

# Tim McMackin

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

Summary	I write technical documentation that simplifies complicated software, APIs, and SDKs. I code sample applications and implement docs-as-code pipelines. I lead teams of technical writers and work directly with development teams.	
Experience	TriliTech (Remote):	Senior Technical Writer 2023–present
	<ul style="list-style-type: none"><li>– Updating the developer documentation for the <a href="#">Tezos blockchain</a>, including API and SDK reference, tutorials, and developer-focused tasks.</li><li>– Documenting the <a href="#">Tezos Etherlink Ethereum compatibility layer</a> and its Solidity kernel.</li><li>– Restructuring and clarifying the documentation for the <a href="#">SmartPy</a>, JsLIGO, and CameLIGO smart contract programming languages.</li><li>– Documenting the <a href="#">Tezos SDK for Unity</a> video game development tooling.</li></ul>	
	Shutterstock (Remote):	Staff Information Engineer IV 2022
		Senior Information Engineer 2017–2022
	<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, UI widgets</a>, and <a href="#">AI tools</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>	
	IBM (Durham, NC):	Advisory Technical Writer/Team Lead 2004–2017
	<ul style="list-style-type: none"><li>– Led a seven-person team providing documentation for IBM's UrbanCode continuous delivery software, including reference docs, live training, videos, white papers, customer forum response, and support response. Covered a wide range of enterprise DevOps scenarios, such as blue-green deployments, rolling deployments, dark launches, and continuous delivery.</li><li>– Documented a wide range of technologies and platforms in Markdown and DITA, including programming languages, distributed software, mainframe software, and cloud systems.</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>	
Education	M.A. Professional and Technical Writing Carnegie Mellon University, Pittsburgh, PA	B.A. English Rhetoric and Literature Xavier University, Cincinnati, OH

Tim McMackin

timothymcmackin@gmail.com

linkedin.com/in/timothymcmackin

timothymcmack.in

**Publication  
samples**

[SmartPy language documentation](#)

This documentation for the SmartPy smart contract programming language needed improvements to make it clearer to programmers coming from more traditional programming environments. I did a full reorganization and rewrite of the language reference manual.

[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.

[Documentation for the Tezos SDK for Unity](#)

This documentation provides the quickest possible route for experienced Unity game developers to use Tezos as their backend. It starts with a quickstart and leads users into task-based tech docs and reference for the SDK objects. 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.