JOHNS HOPKINS
SHERIDAN LIBRARIES | Open Source Programs Office

OSPO Explainer Transcript: README Basics

URL:

Slide 1: Hello, and welcome to the Johns Hopkins University Open Source Programs Office Explainer Series, a collection of bite-sized videos tackling a variety of open-source essentials. New topics are added regularly, check out ospo.library.jhu.edu to see what’s coming up, or drop us a line at ospo@jhu.edu to make a request. Today’s explainer? README Basics.

Slide 2: A README serves as the introduction to your software, and provides users with important information on how to get started.

Slide 3: READMEs aren't just for open source projects. Any code that will be shared publicly or privately can benefit from a README.

Slide 4: Add a README to your repository to help others understand the purpose of your code, why it was created, what it does, and how it works; provide clear instructions on installation, usage, and dependencies; and encourage collaboration by providing clear guidelines for potential contributors.

Slide 5: The OSPO has a great checklist in GitHub for writing a README. You can copy it directly to your repository and fill in the blanks. Visit https://bit.ly/jhuospo_readme. You can click through the following slides to learn what information to gather for your README.

Slide 6: First, help your readers identify your project. Create a file named README, using either plain text or a markup language. At the top of the file, make sure the project's name and logo are the first heading under the project name or linked from it. Add a URL for the project such as a repository or homepage URL. Underneath the project name, clearly identify the owner or author of the project. If your project uses a notice file to identify copyright or authorship, you should point to that file here.

Slide 7: Next, help the reader evaluate your project. Describe your project in terms of what it does or achieves - not how. This is not the place to talk about languages, technologies or tools, that comes later. You can also identify the WHO here if a project was built for or is used by a specific audience. If you're having a hard time you can try a "MadLib” approach to get started. With “project name,” you can “verb,” “plural noun.” Or, if you use “project” then you can ... Also use this space to describe who may use the project and under what terms. For open source software, describe and then name the license.

Slide 8: Next, you want to help the reader use your project. If applicable, you can list your project’s prerequisites. List the steps to install and use the project one time. You want to help your reader go from having your project's files to using the project for the first time. These instructions should stop once the project works once – extended usage instructions really belong in your full documentation. Always make sure you test your installation and setup steps. If you have somebody outside the project, you can test even better and give you honest feedback. There's nothing to write down here for your README. Just make sure that what you've written works.

Slide 9: Next, you want to help your reader engage with your project. First, tell them where to go for more project documentation. This might include your website documentation files, README, or companion files like your license or code of conduct. You want to tell your readers where to go for help. This might include descriptions and links to mailing lists, issue trackers or forums. You can also tell your audience how they can help. If it's open source software, you can link to and summarize the project's contributors guide or describe how and where you want contributors to add to your project. You can also describe how and where bugs should be reported.

Slide 10: For projects with more than one repository, create a README for the project. You can choose one repository to be the primary and place the core README there, or if your repository platform supports it, you can place a README at the project or organization level, create a separate README in each repository that points to the primary project README and describes in more detail the purpose of that specific component.

Slide 11: And for final checks, if your README is long, add a table of contents after your project description. If your README is *very* long, you might move content to other docs. Set a reminder to review your README on a regular basis. This can be annually, when you have a major release, or some other milestone.

Slide 12: The Hopkins OSPO would like to thank Daniel Beck for creating the READEM checklist and dedicating it to the public domain.

Slide 13: Have any questions about what you've just learned? Ask the JHU Open Source Programs Office for more information. You can reach out to us online at ospo.library.jhu.edu or send an email to ospo@jhu.edu.