

# Tim McMackin

timothymcmackin@gmail.com  
linkedin.com/in/timothymcmackin  
timothymcmack.in

|   |   |   |
|---|---|---|
| Summary   | <p>I simplify complicated software suites by providing thorough, useful information that guides customers through their tasks.</p> <p>I work directly with developers and technology by implementing information systems and researching computer systems in a hands-on manner.</p> <p>I lead teams that provide complete information services for software products.</p> |   |
| Experience  | <b>Shutterstock (Remote): Information Engineer IV</b>   | <b>2022</b>                                 |
|   | <b>Senior Information Engineer</b>  | <b>2017-2022</b>                            |
| <ul style="list-style-type: none"><li>- Drove a culture of information sharing and introduced technical documentation processes as Shutterstock's first technical writer. Hosted writing classes and working sessions to guide teams to make documenting their work part of their processes.</li><li>- Developed and wrote documentation for <a href="#">Shutterstock's public API, SDK, CLI</a>, and <a href="#">UI widgets</a>, using a pipeline to merge OpenAPI spec files with manually written examples and task-oriented content. Reduced average partner integration time from 3 to 1.2 months. Partner feedback indicates that documentation is a major factor in getting integrations working quickly.</li><li>- Developed and maintained a modular documentation container that internal teams use to publish documentation. Coded the container, deployment code, and tests using Gatsby, GraphQL, React, GitHub Actions, asynchronous JavaScript, and other open-source tools. Onboarded documentation for 5 internal services in 2 months and eventually 13 systems.</li><li>- Documented Node.JS services and APIs to promote innersourcing and accelerate onboarding.</li></ul> |   |   |
| <b>IBM (Durham, NC): Advisory Technical Writer/Team Lead</b>  |   |   |
|   |   | <b>2004-2017</b>                            |
| <ul style="list-style-type: none"><li>- Led a seven-person agile team providing documentation for IBM UrbanCode software and its 70-person development team, including documentation, education, videos, white papers, customer forum response, and support response. Covered a wide range of realistic customer DevOps scenarios, such as blue-green deployments, rolling deployments, dark launches, continuous delivery, and mainframe deployments.</li><li>- Documented a wide range of technologies and platforms, including programming languages, distributed software, mainframe software, diagramming tools, and cloud systems.</li><li>- Researched and wrote custom documents for our largest customers and for critical needs. Contributed directly to sales deals in the millions of dollars.</li><li>- Collaborated with support teams to identify customer needs and to focus efforts on preventing support calls. Reduced average time to documentation error resolution from 8 days to 2 days.</li></ul>   |   |   |
| <b>Additional experience</b>  |   |   |
| <ul style="list-style-type: none"><li>- Produced print and email communications and competitive reports for an industrial advertising agency</li><li>- Managing API documentation for a financial information services start-up</li></ul>   |   |   |
| Education   | <b>M.A. Professional Writing</b>  | <b>B.A. English Rhetoric and Literature</b> |
|   | Carnegie Mellon University, Pittsburgh, PA  | Xavier University, Cincinnati, OH           |

Tim McMackin

timothymcmackin@gmail.com

linkedin.com/in/timothymcmackin

timothymcmack.in

## Publication samples

### [Shutterstock API, SDK, and CLI reference](#)

I originally built this API reference documentation with Widdershins and Shins but later wrote a custom Node.js and Nunjucks program to generate it with the help of the API development team. It merges handwritten examples and task-based information about using the API with generated examples and endpoint reference from an OpenAPI source file.

By providing clear and thorough information for partners using the Shutterstock API, these reference docs reduced the average partner integration time from 3 months to 1.2 months.

### [Shutterstock developer portal documentation](#)

This documentation covers many of the same topics as the API reference but is aimed at decision-makers and architects. Along with information about using the API, JavaScript SDK, and CLI, I've added a tutorial and information that helps Shutterstock's enterprise clients allow their employees to access Shutterstock through single sign-on. I didn't build this platform, but I expanded it as the content grew, and I wrote most of the content.

### [Tutorial: Using the computer vision features of the Shutterstock API](#)

This tutorial covers using the reverse image search and keyword suggestion features of our computer vision services. It includes end-to-end instructions on setting up a free API account, installing the Shutterstock JavaScript SDK, and writing straightforward Node.js programs to analyze user-submitted images. I researched and wrote all of the content.

### [Tutorial: Licensing and downloading images with the Shutterstock free API subscription](#)

This tutorial covers the end-to-end process of searching, licensing, and downloading images with a free API subscription, requiring no credit card or other prerequisites. It covers what the API can be used for, how our partners use it, and how it provides licensed images. The tutorial includes instructions for Postman, cURL, the Shutterstock CLI, and the Shutterstock JavaScript SDK. I researched and wrote all of the content.

### [Performance characteristics of IBM UrbanCode Deploy](#)

I planned, wrote, and typeset this white paper based on testing data from our product development teams. I consulted with the performance testers about what the test results meant and with the customer-facing teams that needed to explain the performance of the product to current and potential customers.

### [Rollback scenarios in IBM UrbanCode Deploy](#)

This video addresses customer confusion that I learned about from support requests. It demonstrates techniques for reversing problems automatically, and it teaches customers about a common misconception about the product at the same time. Our internal support call analysis showed significantly fewer customer issues in this area after I delivered and promoted the video. I researched, wrote, and recorded the video.