Getting Ready to Submit Your First Pull Request

These instructions provide the step-by-step process to set up your machine to make contributions to the core React Native repository, and create your first pull request.

Prologue: Getting Ready

You will need a few tools and dependencies in order to build and develop for React Native. These are covered as part of the Environment Setup guide under the "Building Projects with Native Code" section.

In order to accept your pull request, we need you to submit a Contributor License Agreement (CLA). You only need to do this once to work on any of Facebook's open source projects. It only takes a minute so you can do it while you wait for your dependencies to install.

Chapter I: Welcome to Open Source

1. Install `git`

The React Native source code is hosted on GitHub. You can interact with the git version control through the `git` command line program. We recommend you follow GitHub's instructions to set up git on your machine.

2. Get the source code

While you can browse the source code for React Native on GitHub, we recommend you set up a fork on your local machine.

1. Go to https://github.com/facebook/react-native.
2. Click on Fork on the upper right.
3. When asked, select your username as the host for this fork.

You will now have a fork of React Native on GitHub at https://github.com/your_username/react-native. Next, you will grab a copy of the source code for your local machine. Open a shell and type the following commands:

```
git clone https://github.com/facebook/react-native.git
git remote add fork https://github.com/your_username/react-native.git
```

> *If the above seems new to you, do not be scared. You can access a shell through the Terminal application on macOS and linux, or PowerShell on Windows.*

A new `react-native/` directory will be created with the contents of the core React Native repository. This directory is actually a clone of the React Native git repository. It is set up with two remotes: `origin` for the upstream https://github.com/facebook/react-native repository, and `fork` for the fork of React Native on your own GitHub account.

3. Create a branch

We recommend creating a new branch in your fork to keep track of your changes:

```
git checkout --branch my_feature_branch --track origin/master
```

Chapter II: Implementing your Changes

1. Make changes to the code

You can now make any changes deemed necessary using your code editor of choice. Visual Studio Code is popular with JavaScript developers. If you're mostly making changes to iOS or Android, using Xcode or Android Studio might provide a nicer integrated experience.

2. Test your changes

Make sure your changes are correct and do not introduce any test failures. You can learn more in Running and Writing Tests.

3. Lint your code

We understand it can take a while to ramp up and get a sense of the style followed for each of the languages in use in the core React Native repository. Developers should not need to worry about minor nits, so whenever possible, we use tools that automate the process of rewriting your code to follow conventions.

For example, we use Prettier to format our JavaScript code. This saves you time and energy as you can let Prettier fix up any formatting issues automatically through its editor integrations, or by manually running `npm run prettier`. We also use a linter to catch styling issues that may exist in your code. You can check the status of your code styling by simply running `npm run lint`.

To learn more about coding conventions, refer to the Coding Style guide.

4. View your changes

Many popular editors integrate with source control in some way. You can also use `git status` and `git diff` on the command line to keep track of what has changed.

Chapter III: Proposing your Changes

1. Commit your changes

Make sure to add your changes to version control using `git`:

```
git add <filename>
git commit -m <message>
```

You can use a short descriptive sentence as your commit message.

> *Worried about writing good git commit messages? Do not fret. Later, when your pull request is merged, all your commits will be squashed into a single commit. It is your pull request description which will be used to populate the message for this squashed commit.*

This guide covers just enough to help you along with your first contribution. GitHub has several resources to help you get started with git:

- Using Git
- The GitHub Flow

2. Push your changes to GitHub

Once your changes have been commited to version control, you can push them to GitHub.

```
git push fork <my_feature_branch>
```

If all goes well, you will see a message encouraging you to open a pull request:

```
remote:
remote: Create a pull request for 'your_feature_branch' on GitHub by visiting:
remote:        https://github.com/your_username/react-native/pull/new/your_feature_branch
remote:
```

Visit the provided URL to proceed to the next step.

3. Create your pull request

You are almost there! The next step is to fill out the pull request. Use a descriptive title that is not too long. Then, make sure to fill out all of the fields provided by the default pull request template:

- **Summary:** Use this field to provide your motivation for sending this pull request. What are you fixing?
- **Changelog:** Help release maintainers write release notes by providing a short description of what will be changed should the pull request get merged.
- **Test Plan:** Let reviewers know how you tested your changes. Did you consider any edge cases? Which steps did you follow to make sure your changes have the desired effect? See What is a Test Plan? to learn more.

4. Review and address feedback

Keep an eye on any comments and review feedback left on your pull request on GitHub. Maintainers will do their best to provide constructive, actionable feedback to help get your changes ready to be merged into the core React Native repository.