TrackHub Registry
Code Review

07/05/2015

EMBL – European Bioinformatics Institute
Wellcome Trust Genome Campus
Hinxton, Cambridge, CB10 1SD, UK

Where we are

- JSON schema
- REST API
- Web front-end

JSON Schema (1)

- Draft03
  - Pros: complete JSON model of UCSC specs
    - proper data types
    - component relations easier to understand
  - Cons: big & complex
    - TH providers might be initially discouraged to adopt

JSON Schema (2)

- v1.0
  - e! web TH parsing code
    - properties mapped to string key/value pairs
  - TrackDb metadata
    - Hub/Species/Assembly
    - Track metadata: key/value strings
  - TrackDb configuration (not indexed by ES)
    - key/value strings + “members” object for container track
    - do not distinguish between hierarchical container types

The Model

- A track hub is a collection of trackDBs, grouped by assembly
- A trackDB is a collection of related tracks
- A document in ES represents a trackDB
  - No explicit hub type
  - Hub redundantly repeated for all trackDBs belonging to the same Hub

```
{
  "hub": "Blueprint example 1",
  "description": "An example of trackhub DB from the Blueprint project (1 track)",
  "version": "v1.0",
  "species": {
    "tax_id": 9606,
    "scientific_name": "Homo sapiens"
  },
  "assembly": {
    "accession": "GCA_000001405.15",
    "name": "GRCh37",
    "synonyms": "hg19"
  },
  "data": [
    {
      "id": "bpDnaseRegionsC0010K46DNaseEBI",
      "molecule": "genomic_DNA",
      "disease": "None",
      "biomaterial_type": "Primary_cells",
      "sample_id": "EGAN00001070025",
      "donor_id": "C0010K46",
      "donor_age": "60-65",
      "donor_health_status": "Healthy",
      "donor_sex": "Female",
      "donor_ethnicity": "Northern_European",
      "cell_type": "CD14-positive, _CD16-negative_classical_monocyte",
      "tissue_type": "Venous_blood",
      "library_strategy": "DNAse-Seq",
      "experiment_type": "Chromatin-_accessibility",
      "experiment_id": "EGAX00001084791",
      "alignment_software": "BWA",
      "alignment_software_version": "0.5.9",
      "analysis_software": "Hotspot",
      "analysis_software_version": "v3",
      "analysis_group": "EMBL-EBI"
    }
  ],
  "configuration": {
    "bpDnaseRegionsC0010K46DNaseEBI": {
      "shortLabel": "C0010K.DNase.Mono",
      "longLabel": "C0010K DNase monocytes peaks (EBI)",
      "visibility": "full",
      "bigDataUrl": "ftp://ftp.ebi.ac.uk/pub/databases/blueprint/data/homo_sapiens/Peripheral_blood/
      "type": "bigbed",
```

```
{
  "hub": "Blueprint example 1",
  "description": "An example of trackhub DB from the Blueprint project (1 track)",
  "version": "v1.0",
  "species": {
    "tax_id": 9606,
    "scientific_name": "Homo sapiens"
  },
  "assembly": {
    "accession": "GCA_000001405.15",
    "name": "GRCh37",
    "synonyms": "hg19"
  },
  "data": [
    {
      "id": "bpDnaseRegionsC0010K46DNaseEBI",
      "molecule": "genomic_DNA",
      "disease": "None",
      "biomaterial_type": "Primary_cells",
      "sample_id": "EGAN00001070025",
      "donor_id": "C0010K46",
      "donor_age": "60-65",
      "donor_health_status": "Healthy",
      "donor_sex": "Female",
      "donor_ethnicity": "Northern_European",
      "cell_type": "CD14-positive,_CD16-negative_classical_monocyte",
      "tissue_type": "Venous_blood",
      "library_strategy": "DNAse-Seq",
      "experiment_type": "Chromatin-_accessibility",
      "experiment_id": "EGAX00001084791",
      "alignment_software": "BWA",
      "alignment_software_version": "0.5.9",
      "analysis_software": "Hotspot",
      "analysis_software_version": "v3",
      "analysis_group": "EMBL-EBI"
    }
  ],
  "configuration": {
    "bpDnaseRegionsC0010K46DNaseEBI": {
      "shortLabel": "C0010K.DNase.Mono",
      "longLabel": "C0010K DNase monocytes peaks (EBI)",
      "visibility": "full",
      "bigDataUrl": "ftp://ftp.ebi.ac.uk/pub/databases/blueprint/data/homo_sapiens/Peripheral_blood/
      "type": "bigbed",
```


