

codes+stock

Authentication Using OpenID Connect and OAuth2

Mark A. Wilson

Senior Developer

MarkW@LogicalAdvantage.com

www.DeveloperInfra.com

@DeveloperInfra



Mark A. Wilson

www.developerinfra.com

@DeveloperInfra

Software craftsman,
consultant and agile
.NET/JavaScript web
developer. User group
leader and event planner.

Loving husband, dog
foster, and Disney
aficionado.



Logical Advantage

www.logicaladvantage.com

Our mission is to partner with our clients to provide strategies and solutions that maximize the ROI of Enterprise Asset Intelligence and Human Capital Development.



Authentication Using Tokens for AngularJS, OWIN, ASP.NET Web API & Identity



AngularJS Authentication Demo

The screenshot shows a web browser window titled "AngularJS Authentication" with the URL "ngauthenticationweb.azurewebsites.net/#/home". The page has a blue header bar with "Home" on the left and "Login" and "Sign Up" on the right. Below the header, the main content area features a large title "AngularJS Authentication" and a descriptive paragraph about the application's purpose. It then splits into two sections: "Login" and "Sign Up". Each section contains a button labeled "Login »" or "Sign Up »". At the bottom of the page, there are footer links for "Created by Taiseer Joudeh" and "Taiseer Joudeh Blog".

AngularJS Authentication

AngularJS Application which uses OAuth Bearer Token for authentication and implements Refresh Tokens. The backend API is built using ASP.NET Web API 2, OWIN middleware, and ASP.NET Identity.

Login

If you have Username and Password, you can use the button below to access the secured content using a token.

Sign Up

Use the button below to create Username and Password to access the secured content using a token.

Created by Taiseer Joudeh. Twitter: [@tjoudeh](#)

Taiseer Joudeh Blog: [bitoftech.net](#)

- <http://bitoftech.net/2014/06/01/token-based-authentication-asp-net-web-api-2-owin-asp-net-identity/>

Open Web Interface for .NET

The OWIN logo consists of the letters "OWIN" in a bold, sans-serif font. The letters are a light blue color and have a thick, rounded appearance.

Open Web Interface for .NET

- A *specification*, not a framework
- A standard for an interface between .NET web applications and web servers
- Decouple the application from the [IIS] server
- Encourage development of simple and lightweight modules
- Project Katana is OWIN implementations for Microsoft servers and frameworks

- https://en.wikipedia.org/wiki/Open_Web_Interface_for_.NET

Project Katana

- A set of OWIN components built by Microsoft
- Enables ASP.NET applications to be flexible, portable, lightweight, and provide better performance
- Host – An executable process
- Server – Opens network sockets
- Middleware – Pipeline of OWIN components
- Application – Your code

Application

Middleware

Server

Host

- <http://www.asp.net/aspnet/overview/owin-and-katana/an-overview-of-project-katana>

OAuth 2.0



- Authorization framework
- Began in November 2006 when Blaine Cook was developing the Twitter OpenID implementation
- OAuth 1.0 – April 2010
- OAuth 2.0 – October 2012
- Allows *access tokens* be issued to third-party clients/websites
- Not without its critics and controversy

- <https://en.wikipedia.org/wiki/OAuth>

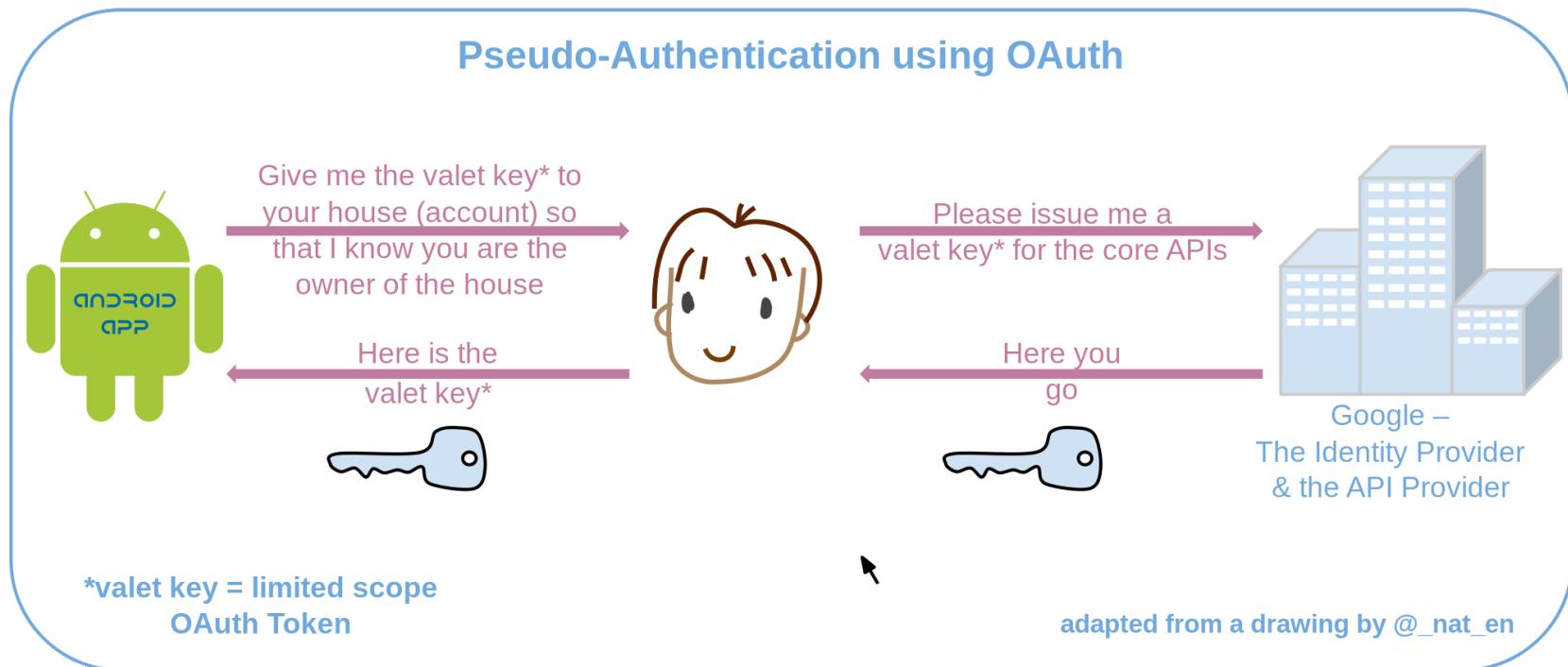
OAuth 2.0



- Does not support signature, encryption, channel binding, or client verification.
- Relies completely on Transport Layer Security (TLS) / Secure Sockets Layer (SSL).
- “Covert Redirect” involving the `redirect_uri`.
- Phishing site attacks.
- Eran Hammer resigned as lead author in July 2012.
- “OAuth 2.0 is not an authentication protocol.” ↗

- <https://en.wikipedia.org/wiki/OAuth>

Pseudo-Authentication using OAuth



- <https://en.wikipedia.org/wiki/OAuth>

Project Katana

"vNext is the successor to Katana (which is why they look so similar). Katana was the beginning of the break away from System.Web and to more modular components for the web stack."

