

This document outlines the TKBio workflow as a series of KSAPI calls

I. Discover a list of candidate concepts

Input: A URL encoded character string to match against the canonical or aliases of a concept

Output: (paged) list of matching concepts from all KS's supporting the API call

Variants: Text and semantic type filters can constrain the list; results are batched as pages

API endpoint:

```
GET /concepts?textFilter=<search string>
```

II. User selects a concept to get at a table of all its relationships

A. System identifies equivalent concepts identifiers across KS's

1. Given a concept, return all associated cross references

Input: A specified globally unique user-selected concept identifier

Output: List of cross-references associated with the concept

API endpoint:

```
GET /xref/{conceptId}
```

2. Initiate an equivalent concept discovery query with cross-references

Input: List of cross-references associated with the concept

Output: Equivalent concept query session identifier (ECQSID)

API endpoint:

```
POST /xref
```

with a JSON-formatted list of cross-references (e.g. output from II.A.1 above)

3. Given a query session identifier, return all associated cross references

Input: Query session identifier (ECQSID)

Output: List of matching concepts annotated with cross-references

API endpoint:

```
GET /xref?queryId=<some valid ECQSID >
```

4. The lists of cross-references from II.A.3 above are consolidated into a union set and identifiers that were already initially run are subtracted from the resulting set, then the difference set of identifiers are used in iterative calls to II.A.2 and II.A.3, until the resulting “equivalent concept” identifier clique ceases to expand.

B. Set of equivalent concepts are used to retrieve related statements.

1. Given an input list of concepts,

Input: “Equivalence clique” of concept identifiers

Output: Statement query session identifier (SQSId)

API endpoint:

`POST /statements`

with a JSON-formatted list of concepts (e.g. output from II.A.4 above)

2. Given a query session identifier, return associated relationships

Input: Query session identifier (SQSId)

Output: List of matching subject-predicate statements

API endpoint:

`GET /statements?queryId=<some valid SQSId>`

III. User selects a specific concept to get at details

Input: A specified globally unique user-selected concept identifier

Output: A more complete report of properties of the concept

API endpoint:

`GET /concepts/{conceptId}`

IV. User selects a specific statement to get at evidence

Input: The evidenceId associated with a given statement (in output of II.B.2)

Output: A list of citation references supporting the statement

API endpoint:

`GET /evidence/{evidenceId}`