moving to an object, e.g.
hub => {
    name => …
    shortLabel => …
    longLabel => ...
}

array of track metadata, otherwise ES won't index !nested types

verbose copy from e! parser, not indexed

```
{
   "hub" : "Blueprint Epigenomics Data Hub",
   "version" : "v1.0",
   "configuration" : {
      "bp" : {
         "track" : "bp",
         "longLabel" : "Blueprint",
         "compositeTrack" : "on",
         "members" : {
            "signal" : {
               "parent" : "bp",
               "visibility" : "pack",
               "shortLabel" : "Blueprint Signal",
               "track" : "signal",
               "signal_range" : [],
               "maxHeightPixels" : "64:32:16",
               "autoscale" : "off",
               "members" : {
                  "bpRNASU-DHL-5_r01RNA-SeqRNAminusCRG" : {
                     "parent" : "signal",
                     "visibility" : "pack",
                     "shortLabel" : "SU-DHL-5.RNA-Seq.-.B_Lymph",
                     "track" : "bpRNASU-DHL-5_r01RNA-SeqRNAminusCRG",
                     "bigDataUrl" : "http://ftp.ebi.ac.uk/pub/databases/blueprint/data/homo_sapiens/
DHL-5_r01.minusStrand.gem_grape_crg.20131126.bw",
```


a composite track, members can be simple tracks or views

a view, contains simple tracks

a simple track

The REST API

- trackDBs programmatic CRUD for track hub providers
- 7 endpoints (GET/PUT/POST/DELETE)
- Token-based authentication, only after online registration

CRUD

- GET /api
  - Return the list of available endpoints


```
[
  ["/api/trackhub", "GET", "Return the list of available docs (id => URI)"],
  ["/api/trackhub/create", "PUT", "Create new trackhub document"],
  ["/api/trackhub/create", "POST", "Create new trackhub document by converting assembly trackdbs from a remote public hub"],
  ["/api/trackhub/:id", "GET", "Return content for a document with the specified ID"],
  ["/api/trackhub/:id", "POST", "Update content for a document with the specified ID"],
  ["/api/trackhub/:id", "DELETE", "Delete document with the specified ID"]
]
```

CRUD (2)

- GET /api/trackhub
  - return the list of available docs for a user
  - doc ID mapped to the URI of the resource which represents the doc
- GET /api/trackhub/:id
  - return doc for a trackDB with the given ID

CRUD

- PUT /api/trackhub/create
  - User submits a new trackDB
  - Argument (optional): JSON schema version (default: v1.0)
  - Body:
    - Content-type: application/json
    - The trackDB doc in a particular schema version

CRUD (2)

- POST /api/trackhub/create
  - User creates trackDB(s) as translations from a remote hub (UCSC)
  - Argument: same as before
  - Body: remote hub URL/assembly
    - Content-type: application/json
    - { url => '…', assembly => '<UCSC synonym>' }
    - Unless assembly, translate/submit all

UCSC Hub Translation

- Dispatch mechanism for handling different versions
  - 1 callback ATMO (v1.0)
  - call rewrite of e! web trackhub parser
- One big manually curated map from UCSC assembly to NCBI assembly set accessions
  - from https://genome.ucsc.edu/FAQ/FAQreleases.html, error correction + other entries found in UCSC public hubs
  - die if mapping not found
  - query the Genome Assembly DB to fill in species/assembly metadata

CRUD (3)

- Validation
  - occurs when a doc is submitted/updated
  - call python script using Draft4Validator from jsonschema library
    - Perl alternative validates everything!
  - If error
    - dump all to STDOUT
    - calling module intercepts and raise exception
    - any error prevent subsequent actions
    - REST API returns bad request with error

Translation tests

- Tested sequential submission of all 27 UCSC listed public track hubs
  - http://genome.ucsc.edu/cgi-bin/hgHubConnect
- 12/27 hubs cannot be submitted
  - 7 do not validate (most have empty fields)
  - 5 generate bad requests
    - 1 unrecognised assembly synonym (Anc08)
    - 2 trackDB not correctly formed (parent track missing)
    - 2 Internal exceptions (599)

CR**U**D

- POST /api/trackhub/:id
  - update content for a doc with the given id
  - user re-submits the whole doc
- Updates in ElasticSearch

  “Partial updates can be done through the API, which accepts a partial document.

