ChatGPT

Web SDK (TypeScript) for WA Digital Wallet

The TypeScript SDK allows web applications and verifiers to request and verify presentations from citizens' wallets. It is designed for integration with React or vanilla JS applications.

Installation

Install via npm:

```
npm install @wa-digital-wallet/sdk --save
```

Configuration

```
import { VerifierClient } from '@wa-digital-wallet/sdk';

const client = new VerifierClient({
  apiBaseUrl: 'https://wallet.api.gov.wa.au/api/v1',
  clientId: 'verifier-client-id',
  redirectUri: 'https://verifier.example.com/callback',
});
```

Creating a presentation request

```
const request = await client.createPresentationRequest({
  requestedClaims: {
    givenName: { essential: true },
    familyName: { essential: true },
    dateOfBirth: { optional: true },
  },
});
// request.qrCode contains an SVG/PNG you can display to the citizen.
```

Verifying a presentation

After the citizen scans the QR code and approves the request in their wallet app, the SDK handles the callback and verification:

```
client.handleCallback().then(async (presentationToken) => {
  const result = await client.verifyPresentation({ presentation:
presentationToken });
  if (result.verified) {
```

```
    console.log('Presentation valid:', result.disclosedClaims);
  } else {
    console.error('Verification failed:', result.status);
  }
});
```

Offline & fallback modes

The SDK supports offline presentation by embedding credential status information in the QR code. If the verifier cannot reach the wallet service, it can still perform basic checks but should sync when connectivity is restored.

Type definitions & extensibility

Full TypeScript definitions are included. The SDK is modular and can be extended to support additional credential schemas or presentation protocols as they emerge.

Contribution & support

Bug reports and contributions are welcome via the GitHub repository. Please follow the coding standards and provide tests for new features.
