Shibboleth V5 New Features

Identity Provider V4 to V5 Notable Changes/Features

- Java 17, Spring V6, Servlet Spec 5 (Jakarta EE 9).
  - From Java 11, Spring V5, Servlet Spec 4 (Jakarta EE8)
- Removed settings and properties.
- Removed features
  - E.g. legacy Duo flow, exportAllAttributes in DataConnectors…
  - Previously deprecated, check your logs from the latest 4.X
- Changes to defaults
- Changes to APIs
- Content-Security-Policy improvements
  - Tighten the default CSP policy and allow better customization (v5.1)
- New view rendering technologies supported as plugins (not just velocity)
- Add arbitrary request headers to the logging Message Diagnostics Context (MDC).
- Save the username into a sealed cookie to pre-populate the username in specific cases (for password flow right now, but extendable to others) (v5.1)

Existing Plugins

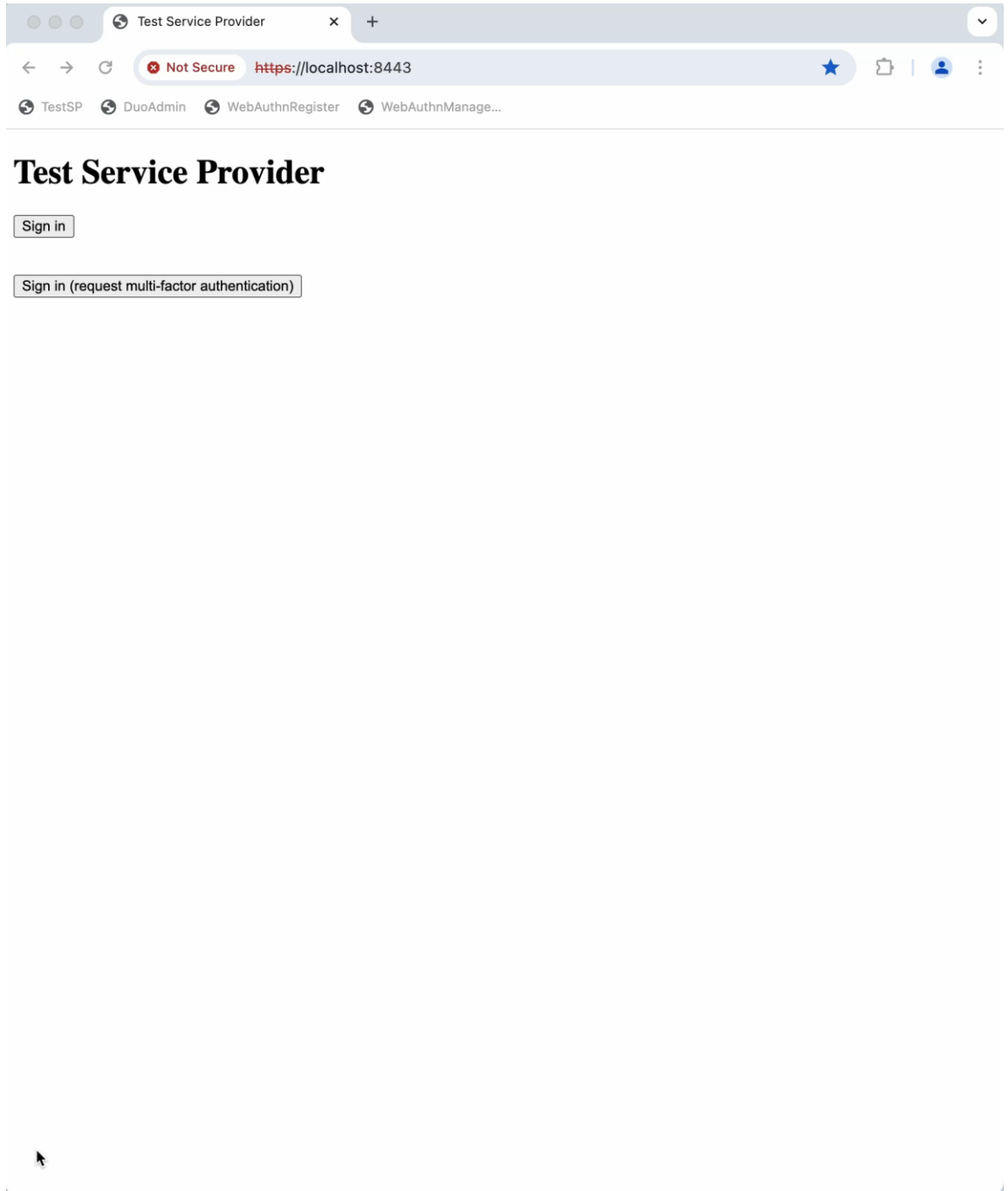
- All
  - Update required for V5 compatibility
- OpenID Provider V4
  - An unregistered (no metadata) RP can access a specific profile if their request matches to a given **policy** e.g. this client_id with this redirect_uri.
  - OIDC Logout
    - OPLogout profile. RP initiated logout via both front-channel and back-channel bindings
  - 4.2.0 (unreleased)
    - Pushed Authorisation Requests (PAR)
      - RP 'pushes' authorization request to OP via the backchannel, before making a front-channel authorization request that references it.
        - Errors spotted in advance of user navigation
        - Other security benefits of back-channeling the request parameters
    - Demonstratable Proof of Possession (DPoP) tokens
      - Bind access_tokens to a particular client. As opposed to bearer tokens

Existing Plugins

- OpenID Relying Party V2
  - Proof of Key Code Exchange (PKCE) support
    - Protect the token exchange.
      - Injection of an authorization code is impossible without knowledge of the PKCE verifier
  - Arbitrary claims can be added to the Request Object during the authorization request.
- DuoOIDC V2
  - 2.1.0 adds ‘passwordless’ support (support of passkeys)
    - Use Duo as the sole authentication mechanism, and not just a second-factor
    - After first use as a second-factor, determine if the user uses a second factor that is acceptable for passwordless. If so, opt-in and user that as a sole factor for subsequent authentications

New Plugins

- WebAuthn (current alpha, beta this week)
  - Public-key-based strong authentication using WebAuthn credentials e.g., passkeys.
  - Usernameless (passkeys)
    - A usernameless flow does not require the user to enter their username during authentication. Authenticator must support and store discoverable (passkey) credentials.
      - Requires user-verification by the authenticator for each authentication e.g. PIN or Biometrics.
  - Passwordless
    - First, collect the username; this does not require storing credentials on the authenticator.
      - Requires user-verification by the authenticator for each authentication e.g. PIN or Biometrics.
  - Second-factor
    - Use a WebAuthn credential as a second factor only.
      - Does NOT require user-verification (only that the user is present i.e. tapping an authenticator button)


Test Service Provider
Not Secure
https://localhost:8443
TestSP
DuoAdmin
WebAuthnRegister
WebAuthnManage...
Test Service Provider
Sign in
Sign in (request multi-factor authentication)

Under Discussion

- Java-based Service Provider
  - Shibd as an ‘IdP’ plugin
  - Integrations to Apache etc.
- OpenID Federation
  - Government use cases
  - eduGAIN pilot starting in September
    - Digital credential use case?
- Verifiable/Digital Credential issuance
  - Digital wallets, Verifiable Credentials.
    - eIDAS regulations (electronic identification and trust services), DC4EU etc.
      - EU citizens having full control of their personal data
    - R&E IdP to issue educational credentials?
    - Others:
      - Social Security, Person Identification Data
      - Professional qualifications
    - Replacing OIDC/SAML?
    - UK?