  However, this just gets merged with the existing document, so the only way to actually update a doc is to retrieve it, change it, and then reindex the whole doc.”

CRUD

- DELETE /api/trackhub/:id
  - deletes a doc with the specified ID
- Delete in ES

  "*Deleting a doc doesn't immediately remove it from the disk – it just marks it as deleted. ES will clean up deleted docs in the background as you continue to index more data.*"

API: TODO

- Should be extended to external clients
  - search for genome browsers
  - no auth, a separate Catalyst controller
- TrackHub parser from e! io?!
- Complete exception handling for all endpoints

Known Issues

- GC DB can be accessed only within EBI
  - ***DBI:Oracle:host=ora-vm5-003.ebi.ac.uk;sid=ETAPRO;port=1571***
- Some translation tests fail (metadata)
  - e! parser does not correctly parse metadata enclosed in double quotes
- Some public hubs do not validate/cannot be submitted
  - empty fields, relax schema?
  - bad requests to be investigated
- Memory intensive validation
  - Some hubs (e.g. WashU Roadmap epigenomics) are very big and do not validate. Validator dumps to STDOUT and consumes a lot of CPU as well

Recently discovered!

- ES wouldn't index all documents in certain circumstances;
  - e.g. during testing of the interface, submit a selection of public hubs;
  - not all of them are available for search and display to the owner dashboard;
  - investigating ...

```
=head2 trackhub_create

Create new trackhub document

Action for /api/trackhub/create (PUT,POST)

=cut

sub trackhub_create :Path('/api/trackhub/create') Args(0) ActionClass('REST') {
  my ($self, $c) = @_;

  # get the version, if specified
  # otherwise set to default (from config parameter)
  my $version = $c->request->param('version') || Registry->config()->{TrackHub}{schema}{default};

  #
  # determine the ID of the doc to create
  #
  # retrieve list of all trackDBs, get the max ID
  my $docs = $c->model('Search')->search_trackhubs();
  my $current_max_id = max( map { $_->{_id} } @{$docs->{hits}{hits}} );

  $c->stash( id =>  $current_max_id?++$current_max_id:1, version => $version );
}
```

Web front-end

- A track hub aware, intuitive interface for searching track collections;
- A simple dashboard for registered track hub providers;
- Trackhub submission instructions and REST API documentation.