– David Fowler
ASP.NET Core Architect



- <http://forums.asp.net/t/2004299.aspx?Katana%20VS%20vNext>

Is it worth it?

Pros

- Authentication & authorization abstracted into a “component” using standard protocol (OAuth2)
- Eliminated the cookie and Cross-Site Request Forgery (CSRF) problems for SPAs
- Encrypted and signed tokens (using shared machine key)
- SSL by default
- Resource owner flow very easy.

Cons

- Exponentially complex
- Adding refresh tokens required implementing persistence
- Implicit flow (native or JS apps) lacked login/consent view engine
- Lacking in validation
- The ASP.NET team decided to discontinue Project Katana to focus on consuming tokens
- “Note: This outline should not be intended to be used for creating a secure production app.” ↗

- <https://leastprivilege.com/2014/03/24/the-web-api-v2-oauth2-authorization-server-middleware-is-it-worth-it/>

Modern Applications

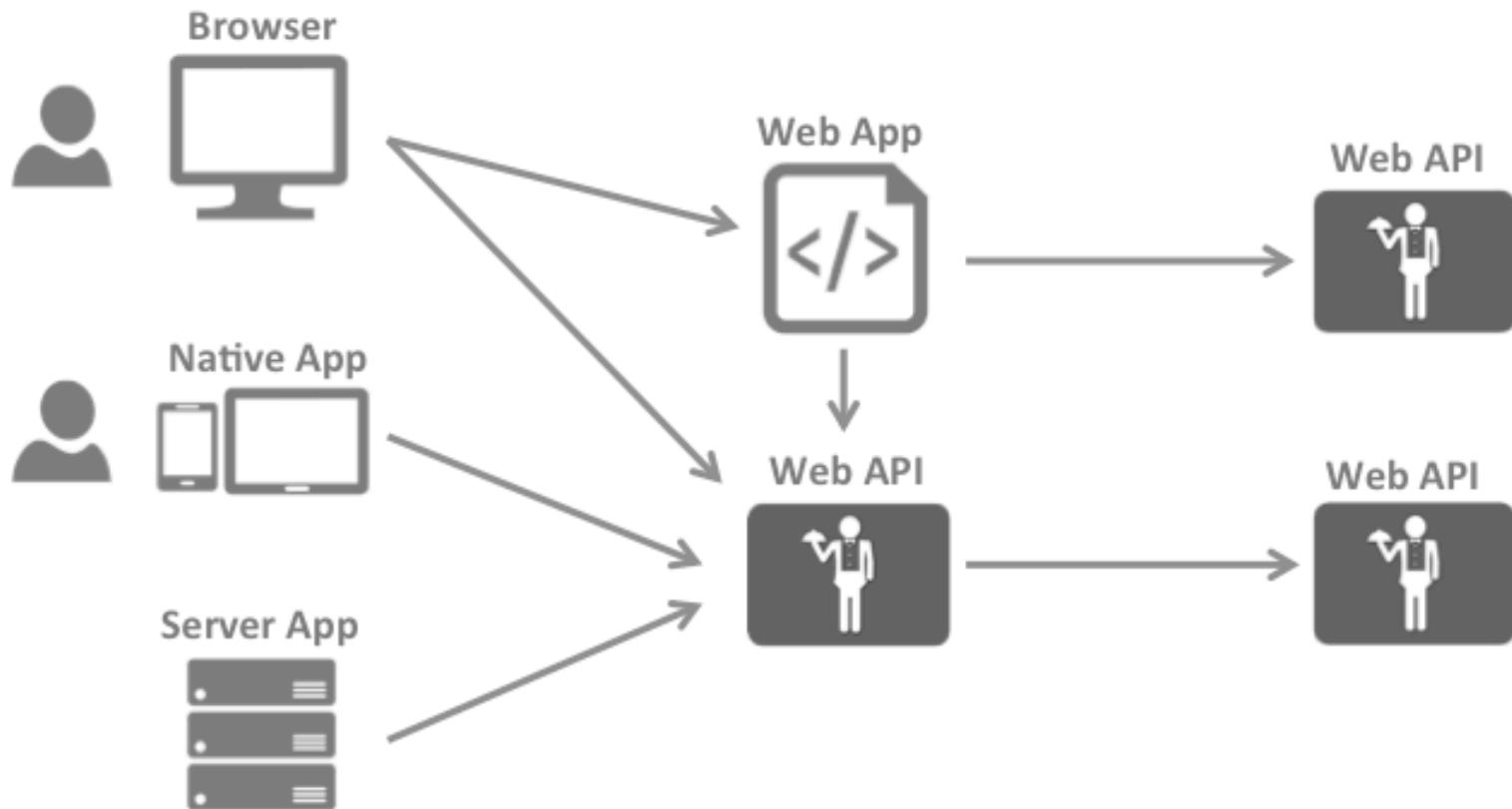


This happened...

