

# Possible Onboarding Resources in Selected Projects

## 1. Vue.js

### 1.1. Main Page

#### About

- Link to External Website
- Link to Code of Conduct

#### README.md

- Documentation: Link to External Website and Link to Live Examples (also in the External Website)
- Stay In Touch: Link to Blog
- Contribution: Link to Contributing Guide (CONTRIBUTING.md)

### 1.2. .github Directory

#### CONTRIBUTING.md

- Link to Contributor Code of Conduct (CODE\_OF\_CONDUCT.md)
- Issue Reporting Guidelines
  - Bug Report
  - Feature Request
- Pull Request Guidelines
- Development Setup
  - Link to Git Commit Message Convention (.github/COMMIT\_CONVENTION.md)
- Project Structure
  - Link to Explanation of Build Files (dist/README.md)

#### CODE\_OF\_CONDUCT.md

#### COMMIT\_CONVENTION.md

#### PULL\_REQUEST\_TEMPLATE.md

#### ISSUE\_TEMPLATE

### 1.3. dist Directory

**README.md:** Explanation of Build Files

## 1.4. External Website

Can be a good starting point to understand what the project itself has to offer to users, and what are the current features. There are many explanations in guides, examples, APIs, documentation, and so on for possible users of the technology.

Has an extensive Community Guide including the following information [additional to the ones in the directory] useful for new contributors:

- Get Support tab including Community Channels and Meetup Links
  - Vue Forum
  - Discord
  - DEV Community
- External resources related to Vue
- Link to the "RFC" (request for comments) repository
- A list of "What you can do as a contributor":
  - Help other users via community channels
  - Help triage issues (verifying the validity, reproducing, finding cause etc.)
  - contribute code (link to contributing.md and RFC repo)
  - share experience
  - translate docs
  - Become community leader

## 1.5. Wiki

- Contributor Code of Conduct
- Governance Document

## 1.6. Discord Channel

- Channels in many different languages for non-native contributors to discuss and help each other.
- Channel to share useful links
- Discussion channels for different purposes including best-practices
- Channels to get help on tools and libraries
- Channels for code help
- Meetups & events & job finding related to Vue

Note: I found the community channels mostly built for users, not contributors, so I'm leaving it here.

# 2. Tensorflow

## 2.1. Main Page

### About

- Link to External Website
- Link to Code of Conduct

### **Code of Conduct (CODE\_OF\_CONDUCT.md)**

- Responsibilities
- Conflict Resolution
- Reporting Violations

### **Contributing Guidelines (CONTRIBUTING.md)**

- Contribution guidelines and standards
  - Reference to GitHub How to create Pull Request page
  - Reference to the “Good first issue examples” page
  - Reference to the “Issues with contributions welcome title” page
  - Reference to the “API Review Practices” page
  - Link to external website “Contribute tests” tab
  - License and build example
- Coding Style for Different Languages
- Running Unit Tests
- Debug Builds
- Link to “Contributor License Agreement (CLA)” and information about CLAs

### **ISSUES.md**

- GitHub issue policy

### **ISSUE\_TEMPLATE.md**

- Template and guideline for creating issues
- StackOverflow community link for help and support

### **LICENSE.md**

### **README.md**

- Links to External Website [tools, libraries, community, contribution] for users
- Links to coding language APIs
- Installation guide [+Link to more detailed guide]
- Contribution Guidelines
  - Links to Google Groups page and StackOverflow for questions and discussions
  - References to CONTRIBUTING.md, CODE\_OF\_CONDUCT.md, Issues tab
- Tensorflow Resources

## **2.2. External Website**

[Same] Can be a good starting point to understand what the project itself has to offer to users, and what are the current features. There are many explanations in guides, examples, APIs, documentation, and so on for possible users of the technology.

The community tab has quality resources for new contributors:

- Join: Explains the ways to get involved and stay informed.
  - Links to forum, groups, RFC Process, Contributor Guide and Bug Reporting, Social Media Accounts
- Blog
- Forum
- Groups
- Contribute
  - Help for the “First Contribution”
  - Community Values

- Detailed guidelines and rules about contributing code, tests, documentation, or community.
- Links to external resources on used technologies (e.g. Git)
- Explanation on RFC process

## 3. Laravel

### 3.1. Main Page

#### About

- Link to External Website

#### README.md

- Link to Contribution Guide in the external website
- Information for new users

### 3.2. External Website

The documentation has a lot of information for users like the previous projects.

