

😊 Add icon 🖼️ Add cover

9.08 - Add filters to Datasets

Currently have filters for tags, groups, orgs. Maybe consider additional filters for date range, created by me, datasets I have access to etc.

As a user of CORDAP,

I want to be able to filter datasets in the search results

So that I can easily find the dataset I need from a long list of datasets

Acceptance Criteria:

1. Relevant filters are available in the list of datasets to organise which are presented (e.g. filter by owned datasets, dataset role, time period)

1. Overview

This document outlines the technical design for implementing additional dataset filters in CKAN to improve user experience when searching through large datasets. The enhancement will add new filtering capabilities including date range, ownership, access permissions, and user roles.

2. Current State Analysis

Based on the codebase analysis, CKAN currently has:

- Basic facet-based filtering (tags, groups, organizations)
- Search functionality with sorting options
- Filter infrastructure using `view_form_filters.html`
- JavaScript modules for filter handling in `resource-view-filters.js`

3. Proposed Solution

3.1 New Filter Types

3.1.1 Date Range Filters

- **Date Range:** Filter datasets by creation date or Filter datasets by last modification date

3.1.2 Access Filters

- **Datasets I have access to:** Show datasets the user has read/write permissions

3.1.3 Role-based Filters

- **My Role in Dataset:** Filter by user's role (admin, editor, member) in specific datasets

4. Technical Implementation Strategy

4.1 Backend Components

4.1.1 Search Schema Extensions

</> Python

```

1 # Location: ckan/logic/schema.py
2 def enhanced_package_search_schema():
3     schema = default_package_search_schema()
4     schema.update({
5         'created_date_from': [ignore_missing, isodate],
6         'created_date_to': [ignore_missing, isodate],
7         'modified_date_from': [ignore_missing, isodate],
8         'modified_date_to': [ignore_missing, isodate],
9         'created_by_me': [ignore_missing, boolean_validator],
10        'accessible_to_me': [ignore_missing, boolean_validator],
11        'my_role': [ignore_missing, unicode_safe],
12        'license_type': [ignore_missing, unicode_safe],
13        'resource_format': [ignore_missing, unicode_safe]
14    })
15    return schema
16

```

4.1.2 Search Action Modifications

Extend get.py `package_search` function:

</> Python

```

1 # Enhanced package_search logic
2 def package_search(context, data_dict):
3     # Existing logic...
4
5     # Add custom filter processing
6     filters = _process_enhanced_filters(context, data_dict)
7
8     # Apply filters to Solr query
9     query = _build_enhanced_solr_query(query, filters)
10
11    return existing_search_logic(query)
12
13 def _process_enhanced_filters(context, data_dict):
14    user = context.get('user')
15    filters = {}
16
17    # Date range filters
18    if data_dict.get('created_date_from') or data_dict.get('created_date_to'):
19        filters['date_range'] = _build_date_filter(
20            'metadata_created',
21            data_dict.get('created_date_from'),
22            data_dict.get('created_date_to')
23        )
24
25    # Ownership filters
26    if data_dict.get('created_by_me') and user:
27        filters['creator_user_id'] = user
28
29    return filters
30

```

4.1.3 Database Schema Updates

</> SQL

```

1 -- Add indexes for performance
2 CREATE INDEX idx_package_creator_user_id ON package(creator_user_id);

```

```

3 CREATE INDEX idx_package_metadata_created ON package(metadata_created);
4 CREATE INDEX idx_package_metadata_modified ON package(metadata_modified);
5

```

4.2 Frontend Components

4.2.1 Template Enhancements

Create new filter template: `ckan/templates-bs3/snippets/enhanced_dataset_filters.html`

```
{% macro enhanced_filters() %}
```

Advanced Filters

Date Range

Ownership

```
{% endmacro %}
```

4.2.2 JavaScript Module for Enhanced Filters

Create `ckan/public-bs3/base/javascript/modules/enhanced-dataset-filters.js`

```

1 ckan.module('enhanced-dataset-filters', function (jQuery) {
2   return {
3     initialize: function () {
4       this.setupDateRangeFilters();
5       this.setupOwnershipFilters();
6       this.setupFormSubmission();
7     },
8
9     setupDateRangeFilters: function() {
10      // Date picker initialization and validation
11    },
12
13    setupOwnershipFilters: function() {
14      // Checkbox handling and AJAX updates
15    }
16  };
17 });
18

```

5. Implementation Tasks and Estimates

| | ☰ Task | ☰ Estimate (Hrs) | ☰ Description |
|---|---------------------|------------------|-----------------------|
| 1 | Database Migrations | 2 | Add necessary indexes |

| | | | |
|---|--------------------------|----|--|
| 2 | Search Logic Enhancement | 6 | Modify package_search to handle new filters |
| 3 | Date Range Validation | 3 | Implement proper date validation |
| 4 | AJAX Filter Updates | 8 | Real-time filter updates without page reload |
| 5 | Filter State Persistence | 6 | Remember filter states in URL/session |
| 6 | Testing and Deployment | 7 | Update user and developer documentation |
| 7 | Total Effort | 32 | |

6. Technical Considerations

- **Database Indexing:** Add appropriate indexes for date and user-based queries

Security Considerations

- **SQL Injection:** Validate all filter inputs to prevent injection attacks