No SOAP
No SAML
No WS*
No Windows
No Enterprise

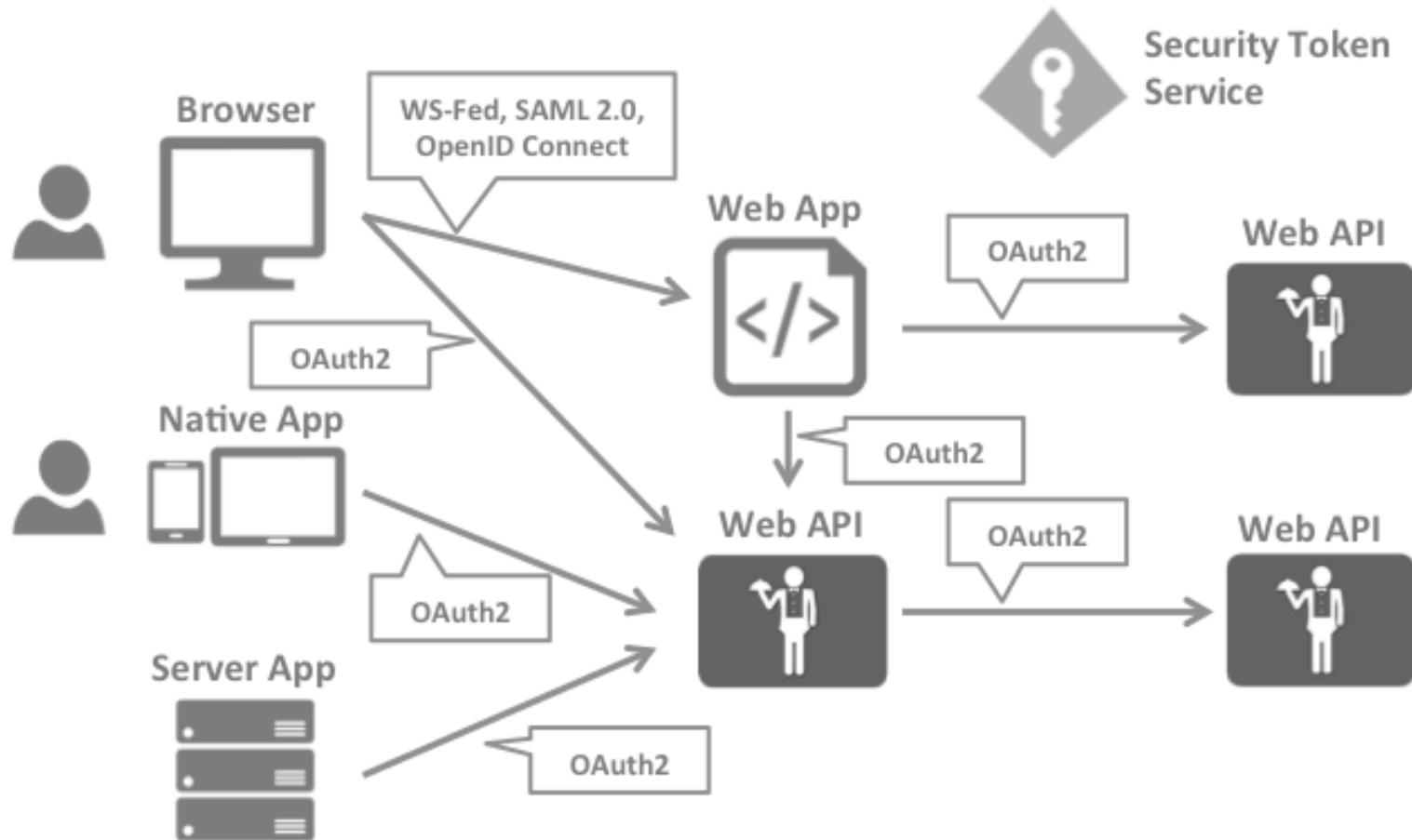


The Big Picture



- <https://identityserver.github.io/Documentation/docsv2/overview/bigPicture.html>

Security Protocols



- <https://identityserver.github.io/Documentation/docsv2/overview/bigPicture.html>

Security Concerns

Authentication



Identity of the current user

API Access



Application identity and/or delegating the user's identity

- <https://identityserver.github.io/Documentation/docsv2/overview/bigPicture.html>

Security Assertion Markup Language



- Most popular & widely deployed
- SAML 1.0 – November 2002
- SAML 1.1 – September 2003
- SAML 2.0 – March 2005
- Browser single sign-on (SSO)
- Problematic beyond intranet
- XML-based
- “Enterprisey” and “not trivial”

- <https://en.wikipedia.org/wiki/Security Assertion Markup Language>

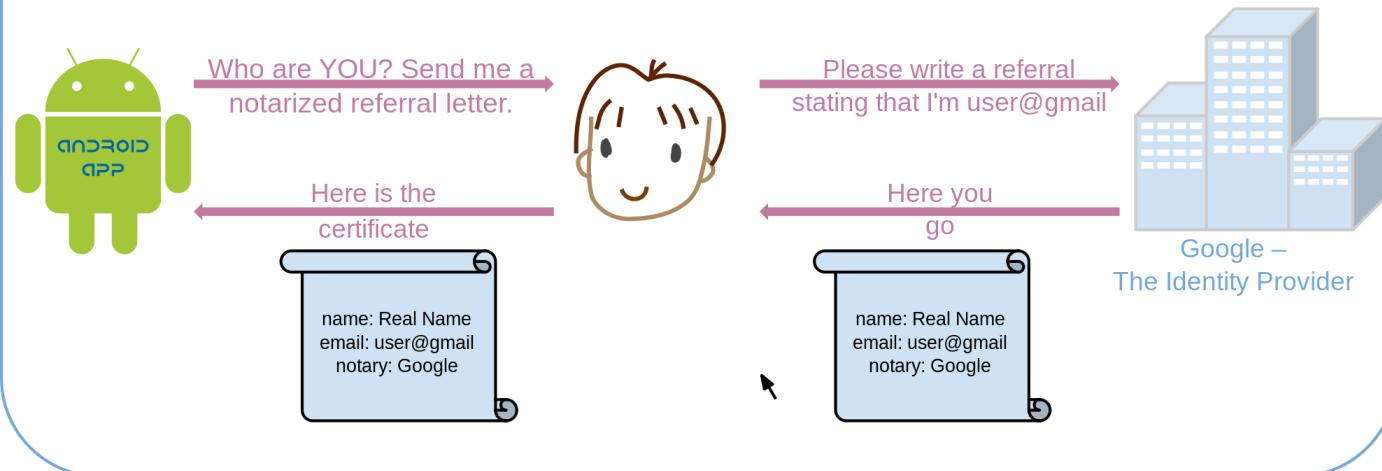
OpenID Connect



- Newest but generally considered to be the future
- OpenID – May 2005
- OpenID 2.0 – December 2007
- OIDC [3.0] – February 2014
- Designed to be more usable by native and mobile applications
- RESTful HTTP API using JSON
- Authentication layer on top of the OAuth 2.0 authorization framework
- Uses the JSON Web Token (JWT) and JSON Object Signing and Encryption (JOSE) specs

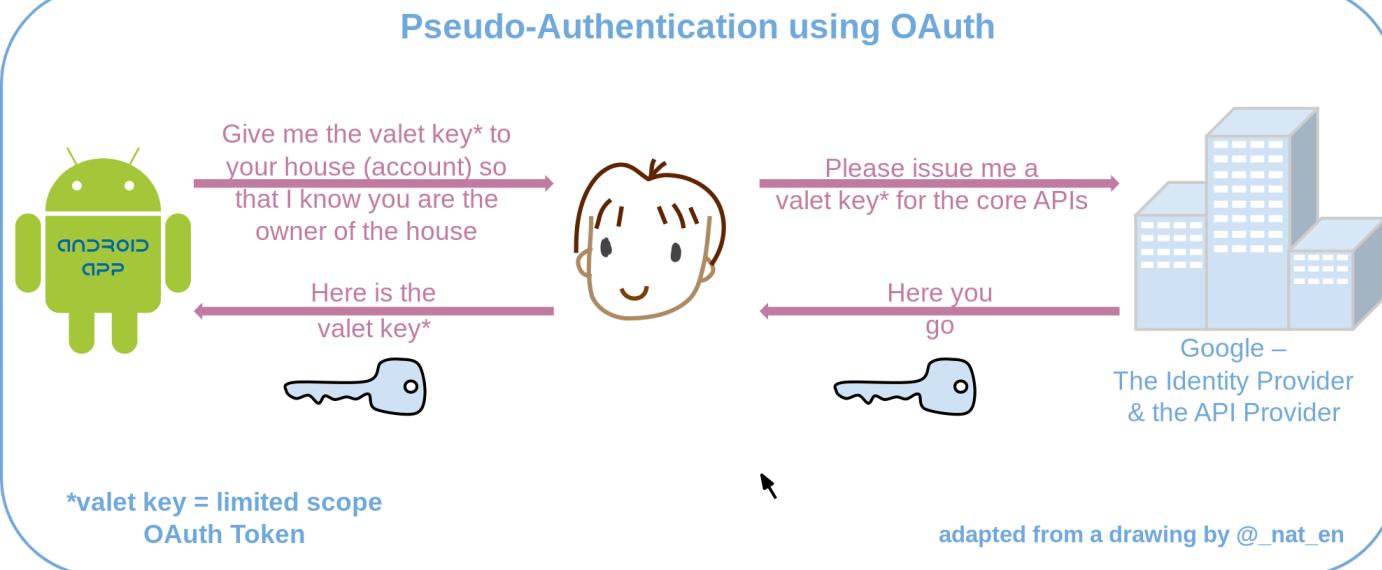
- <https://en.wikipedia.org/wiki/OpenID>

OpenID Authentication



VS.