#### Contribution Guide

- Bug Reports
- Support Questions (Community channels, forums etc.)
- Core Development Discussion
- Which Branch?
- Compiled Assets
- Security Vulnerabilities
- Coding Style
  - PHPDoc
  - StyleCI
- Code of Conduct

## 4. Ruby on Rails

### 4.1. Main Page

Each directory has its own .md file that also has an external link to the detailed guide about that topic.

#### About

- Link to External Website
- Link to Code of Conduct

#### Code of Conduct (CODE\_OF\_CONDUCT.md)

- Link to External Website

#### RELEASING\_RAILS.md

- Detailed guide on the release steps

#### Contributing Guidelines (CONTRIBUTING.md)

- A list of possible ways of contributing
- Links to contribution guide and security policy in external website

- References to issues & bug\_report\_templates directory

#### **README.md**

- Information about the technology and installation guide [+Link to more detailed guide]
- Links to contribution guide and other resources on the external website

## 4.2. guides/source Directory

Has all the guide pages on the external website, including the contribution guide.

## 4.3. Contribution Guide

Can be found at the External Website or in guides/source/contributing\_to\_ruby\_on\_rails.md

- How to use GitHub to report issues.
- How to clone main and run the test suite.
- How to help resolve existing issues.
- How to contribute to the Ruby on Rails documentation.
- How to contribute to the Ruby on Rails code.
- Links to some resources:
  - Python
  - git-cherry-pick documentation
  - Ignoring Files (Git documentation)
  - Rails API
  - Bug report templates
  - Security Policy
  - VS Code remote containers plugin.
  - rails-dev-box
  - Local Development guide
  - yarn link documentation
  - RuboCop
  - codespell project

# 5. Scrappy

## 5.1. Main Page

#### **About**

- Link to External Website
- Link to Code of Conduct

#### **README.rst, CONTRIBUTING.md, INSTALL.md**

- Links to external website for documentation, installation, contribution etc.

#### **CODE\_OF\_CONDUCT.md**

- Information about Contributor Covenant Code of Conduct

## 5.2. docs Directory

Has the documentation presented at the external website.

#### **README.rst**

- Quick guide on how to compile the Scrappy documentation.

#### **versioning.rst**

- Information about Versioning and API stability

### **topics directory**

- Has all the documentation under different topics (as .rst files)

### **contributing.rst**

- The ways to contribute to the project:
  - Blog about Scrapy. Tell the world how you're using Scrapy. This will help newcomers with more examples.
  - Report bugs and request features in the issue tracker.
  - Submit patches for new functionalities and/or bug fixes.
  - Join the Scrapy subreddit and share your ideas on how to improve Scrapy.
  - Answer Scrapy questions at Stack Overflow.
- Examples, rules and guidelines for different topics
- Links to forums and community channels [Reddit, StackOverflow]
- Link to GitHub docs
- Link to “PEP 8 – Style Guide for Python Code”

### **faq.rst**

- Frequently Asked Questions

### **intro directory**

- Has example projects, installation guide, overview and tutorial as .rst files. Mostly for the new users of the project but can be useful for new contributors to discover the project further as well.

## **5.3. sep Directory**

- Has Scrapy Enhancement Proposals, which can be good examples for a contributor who wished to propose an enhancement to the project.

## **5.4. .github/ISSUE\_TEMPLATE Directory**

- Has bug report and feature request templates with explanations.

## **5.5. Wiki**

- Links for new users directing to the Scrapy external website.
- Developer Documentation links:
  - Possible use cases
  - Release Procedure

## **5.6. External Website**

- Documentation
  - Installation Guide
  - Tutorial
  - How to get help
  - Examples
  - FAQs
  - Built-in Services
  - Basic & Advanced Concepts
  - Extensions

- Solving Specific Problems
  - Contribution Guide (mentioned in **contributing.rst** )
- Community Channels
- Curated Resources
  - Articles & Blog Posts
  - Videos
  - Courses
  - Books
  - Slides

# A List of Possible Resources for New Contributors

(One may involve the others)

- Contribution Guide
- ReadMe file
- Issue Templates: Bug Report Template, Feature Request Template
- Code of Conduct
- Releasing Guide
- Example issues tagged for new contributors like "Good-first-issues" or "welcome"
- Wiki pages
- Installation Guide, Build Examples
- Resources (APIs, courses, style guides etc.) for used coding languages
- Resources for any technology will be used during development.
- License statements
- Code Review Convention Statements
- Conflict Resolution Convention Statements
- Resources for using GitHub features (pull requests, forking repos etc.)