The Developer Goodie Bag

August 21, 2019

Carsten Jacobsen
Open Source Developer Advocate
@CarstenJacobsen

Uber
Open Source @ Uber

350+ Repositories on GitHub

2,000+ Contributors worldwide

117K+ Stars on GitHub. Average 365 per repository

5 Community projects: Jaeger, Horovod, Pyro, Apache Hudi, Kepler.gl

2 2 of the top open source projects on InfoWorld's awards list
What is the Developer Goodie Bag?

- It’s a metaphor for the resources you provide developers to help them adopt your project.

- It’s the resources that will enable a smooth developer journey and developer experience.
Developer Journey:

The journey, or the steps you take, from you initially discovering a project to successfully adopting it into your project.
Steps in the Developer Journey

1. Explore
2. Getting Started
3. Experience Features
4. Integration

If the developer gets stuck on any of those four steps, there’s a high risk the developer will not adopt a project.
Let’s look at some examples of what we should put in our Developer Goodie Bag to make sure we support the developers through the journey.
An open source project’s GitHub repository is often one of the first places developers go to learn more about the project.

It’s the project’s landing page.
So, what should be in the READ.ME file?
A good approach is to answer the following questions:

- What is it, and why does it matter?
- What are the most important features and how does it work?
- How do I install it?
- How do I use it?
- Can you help me test it?
- Now, how do I deploy it?
What is it?

Project Title

One Paragraph of project description goes here.

- What is the key value proposition for the project?
- What are the benefits using this project?
- Have a logo? Add it to this section!
- Add any badges you think make sense. You can find an overview on shields.io. We suggest to add: build status, code coverage, and dependency status.
- How does it look like in action? Have any pictures, gifs, videos, etc?
Features

Introduction

This is a high level description of the project. Describe what the project is for, what it is doing and which problem it solves. This should not be long, usually 2-3 lines is good, keep it short and precise.

Highlight some features:

- **Topic in bold.** Longer description about the topic, but try to keep it short.
- **Lorem ipsum dolor sit amet.** Consectetur adipiscing elit integer eleifend congue justo.
- **Nunc ac libero vulputate.** Feugiat lectus nec, lobortis turpis. Praesent maximus, elit at finibus rhoncus.

Donec arcu eros, malesuada convallis accumsan eu, laoreet ac felis. Donec quis libero nec est finibus eleifend nec non odio. Vestibulum feugiat sem urna, sit amet pharetra augue feugiat et.
Installation

Step by step guidelines to get a copy of the project up and running locally.

Prerequisites

What things you need to install the software and how to install them.

$ install required software and libraries

Mauris cursus molestie mauris, porttitor eleifend sem. In ut vehicula tellus.

Install

A step-by-step guide to installing the project, including necessary configuration etc.

$ download/clone/yarn/npm project
$ build/make project
$ environment config etc.

How to use

Usage

Maecenas eleifend diam turpis, et fringilla libero laoreet at. Donec quis pretium lectus. Fusce dignissim ante at odio ultricies, ac consequat nibh pelentesque.

Example 1

Usage examples can be commands

```bash
$ do something
```

Example 2

It can be code:

```javascript
const btn = document.getElementById('button');
btn.addEventListener('click', function() {
    alert('Hello World!');
});
```

Nam et nisl aliquam, tempus magna eget, feugiat nunc. Ut quam leo, venenatis eget urna vitae, aliquet semper sem. Vestibulum at odio nisl. Donec arcu eros, malesuada convallis accumsan eu, laoreet ac felis.
How to use

Documentation

The documentation, including method/API references, can be found here.

Documentation can also exist in the README file, but generally it's not recommended to do so, unless the documentation is very limited.

someFunction

```typescript
const token: Token = someFunction(name: string);
```

- name: string - a human-readable name for the token
- returns token: Token

anotherFunction

```typescript
register(token: Token, value: any);
```

- token: Token - a Token created via someFunction
- value: any - a configuration value
- returns undefined
READ.ME File

How to test

Testing

Test instructions, and test scripts, can be found here.

Alternatively, test instructions can exist in the README file.

$ run test

Test specific feature

$ run test 1

How to deploy

Deployment

Add notes about how to deploy this on a live system

1. Donec ac vehicula elit, vel tempor dui.
2. Vivamus ac ipsum vitae diam viverra maximus.
3. Donec magna orci, convallis eu nulla et, laoreet pulvinar nulla.

Integer maximus justo quis varius malesuada. Interdum et malesuada fames ac ante ipsum primis in faucibus. Phasellus urna ipsum, pellentesque at est non, tempor fringilla leo.

ENV:

- ASDF: 1
- QWE: 123
- ZXC: zxc

Curabitur malesuada aliquet massa, non porttitor diam. Nunc vitae risus tincidunt nisl congue elementum vitae id ex. Vivamus ultricies, nisl quis cursus aliquet, tortor lorem maximus lorem, sit amet lacinia ante ipsum ac ligula.

# First run this command to ...
somecmd --qwe=123 asd.qw
# Then run this command...
othercmd qwe asd

Interdum et malesuada fames ac ante ipsum primis in faucibus.
... and the rest

**Built With**

{(References)}

**Contributing**

Please read our code of conduct before you contribute! You can find detailed requests in the CONTRIBUTING.md file. Issue template.

**Versioning**

We document versions and changes in our changelog - see the CHANGELOG.md file.

**License**

This project is licensed under the {LICENSE} License - see the LICENSE.md file for details.

**Acknowledgments**

- Hat tip to anyone who's code was used
- Inspiration
- etc

**References**

- Phasellus luctus, libero tincidunt blandit blandit
- Nullam in eros in nunc dignissim tempus
- Vivamus faucibus orci ac consequat venenatis
Getting Started
Getting Started

The Getting Started section should go through the process of doing a minimalistic installation or implementation.
Getting Started

At this point we have provided guidance, which enables developers to get hands on experience with the project.
Tutorials
Tutorials

The most common tutorial is Hello World!

It’s super simple, and it walks through the steps from installation to creating some kind of output.

Good tutorials will show, step-by-step, how to do basic things with the project, which can easily be replicated by the developer.
The code is revealed as the tutorial explains the steps. This breaks larger code blocks down to smaller and more chunks, which are easier to comprehend.
At this point, the developer has replicated a series of steps, a working application, and a basic understanding of how things work.
Sample Code
In contrast to the tutorials, sample code doesn’t walk through a typical use case by breaking the code down to small chunks with detailed explanations.

Sample code is complete projects, which shows best practice of implementation of selected features.
Sample code is typically independent projects on GitHub.
Reference
My guess is that a lack of good documentation is a Top 3 reason why developers choose not to adopt a project.
There are many opinions on how good documentation is structured, and what it contains…

… and many of them are right.
The reference can look very different, depending on the project type.
Sandbox
Is there anything you can do to have developers testing your project faster?
Examples

- Sandbox environment
- Docker containers
- Demo accounts

Think about ways a developer can experience the project without installing or configuring a demo environment.
Support
Is there any way to get in contact with the project maintainer?
Recap

Make sure to fill up your developer goodie bag, it makes it easier for developers to evaluate your project, which drives adoption.

- READ.ME File
- Getting Started
- Tutorials
- Sample Code
- Reference
- Sandbox
- Support
Thank you

Carsten Jacobsen
Open Source Developer Advocate, Uber
carstenj@uber.com
@CarstenJacobsen

opensource.uber.com