Pseudo-Authentication using OAuth



adapted from a drawing by @_nat_en

- <https://en.wikipedia.org/wiki/OAuth>

Better Together



- <https://identityserver.github.io/Documentation/docsv2/overview/bigPicture.html>

Authentication Using OpenID Connect and OAuth2



OpenID Connect

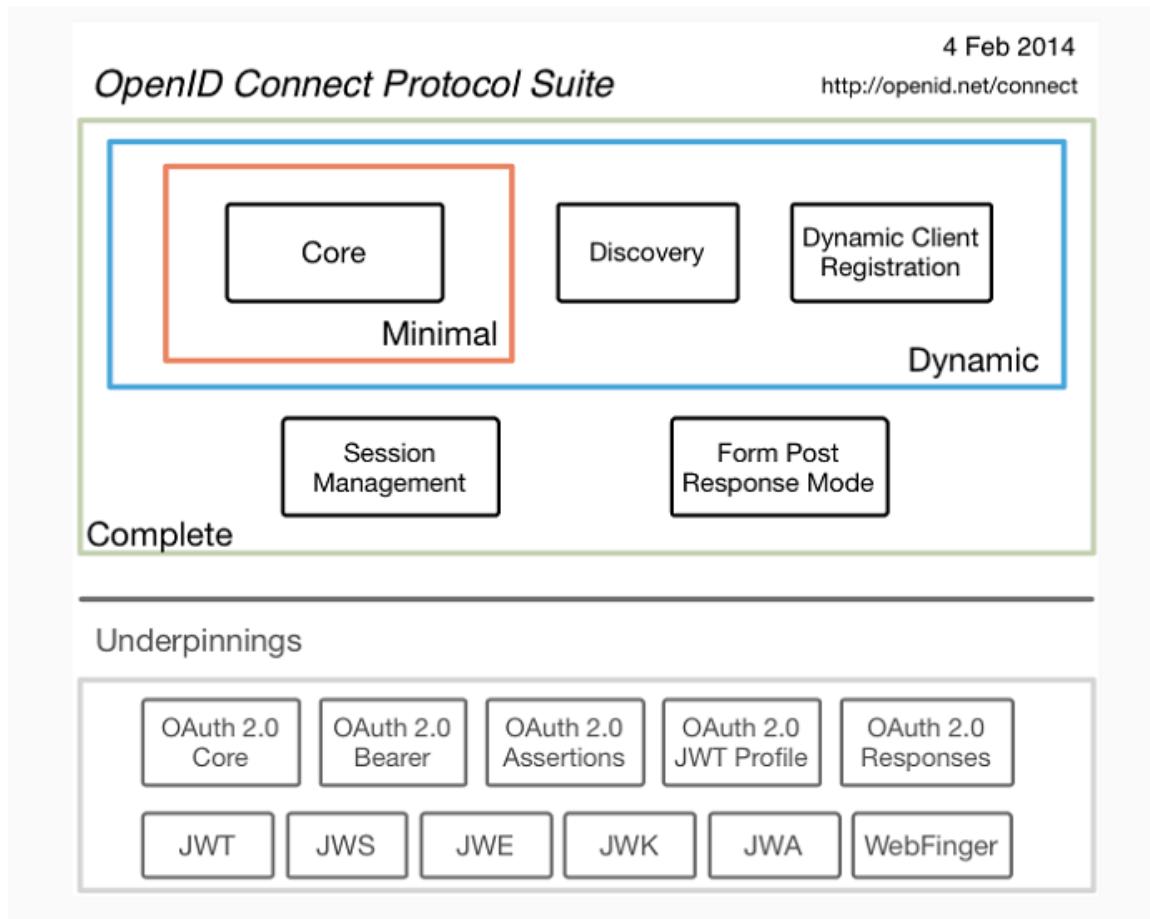


“OpenID Connect 1.0 is a simple identity layer on top of the OAuth 2.0 protocol.”

- Defines identity tokens
- Defines standard token type
- Defines standard cryptography
- Defines validation procedures
- Defines flows for browser, native, and server-based apps
- Combines authentication with short/long-lived delegated API access

- <http://openid.net/connect/>

OpenID Connect



- <http://openid.net/connect/>

OpenID Connect

**Authorize
Endpoint**



**Token
Endpoint**

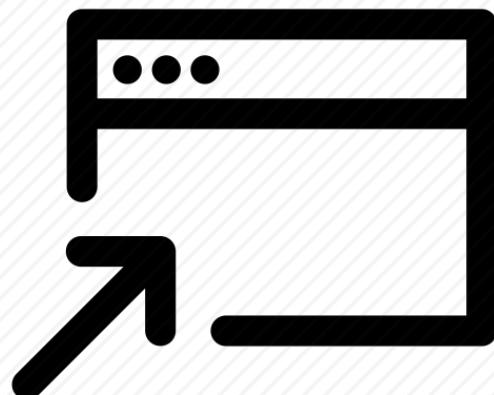


**Userinfo
Endpoint**

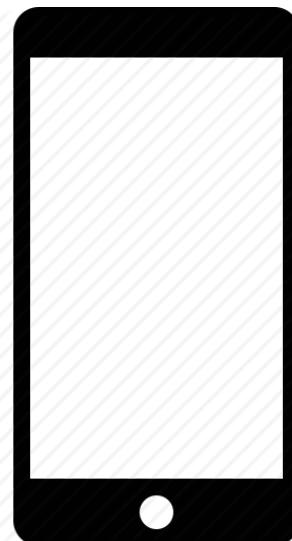


Flows

Implicit Flow
(Browser-based)



Hybrid Flow
(Native/Mobile)



Client Credentials Flow
(Server-to-Server/IoT)



