc in codebase

For more information see [How does the website work](https://camel.apache.org/manual/latest/faq/how-does-the-website-work.html) or [How do I edit the website for more details](https://camel.apache.org/manual/latest/faq/how-do-i-edit-the-website.html).

## If you find a bug or problem

Please raise a new issue in our [issue tracker](https://issues.apache.org/jira/browse/CAMEL)

If you can create a JUnit test case then your issue is more likely to be resolved quicker.

e.g. take a look at some of the existing [unit tests cases](https://svn.apache.org/repos/asf/camel/trunk/camel-core/src/test/java/)

## Working on the code

We recommend to work on the code from [github](https://github.com/apache/camel/).

git clone https://github.com/apache/camel.git

cd camel

Build the project (fast build).

mvn clean install -Pfastinstall

If you intend to work on the code and provide patches and other work you want to submit to the Apache Camel project, then you can fork the project on github and work on your own fork. The custom work you do should be done on branches you create, which can then be committed and pushed upstream, and then submitted to Apache Camel as PRs (pull requests). You can find many resources online how to work on github projects and how to submit work to these projects.

## Running checkstyle

Apache Camel source code is using a coding style/format which can be checked whether is complying using the checkstyle plugin.

To enable source style checking with checkstyle, build Camel with the -Psourcecheck parameter

mvn clean install -Psourcecheck

Please remember to run this check on your code changes before submitting a patch or github PR. You do not need to run this against the entire project, but for example in the modules you work on. Lets say you do some code changes in the camel-ftp component, then you can run the check from within this directory:

cd camel-ftp

mvn clean install -Psourcecheck

## Verify Karaf features

Camel-Karaf lives now in his own repository, so to verify a Karaf feature you'll need to fork the following [repository](https://github.com/apache/camel-karaf).

To check a new Karaf feature or an existing one you should run a verification on the features.xml file. You'll need to follow these steps:

First thing to be done is running a full build of Camel. Then

cd platform/karaf/features/

mvn clean install

If you modified a component/dataformat or updated a dependency in the main camel repository, you'll first need to build the main camel locally and then run a full build of camel-karaf.

## Apache Camel committers should work on the ASF git repo

If you are an Apache Camel committer then clone the ASF git repo at

git clone https://gitbox.apache.org/repos/asf/camel.git

cd camel

or

git clone https://github.com/apache/camel.git

cd camel

Build the project (without testing).

mvn clean install -Dtest=false

PS: You might need to build multiple times (if you get a build error) because sometimes maven fails to download all the files.

Then import the projects into your workspace.

## Creating patches

We recommend you create patches as github PRs which is much easier for us to accept and work with. You do this as any other github project, where you can fork the project, and create a branch where you work on the code, and then commit and push that code to your fork. Then navigate to the Apache Camel github webpage, and you will see that github in the top of the page has a wizard to send your recent work as a PR (pull request).

## Pull request at Github

There is also a Git repository at Github which you could fork. Then you submit patches as any other github project - eg work on a new feature branch and send a pull request. One of the committers then needs to accept your pull request to bring the code to the ASF codebase. After the code has been included into the ASF codebase, you need to close the pull request because we can't do that...

When providing code patches then please include the Camel JIRA ticket number in the commit messages.

We favor using the syntax:

CAMEL-9999: Some message goes here

## Manual patch files

We gladly accept patches if you can find ways to improve, tune or fix Camel in some way.

We recommend using github PRs instead of manual patch files. Especially for bigger patches.

Most IDEs can create nice patches now very easily. e.g. in Eclipse just right click on a file/directory and select Team -> Create Patch. Then just save the patch as a file and attach it to the corresponding issue on our https://issues.apache.org/jira/browse/CAMEL[JIRA issue tracker].

If you're a command line person try the following to create the patch

diff -u Main.java.orig Main.java >> patchfile.txt

or

git diff --no-prefix > patchfile.txt

## Submitting patches

The easiest way to submit a patch is to

- [create a new JIRA issue](https://issues.apache.org/jira/browse/CAMEL) (you will need to register),

- attach the patch or tarball as an attachment (if you create a patch file, but we recommend using github PRs)

- \*\*tick the Patch Attached\*\* button on the issue

We prefer patches has unit tests as well and that these unit tests have proper assertions as well, so remember to replace your system.out or logging with an assertion instead!

## Using the issue tracker

Before you can raise an issue in the [issue tracker](https://issues.apache.org/jira/browse/CAMEL) you need to register with it. This is quick & painless.

## Becoming a committer

Once you've got involved as above, we may well invite you to be a committer. See [How do I become a committer](http://camel.apache.org/how-do-i-become-a-committer.html) for more details.

The first step is contributing to the project; if you want to take that a step forward and become a fellow committer on the project then see the [Committer Guide](https://camel.apache.org/manual/latest/faq/how-do-i-become-a-committer.html)