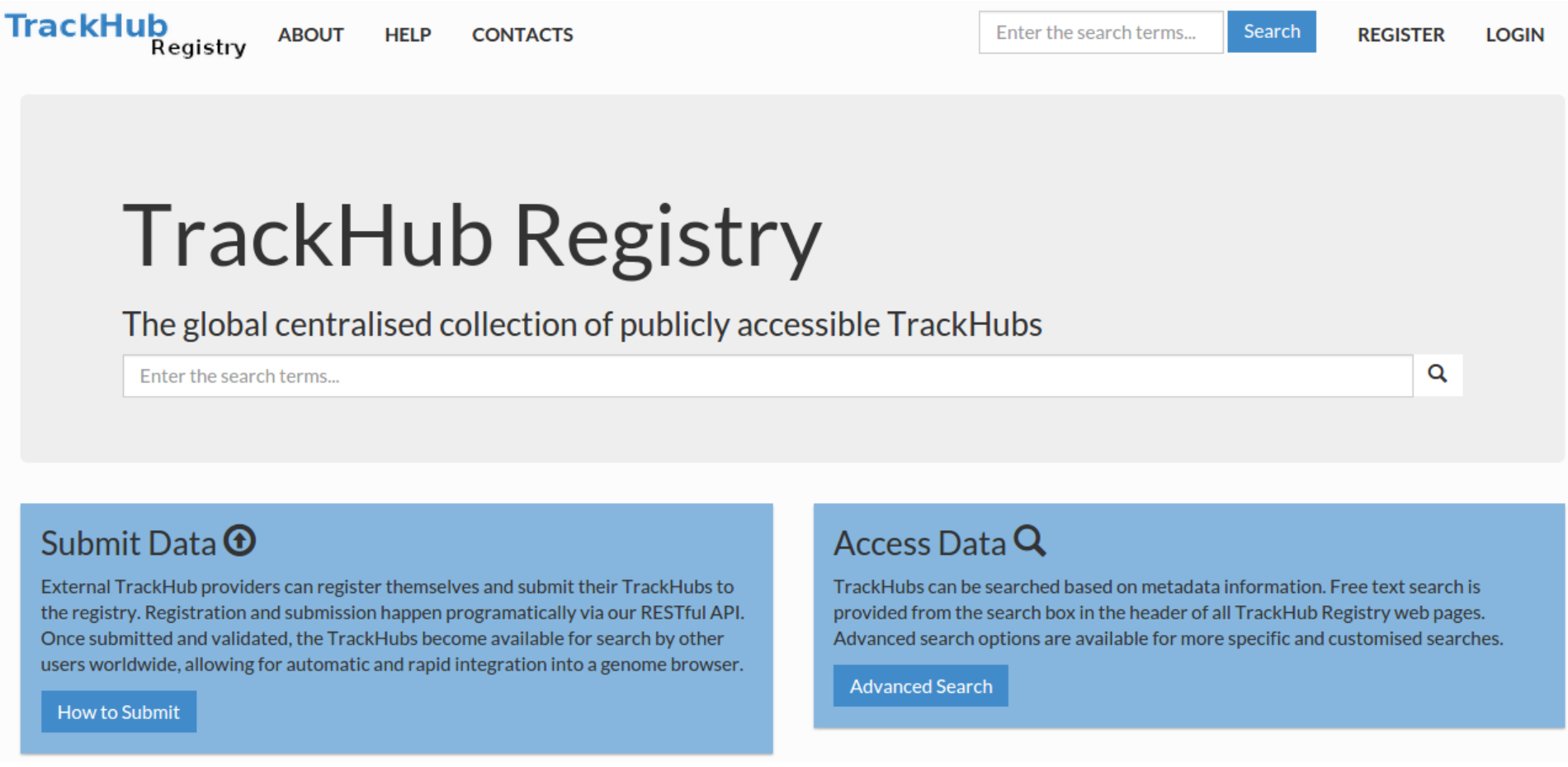
Git Repo

- https://github.com/Ensembl/trackhub-registry
- 3 branches
  - master
  - angularjs-bootstrap-frontend
  - jquery-bootstrap-frontend

master

- Catalyst application 

  - REST API
  - front-end with slightly customised bootstrap theme 
    - initial attempt
    - old bootstrap version
    - lots of things to do/refine, development stopped


TrackHub Registry
ABOUT
HELP
CONTACTS
Enter the search terms...
Search
REGISTER
LOGIN
TrackHub Registry
The global centralised collection of publicly accessible TrackHubs
Enter the search terms...
Submit Data
External TrackHub providers can register themselves and submit their TrackHubs to the registry. Registration and submission happen programatically via our RESTful API. Once submitted and validated, the TrackHubs become available for search by other users worldwide, allowing for automatic and rapid integration into a genome browser.
How to Submit
Access Data
TrackHubs can be searched based on metadata information. Free text search is provided from the search box in the header of all TrackHub Registry web pages. Advanced search options are available for more specific and customised searches.
Advanced Search