Implicit Flow

GET /authorize

?**client_id**=app1

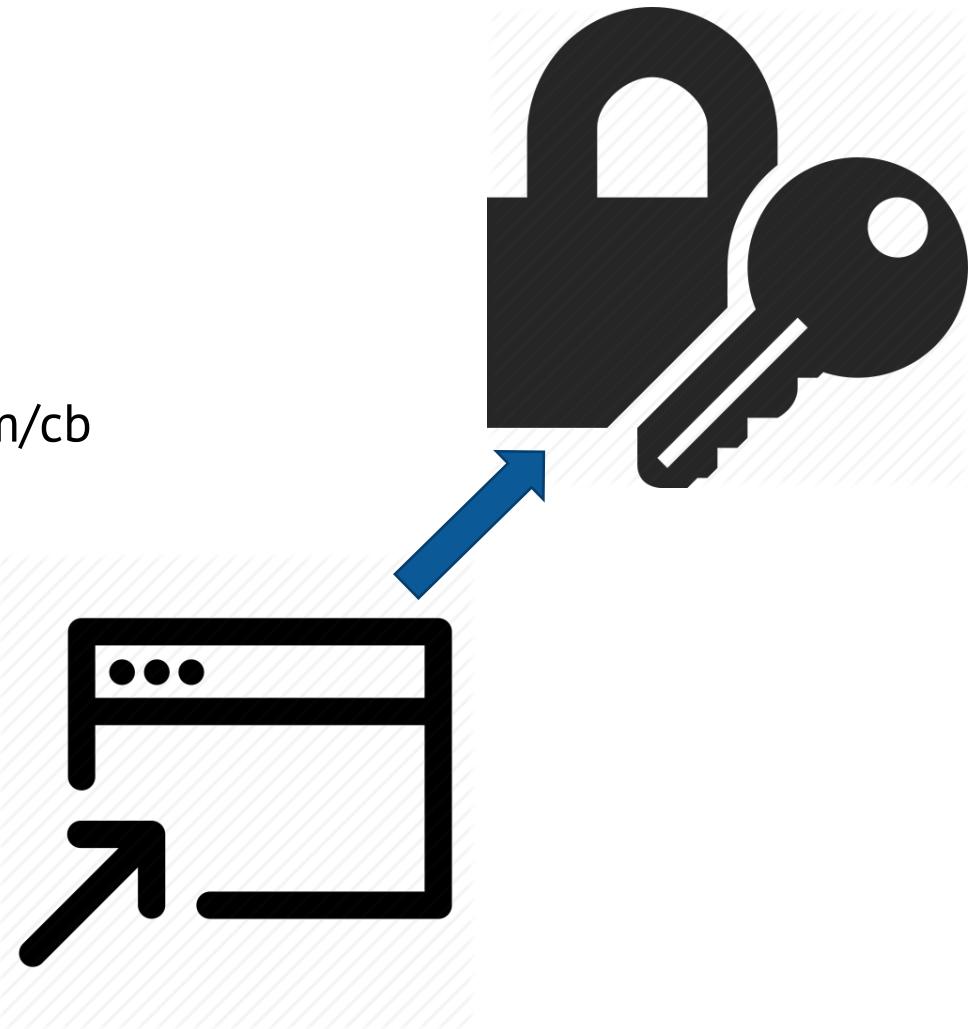
&**redirect_uri**=https://app.com/cb

&**response_type**=id_token

&**response_mode**=form_post

&**nonce**=a1b...x9z

&**scope**=openid email



Identity Token

The screenshot shows the jwt.io website interface. On the left, under the 'Encoded' section, is a large text input containing a JWT token: `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvG4gRG91IiwiYWRtaW4iOnRydWV9.TJVA950rM7E2cBab30RMHrHDcEfxjoYZgeFONFh7HgQ`. On the right, under the 'Decoded' section, are three tabs: 'HEADER: ALGORITHM & TOKEN TYPE', 'PAYLOAD: DATA', and 'VERIFY SIGNATURE'. The 'HEADER' tab shows the JSON object:

```
{ "alg": "HS256", "typ": "JWT" }
```

. The 'PAYLOAD' tab shows the JSON object:

```
{ "sub": "1234567890", "name": "John Doe", "admin": true }
```

. The 'VERIFY SIGNATURE' tab contains the HMACSHA256 verification code:

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  secret  
)
```

, with a note '(secret base64 encoded)' and a redacted 'secret' field. At the bottom of the page is a blue button with a checkmark icon and the text 'Signature Verified'.

- <https://jwt.io/>

Getting Started

The screenshot shows the homepage of the OpenID Connect | OpenID website. The URL in the address bar is openid.net/connect/. The page features a navigation bar with links to 'OpenID Foundation', 'Current Working Groups', 'Specs & Dev Info', 'OpenID® Certification', and 'OpenID Connect FAQ and Q&As'. Below the navigation, there is a section titled 'News!' with a note about the certification program launch. A large section titled 'What is OpenID Connect?' provides an overview of the protocol. Another section, 'How is OpenID Connect different than OpenID 2.0?', compares the two. A 'Specification Organization' section lists the documents that make up the OpenID Connect 1.0 specification. On the right side of the page, there is a sidebar with a search bar and a link to 'FAQ'.

News!

The certification program for OpenID Connect was launched on April 22, 2015. Google, Microsoft, Ping Identity, ForgeRock, Nomura Research Institute, and PayPal OpenID Connect deployments were the first to self-certify conformance.

What is OpenID Connect?

OpenID Connect 1.0 is a simple identity layer on top of the OAuth 2.0 protocol. It allows Clients to verify the identity of the End-User based on the authentication performed by an Authorization Server, as well as to obtain basic profile information about the End-User in an interoperable and REST-like manner.

OpenID Connect allows clients of all types, including Web-based, mobile, and JavaScript clients, to request and receive information about authenticated sessions and end-users. The specification suite is extensible, allowing participants to use optional features such as encryption of identity data, discovery of OpenID Providers, and session management, when it makes sense for them.

See <http://openid.net/connect/faq/> for a set of answers to Frequently Asked Questions about OpenID Connect.

How is OpenID Connect different than OpenID 2.0?

OpenID Connect performs many of the same tasks as OpenID 2.0, but does so in a way that is API-friendly, and usable by native and mobile applications. OpenID Connect defines optional mechanisms for robust signing and encryption. Whereas integration of OAuth 1.0a and OpenID 2.0 required an extension, in OpenID Connect, OAuth 2.0 capabilities are integrated with the protocol itself.

Specification Organization

The OpenID Connect 1.0 specification consists of these documents:

- **Core** – Defines the core OpenID Connect functionality: authentication built on top of OAuth 2.0 and the use of Claims to communicate information about the End-User
- **Discovery** – (Optional) Defines how Clients dynamically discover information about OpenID Providers
- **Dynamic Registration** – (Optional) Defines how clients dynamically register with OpenID Providers
- **OAuth 2.0 Multiple Response Types** – Defines several specific new OAuth 2.0 response types
