# 10. Performing the Git configuration

## 10.1. User configuration

You have to configure at least your user and email address to be able to commit to a Git repository because this information is stored in each commit.

## 10.2. Exercise: User configuration

Configure your user and email for Git via the following command.

```
# configure the user which will be used by Git
# this should be not an acronym but your full name
git config --global user.name "Firstname Lastname"

# configure the email address
git config --global user.email "your.email@example.org"
```

## 10.3. Push configuration

If your are using Git in a version below 2.0 you should also execute the following command.

```
# set default so that only the current branch is pushed
git config --global push.default simple
```

This configures Git so that the `git push` command pushes only the active branch (in case it is connected to a remote branch, i.e., configured as remote-tracking branches) to your Git remote repository. As of Git version 2.0 this is the default and therefore it is good practice to configure this behavior.

You learn about the push command in **Section 21.1, “Push changes to another repository”**.

## 10.4. Avoid merge commits for pulling

If you pull in changes from a remote repository, Git by default creates merge commits if you pull in divergent changes. This may not be desired and you can avoid this via the following setting.

```
# set default so that you avoid unnecessary commits
git config --global branch.autosetuprebase always
```

### Note

This setting depends on the individual workflow. Some teams prefer to create merge commits, but the author of this book likes to avoid them.

### 10.5. Color Highlighting

The following commands enables color highlighting for Git in the console.

```
git config --global color.ui auto
```

### 10.6. Setting the default editor

By default Git uses the system default editor which is taken from the *VISUAL* or *EDITOR* environment variables if set. You can configure a different one via the following setting.

```
# setup vim as default editor for Git (Linux)
git config --global core.editor vim
```

### 10.7. Setting the default merge tool

File conflicts might occur in Git during an operation which combines different versions of the same files. In this case the user can directly edit the file to resolve the conflict.

Git allows also to configure a merge tool for solving these conflicts. You have to use third party visual merge tools like tortoisemerge, p4merge, kdiff3 etc. A Google search for these tools help you to install them on your platform. Keep in mind that such tools are not required, you can always edit the files directly in a text editor.

Once you have installed them you can set your selected tool as default merge tool with the following command.

```
# setup kdiff3 as default merge tool (Linux)
git config --global merge.tool kdiff3

# to install it under Ubuntu use
sudo apt-get install kdiff3
```

### 10.8. More settings

All possible Git settings are described under the following link: **git-config manual page**

### 10.9. Query Git settings

To query your Git settings, execute the following command:

```
git config --list
```

If you want to query the global settings you can use the following command.

```
git config --global --list
```