W3C WebAuthn Adoption Community Group

Practical WebAuthn

Technical Design Document

Summary

For the purposes of helping promote education, adoption, and implementation of the Web Authentication standard, the WebAuthn Adoption Community Group plans to develop a website that can act as a repository of wireframe implementations of the standard, as well as hosting educational information, such as common questions and pitfalls, about developing with WebAuthn.This document aims to outline the architecture and design of the website, the GitHub repository for code related to the project, and describe some of the technical problems and solutions for implementation that may be involved. This project and site are titled Practical WebAuthn.

User Stories

There are two, perhaps overlapping, types of users that we are targeting for the initial release of Practical WebAuthn: Developers and Contributors. The former are likely other web application developers looking for more information and code resources for implementing Web Authentication or FIDO2, while the latter are also developers, but interested in contributing to Practical WebAuthn by providing code, images, or feedback to the three core services that make up the project: The Github repository, the web host and potentially the authentication service, and the site itself, which is hosted on the GitHub repository and served via the host.

Developers

Developers interact with Practical WebAuthn by visiting the Practical WebAuthn site. Through the site, they are directed to view a working wireframe of WebAuthn code implemented in a framework of their choosing. They are able to create and assert ownership of credentials in the example and, either through a console view or on the page itself, see the JS objects created and steps that are being done. Developers should be able to select different authentication options for testing, such as only allowing platform or roaming authenticators, but not be forced outright to choose.

An FAQs section can also be included to answer common questions about best practices for implementing WebAuthn, which types of authenticators are supported by attestation type, and more.

After seeing WebAuthn in action through the wireframe example of their choosing, they are directed to either go to the associated GitHub repository to pull down the associated code or to seek further information from Practical WebAuthn's educational resource section.

Contributors

Contributors may already be well aware of the Practical WebAuthn site and wish to provide additional features or new frameworks, and visit the Github repository to create issues and pull requests. If the user is creating a pull request for existing frameworks, the code should go through unit and integration testing and human review. If the contributor is providing a new framework, it should include its own unit and integration testing and adhere to some degree of coverage.

Contributors should be able to know if the code they are submitting will conform to the guidelines of the repository and group not just by passing integration testing, but also through contribution documents (such as a standard CONTRIBUTING.md document) that can help developers produce conformant code from the outset.

After going through review with the community group and being discussed at the biweekly meeting, the code is merged to the master branch of the repository (or potentially a milestone branch) and is deployed to the web host.

The GitHub Repository

The code repository, broken up into two branches, for educations resources and materials in one branch and all

Requirements

Must have a robust and transparent testing framework for contributing. Ideally we can identify a CI-based testing apparatus through which all demos can be run for a degree of unified “compliance” with an agreed-upon standard (aim for FIDO conformance testing? That’ll need massive refactor to make it usable in CI/CD pipelines though…).

Organization

Main Branch

The Github Repository should be organized by language and framework, with each framework containing the server-side webauthn code needed for that specific resource to operate.

Main Branch

The structure should conform to the following example:

Main Branch
Java
↳ <Specifc Java Framework>
↳ README.md
build
test
src
Ruby
↳ README.md
bin
lib
test
Rakefile
webauthn.gemspec
<Other Language Frameworks>
↳ <Specific Frameworks>
↳ README.md
<Language/Framework specific
Client
↳ README.md

webauthn.js
Testkit
↳ README.md
<webauthn conformance markup>

Site Branch
webcode!

Contribution

Contributors to the github repository can submit pull requests

The Authentication Service and Web Host

Managed in part by The FIDO Alliance and the server for Practical WebAuthn.

Authentication Service Options

In order to support the many different frameworks that implement webauthn on the site, there should be a generic service for verifying authentication of the credentials that users create.

Locally Hosted (Chosen Route as per meeting)

In this option, requests and responses are handled locally. Such demos should be provided to a project-wide step-by-step set of instructions for pulling a demo down to the developer's or contributor's computer and then starting it to serve the client over HTTPS with routing to the included backend instance.

This option trades the need for attestations and assertions to be handled by an actual remote server to offer a more authentic FIDO2/WebAuthn experience.

Back End Hosted

In this configuration, backend infrastructure would be required to host an "immutable" WebAuthn-capable service from which attestation options could be requested, and to which attestations and assertions could be sent. The hosted service would be "immutable" in the

sense that all registered authenticators would be cleared out after a period of time as they are only for demo purposes.

This would enable the possibility of demos that focus solely on the client-side experience. Compliance to instructions for hosting such “front-end only” demos over HTTPS would still be needed.

The Practical WebAuthn Site

Organization

The Practical WebAuhn site should be a simple way to both test and learn more about the Web Authentication standard. The site has a small footprint, but is meant to be a bastion site for developers to learn more about WebAuthn and add it to their applications. With that in mind, the site should be broken into at least three pages:

Landing Page

The landing page should have an introduction to the site and its purpose, as well as a brief explanation of the Web Authentication specification. This page should also describe the other two sections of the site and potentially provide direct links to the GitHub repository, the W3C specification, and The FIDO Alliance as well.

Debugger and Inspector

The Debugger and Inspector should be a similar tool to those on other WebAuthn sites, where developers can craft and then execute WebAuthn requests in the browser. After executing, developers should be able to inspect the PublicKeyCredential response in the window, as well as know if the request was valid and, if not, where it was malformed.

Resources and Testing

This page should link to the Practical WebAuthn github, as well as potentially answer some of the frequently asked questions about WebAuthn and FIDO2. Developers should be able to quickly link to specific resources for the language that they’re interested in.