org.clazzes.login.oauth

The OAuth login module is a planned login facility providing access to third party OAuth-2.0 and OpenID/Connect Services.

The login service might also be configured to accept access tokens of issued to third parties by an authorization provider.

Configuration

The org.clazzes.login.oauth HttpLoginService is configured by the standard OSGi configuration service using the properties mentioned below:

Property	Description
sessionCookie	The name of the cookie to set in user agents.
sessionTimeout	The timeout for cookie-based sessions in minutes. Sessions inactive for this time interval will be purged including all access /refresh/ID tokens requested from an OAuth/OpenID Provider.
secureCookie	The secure flag of the issued cookie. Set this value to true, if your are located behind an SSL-terminated ReverseProxy.
delegateDomain	The domain against which to check incoming bearer tokens. If not set, incoming bearer tokens will not be accepted by the OAuth HttpLoginService.
domain.<domain>.label	The mandatory human-readable label for the configured domain with identifier <domain>.
domain.<domain>. authorizationLocation	The OAuth2 authorization endpoint URL. This value does not need to be set for full-featured OpenID Providers, where this value is fetched from the specified configurationLocation
domain.<domain>. tokenLocation	The OAuth2 token endpoint URL. This value does not need to be set for full-featured OpenID Providers, where this value is fetched from the specified configurationLocation
domain.<domain>. userLocation	The optional OAuth2 userinfo endpoint URL. This value does not need to be set for full-featured OpenID Providers, where this value is fetched from the specified configurationLocation
domain.<domain>. configurationLocation	The well-known OpenID Connect configuration location.
domain.<domain>. faviconLocation	The optional favicon location for domains, which do not have a `/favicon.ico` resource on the root of their authorization web host.
domain.<domain>.clientId	The client ID of our application as registered at the OAuth Provider.
domain.<domain>. clientPassword	The password for the client ID of our application as registered at the OAuth Provider.
domain.<domain>.scope	The mandatory scope to pass to the authorization endpoint.
domain.<domain>.prompt	The optional prompt value to pass to the authorization endpoint.
domain.<domain>. responseType	The optional response type to pass to the authorization endpoint.
domain.<domain>.options	Comma-separated list of options from the set • `lenientAccessTokenCheck` - Used to by pass at_hash checks in issued ID tokens, need e.g. for microsoft providers. • `propagateLocale` - Used to propagate the locale of the login iframe to the OAuth provider as the `locale` URL parameter.

Examples

github.com

Github implements OAuth2 and is not a full-features OpenID Connect provider.

Property	Value
domain.GITHUB.authorizationLocation	http://github.com/login/oauth/authorize
domain.GITHUB.userLocation	https://api.github.com/user

domain.GITHUB.label	github.com
domain.GITHUB.clientId	Cleint ID a registered under 'Authorized OAuth Apps' https://github.com/settings/applications
domain.GITHUB.clientPassword	Password of the above mentioned client ID.
domain.GITHUB.tokenLocation	https://github.com/login/oauth/access_token
domain.GITHUB.scope	user

google.com

Google implements a clean OpenID Connect provider with no hazzles.

Property	Value
domain.GOOGLE.clientId	Client ID as registered under https://console.developers.google.com/apis/credentials
domain.GOOGLE.clientPassword	Password of the above mentioned client ID.
domain.GOOGLE.configurationLocation	https://accounts.google.com/.well-known/openid-configuration
domain.GOOGLE.label	google.com
domain.GOOGLE.scope	openid profile email
domain.GOOGLE.accessType	offline
domain.GOOGLE.prompt	consent

microsoftonline.com

Microsoft implements OpenID connect, but leaves out the `at_hash` claim in ID tokens.

Property	Value
domain.MICROSOFT.clientId	https://apps.dev.microsoft.com/#/appList
domain.MICROSOFT.clientPassword	Password of the above mentioned client ID.
domain.MICROSOFT.configurationLocation	https://login.microsoftonline.com/common/v2.0/.well-known/openid-configuration
domain.MICROSOFT.label	microsoft.com
domain.MICROSOFT.scope	openid profile User.Read offline_access
domain.MICROSOFT.responseType	token id_token
domain.MICROSOFT.options	lenientAccessTokenCheck
domain.MICROSOFT.prompt	consent
domain.MICROSOFT.faviconLocation	https://www.microsoft.com/favicon.ico

Further Readings

OpenID 1.0 Specification: http://openid.net/specs/openid-connect-core-1_0.html

Microsoft's implementation notes: https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-v2-tokens

Microsoft Online OpenID configuration: https://login.microsoftonline.com/common/.well-known/openid-configuration

Google's OpenID Connect implementation notes: https://developers.google.com/identity/protocols/OpenIDConnect

Google Accounts OpenID configuration: https://accounts.google.com/.well-known/openid-configuration

github OAuth Guide: https://developer.github.com/v3/oauth

IANA registry of JSON Web Token Claims: https://www.iana.org/assignments/jwt/jwt.xhtml

RFCs

RFC 7515, JSON Web Signature (JWS), https://tools.ietf.org/html/rfc7515

RFC 7516, JSON Web Encryption (JWE), https://tools.ietf.org/html/rfc7516

RFC 7517, JSON Web Key (JWK), https://tools.ietf.org/html/rfc7517

RFC 7518, JSON Web Algorithms (JWA), https://tools.ietf.org/html/rfc7518

RFC 7519, JSON Web Token (JWT), https://tools.ietf.org/html/rfc7519