- **OAuth 2.0 Form Post Response Mode** – (Optional) Defines how to return OAuth 2.0 Authorization Response parameters (including OpenID Connect Authentication Response parameters) using HTML form values that are auto-submitted by the User Agent using HTTP POST
- **Session Management** – (Optional) Defines how to manage OpenID Connect sessions, including postMessage-based logout functionality
- **Front-Channel Logout** – (Optional) Defines a front-channel logout mechanism that does not use an OP iframe on RP pages
- **Back-Channel Logout** – (Optional) Defines a logout mechanism that uses back-channel communication between the OP and RPs being logged out

- <http://openid.net/connect/>

Getting Started

The screenshot shows a web browser displaying the OpenID website at openid.net/developers/libraries/. The page title is "Libraries, Products, and Tools". The main content area includes a table of contents for OpenID Connect, a detailed description of OpenID Connect 1.0, and a sidebar with search, news archives, categories, and recent posts sections.

OpenID®

The Internet Identity Layer

OpenID Foundation ▾ Current Working Groups ▾ Specs & Dev Info ▾ OpenID® Certification ▾

OpenID Connect FAQ and Q&As

Home » Developers » Libraries, Products, and Tools

Libraries, Products, and Tools

Below is a list of libraries, products, and tools implementing current OpenID specifications and related specs. While several of these implementations have been tested, they are maintained by members of the OpenID community or vendors and are not necessarily known to work. Please review the documentation and test your own implementation thoroughly before releasing to the public.

To discuss these implementations, please consider joining the [code@openid.net mailing list](mailto:code@openid.net). To participate in interop testing, also join the [openid-connect-interop@googlegroups.com mailing list](mailto:openid-connect-interop@googlegroups.com).

Table of Contents

- [OpenID Connect](#)
- [JWT/JWS/JWE/JWK/JWA](#)
- [Obsolete Specifications](#)
- [Additions](#)

OpenID Connect 1.0

OpenID Connect is an interoperable authentication protocol based on the OAuth 2.0 family of specifications. It uses straightforward REST/JSON message flows with a design goal of “making simple things simple and complicated things possible”. It’s uniquely easy for developers to integrate, compared to any preceding Identity protocol.

Search for:

Search

News Archives

News Archives

Select Month

Categories

Categories

Select Category

Recent Posts

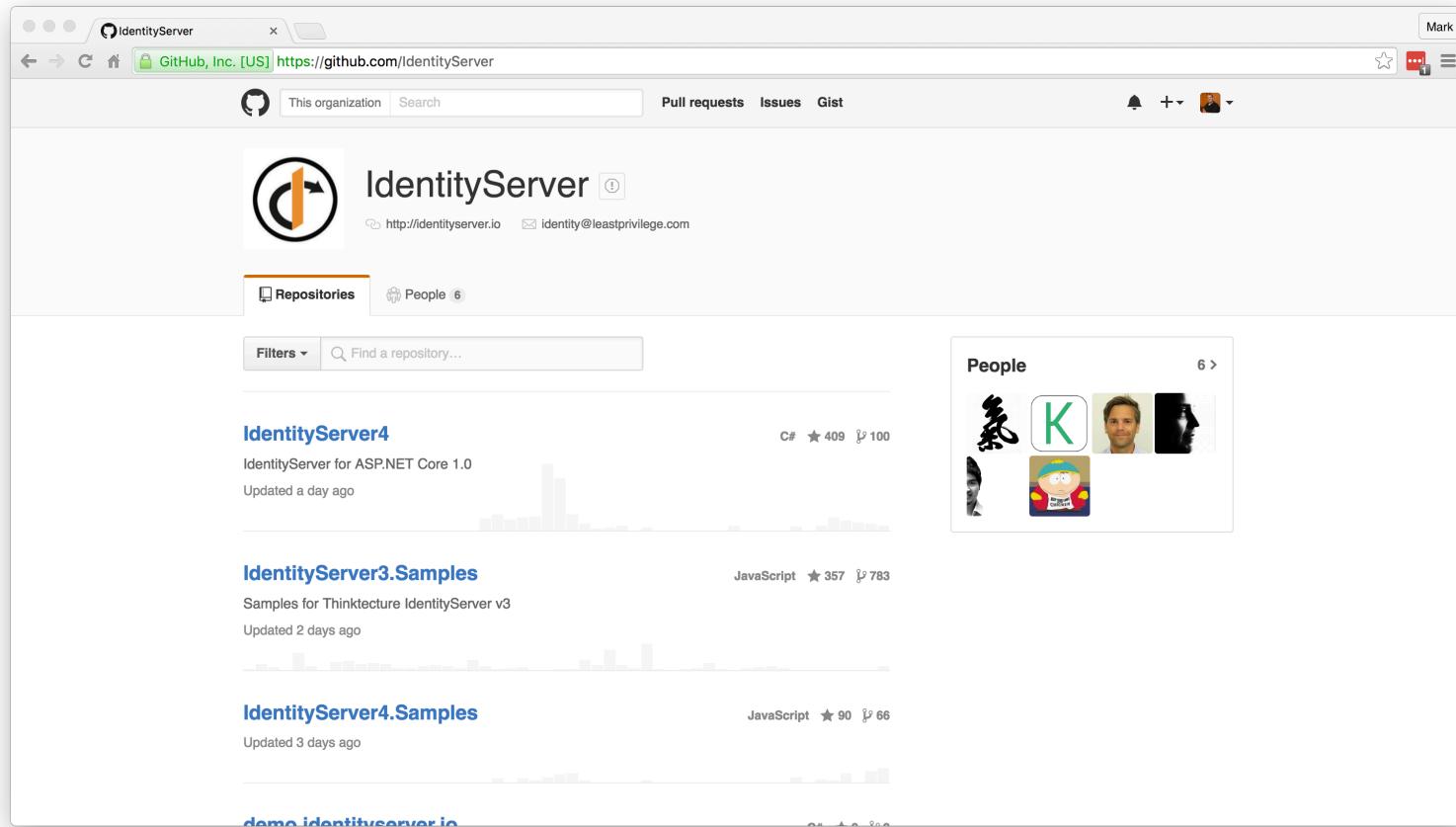
- › Announcing the Financial API (FAPI) Working Group
- › HEART Implementer's Drafts Approved
- › Vote Early and Often!

- <http://openid.net/developers/libraries/>

IdentityServer

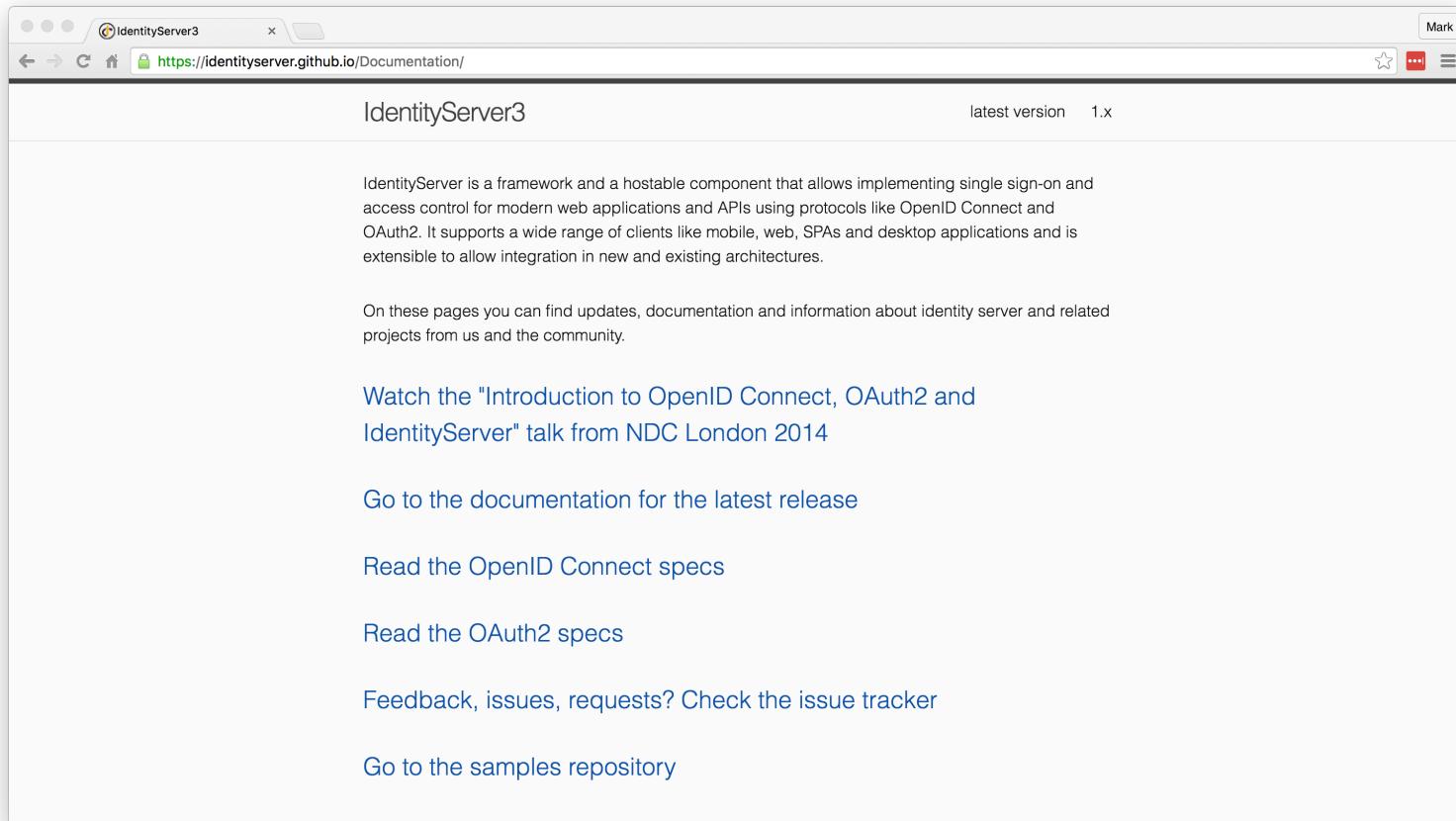


Getting Started



- <https://github.com/IdentityServer>

Getting Started



