API Usage (JSUI)

This document summarizes the API calls used by the JSUI single-page client to retrieve and save account/repository metadata and repository views. The calls match the patterns used in the `dydra-client` implementation.

Base host is captured from login and normalized to `https://<host>`.

Authentication

Requests use either Basic auth or a Bearer token. After login, the access token returned by the account configuration endpoint is used as a Bearer token.

Headers: - `Authorization: Basic <base64(account:password)>` (initial login) - `Authorization: Bearer <token>` (after login)

Account Metadata

Get account configuration

Used to authenticate and to load account metadata.

```
GET /system/accounts/<account>/configuration
Accept: application/json
Authorization: <Basic or Bearer>
```

JSUI usage: - `authenticateAccount()` in `jsui/lib/auth.js` - Returned `accessToken` is stored and used for subsequent requests.

Update account configuration

Used to save edited account profile fields. JSUI tracks edits by swapping only double-clicked fields into inputs; only those inputs are collected for saving. Unedited fields are not included in the payload.

```
POST /system/accounts/<account>/configuration
Accept: application/json
Content-Type: application/json
Authorization: Bearer <token>

{ ...edited fields only... }
```

JSUI usage: - `updateAccountConfiguration()` in `jsui/ui/pages/index.js` - Triggered by the per-tab Save button after editing profile fields.

Repository Metadata

Get repository configuration

Used to fetch repository metadata (description, privacy, prefixes, etc.) and the list of views.

```
GET /system/accounts/<account>/repositories/<repository>/configuration
Accept: application/json
Authorization: Bearer <token>
```

JSUI usage: - `fetchRepositoryConfig()` in `jsui/ui/pages/index.js` - `fetchRepositoryViews()` pulls `config.views` from this response.

Update repository configuration

Used to save edited repository metadata fields. JSUI only includes fields that were edited (inputs created via double-click), not the full config object.

```
POST /system/accounts/<account>/repositories/<repository>/configuration
Accept: application/json
Content-Type: application/json
Authorization: Bearer <token>

{ ...edited fields only... }
```

JSUI usage: - `updateRepositoryConfiguration()` in `jsui/ui/pages/index.js` - Triggered by the per-tab Save button after editing repository metadata.

Views

List views (via repository configuration)

Views are provided as `config.views` from the repository configuration response. JSUI normalizes either string or object view entries.

```
GET /system/accounts/<account>/repositories/<repository>/configuration
Accept: application/json
Authorization: Bearer <token>
```

Read view query text

Used to load the SPARQL query text for a view.

```
GET /system/accounts/<account>/repositories/<repository>/views/<view>
Accept: application/sparql-query
Authorization: Bearer <token>
```

JSUI usage: - `fetchViewText()` in `jsui/ui/pages/index.js`

Execute view query (SPARQL endpoint)

Used by the SPARQL editor to execute queries.

```
POST /<account>/<repository>/sparql
Content-Type: application/sparql-query
Accept: application/sparql-results+json
Authorization: Bearer <token>
```

JSUI usage: `-` `createSparqlEditor()` is configured with `sparqlEndpoint` and `accessToken`.