The screenshot shows a web browser window with the title bar "IdentityServer3". The address bar contains the URL <https://identityserver.github.io/Documentation/>. The main content area displays the IdentityServer3 documentation page. At the top, it says "IdentityServer3" and "latest version 1.x". Below this, a paragraph describes IdentityServer as a framework for implementing single sign-on and access control. It supports various clients and is extensible. A note below states that updates and information can be found on the site. Further down, links are provided for "Introduction to OpenID Connect, OAuth2 and IdentityServer" (a talk from NDC London 2014), documentation for the latest release, OpenID Connect specs, OAuth2 specs, the issue tracker, and the samples repository.

IdentityServer3

latest version 1.x

IdentityServer is a framework and a hostable component that allows implementing single sign-on and access control for modern web applications and APIs using protocols like OpenID Connect and OAuth2. It supports a wide range of clients like mobile, web, SPAs and desktop applications and is extensible to allow integration in new and existing architectures.

On these pages you can find updates, documentation and information about identity server and related projects from us and the community.

Watch the "Introduction to OpenID Connect, OAuth2 and IdentityServer" talk from NDC London 2014

[Go to the documentation for the latest release](#)

[Read the OpenID Connect specs](#)

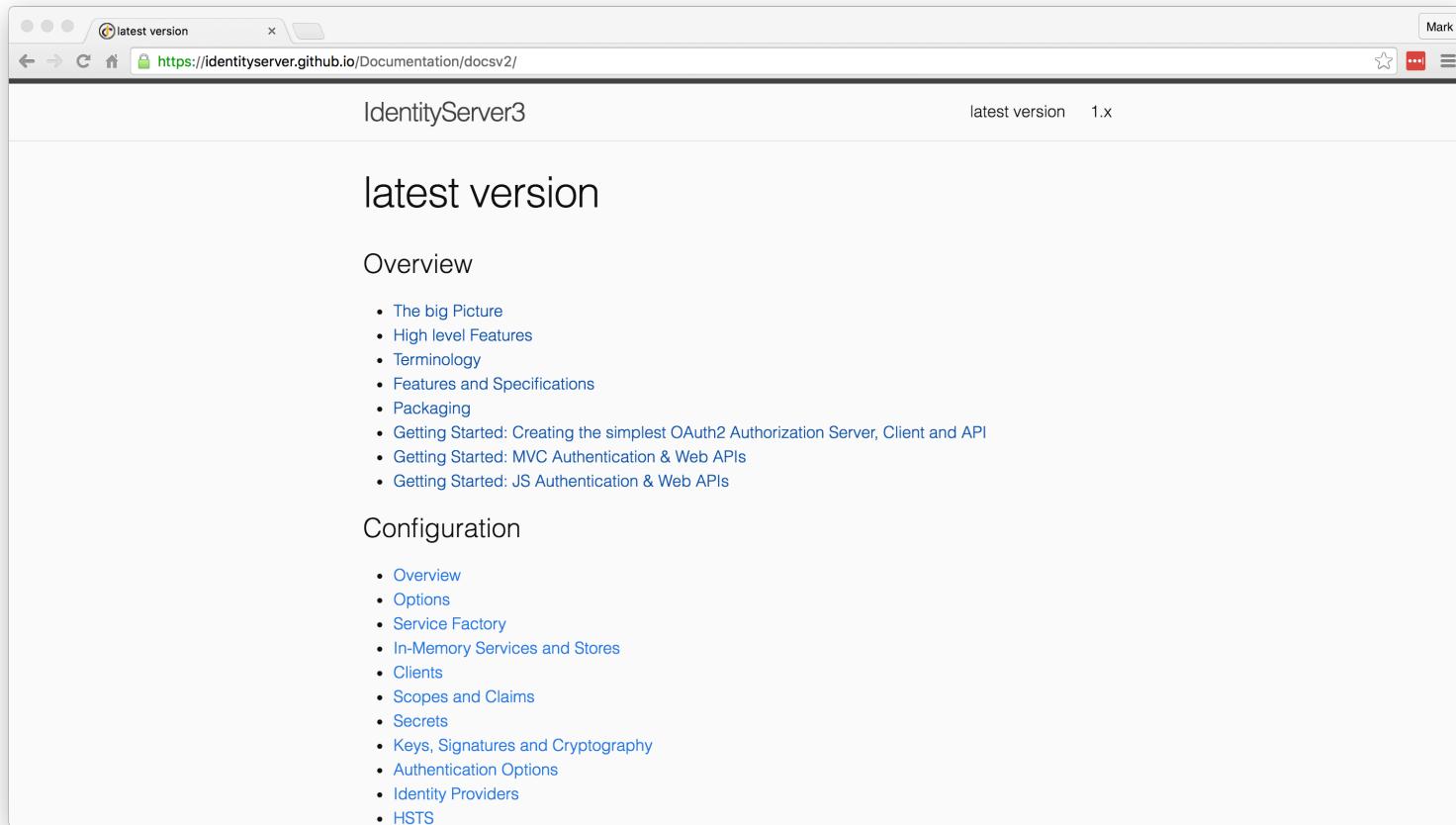
[Read the OAuth2 specs](#)

[Feedback, issues, requests? Check the issue tracker](#)

[Go to the samples repository](#)

- <https://identityserver.github.io/Documentation/>

Getting Started

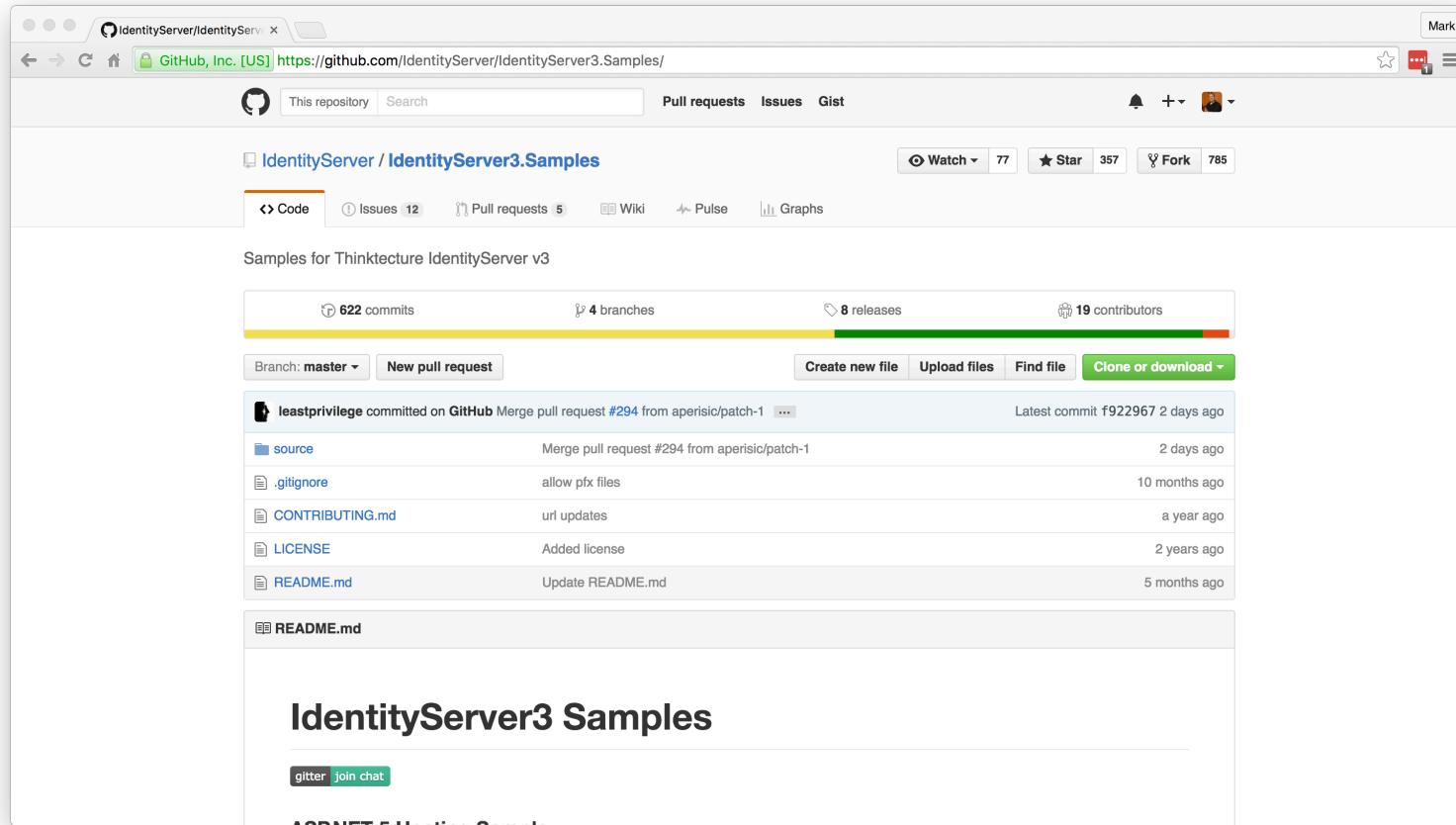


The screenshot shows a web browser window with the following details:

- Title Bar:** "latest version" (highlighted in red)
- Address Bar:** <https://identityserver.github.io/Documentation/docsv2/>
- Page Title:** IdentityServer3
- Page Subtitle:** latest version 1.x
- Content Area:**
 - Section:** latest version
 - Section:** Overview
 - [The big Picture](#)
 - [High level Features](#)
 - [Terminology](#)
 - [Features and Specifications](#)
 - [Packaging](#)
 - [Getting Started: Creating the simplest OAuth2 Authorization Server, Client and API](#)
 - [Getting Started: MVC Authentication & Web APIs](#)
 - [Getting Started: JS Authentication & Web APIs](#)
 - Section:** Configuration
 - [Overview](#)
 - [Options](#)
 - [Service Factory](#)
 - [In-Memory Services and Stores](#)
 - [Clients](#)
 - [Scopes and Claims](#)
 - [Secrets](#)
 - [Keys, Signatures and Cryptography](#)
 - [Authentication Options](#)
 - [Identity Providers](#)
 - [HSTS](#)

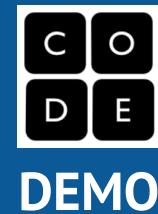
- <https://identityserver.github.io/Documentation/docsv2/>

Getting Started



- <https://github.com/IdentityServer/IdentityServer3.Samples/>

IdentityServer



Resources

- OpenID Connect:
<http://openid.net/connect/>
- OpenID Connect Libraries:
<http://openid.net/developers/libraries/>
- IdentityServer:
<https://github.com/IdentityServer>
- IdentityServer Samples:
<https://github.com/IdentityServer/IdentityServer3.Samples/>
- OAuth for ASP.NET:
<http://www.oauthforaspnet.com/>
- JSON Web Tokens: <https://jwt.io/>



Authentication Using OpenID Connect and OAuth2

Mark A. Wilson
Senior Developer
MarkW@LogicalAdvantage.com
www.DeveloperInfra.com
[@DeveloperInfra](https://twitter.com/DeveloperInfra)