angularjs-bootstrap-frontend

- Catalyst app for REST API 

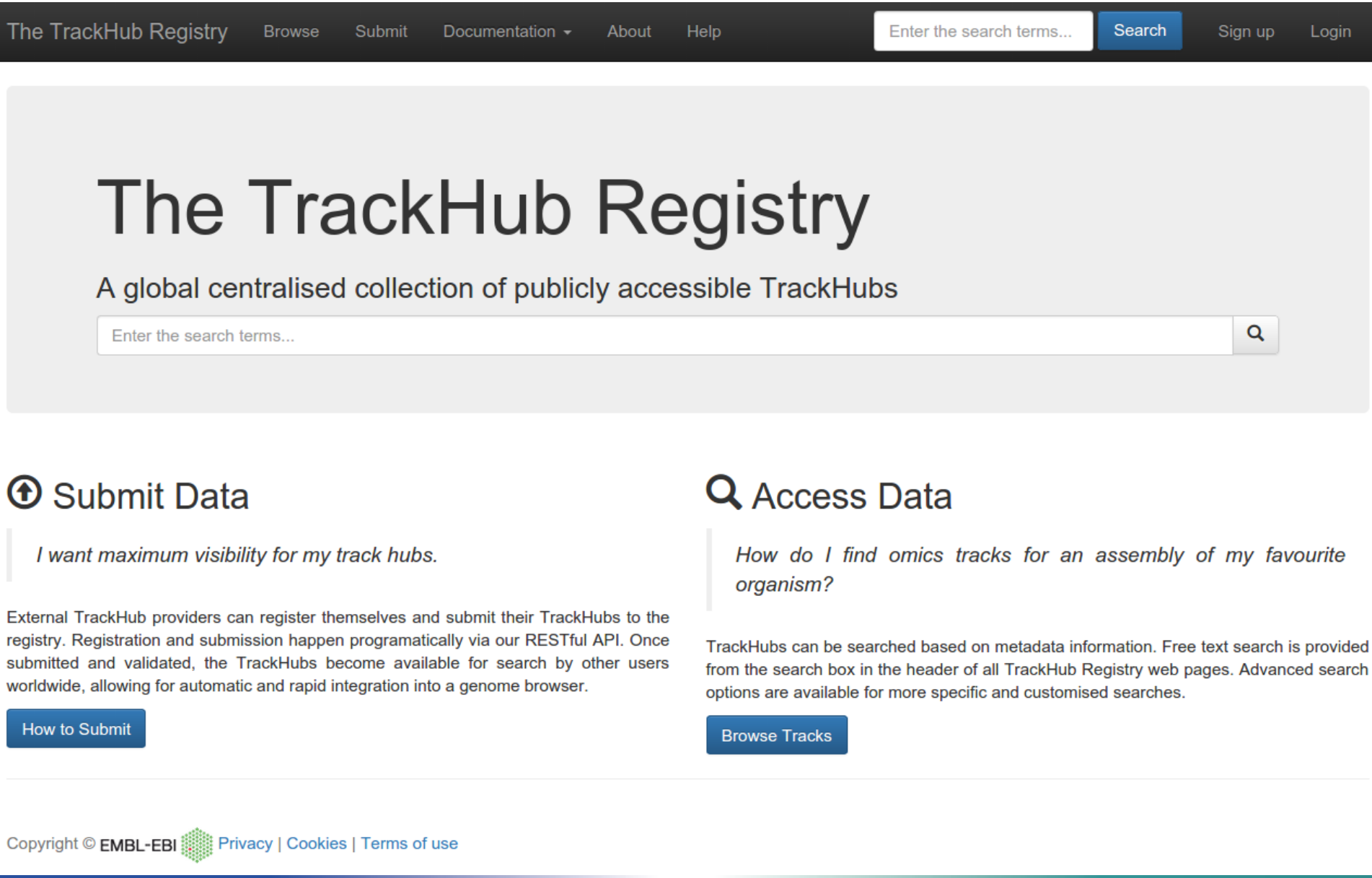
- Front-end as SPA
  - AngularJS 
  - UI Bootstrap 
  - Elasticui
- Taking inspiration from elasticsearch-gui
- Dev stalled
  - do not know enough AngularJS
  - steep learning curve
  - v2.x coming out soon, complete rewrite

jquery-bootstrap-frontend

- where the action is taking place
- Catalyst App 

  - REST API
  - Web front-end 


    - default bootstrap theme (minimalistic CSS)
    - jQuery to interact with bootstrap components, plus some effects for search


The TrackHub Registry
Browse
Submit
Documentation
About
Help
Enter the search terms...
Search
Sign up
Login
The TrackHub Registry
A global centralised collection of publicly accessible TrackHubs
Enter the search terms...
Submit Data
I want maximum visibility for my track hubs.
External TrackHub providers can register themselves and submit their TrackHubs to the registry. Registration and submission happen programatically via our RESTful API. Once submitted and validated, the TrackHubs become available for search by other users worldwide, allowing for automatic and rapid integration into a genome browser.
How to Submit
Access Data
How do I find omics tracks for an assembly of my favourite organism?
TrackHubs can be searched based on metadata information. Free text search is provided from the search box in the header of all TrackHub Registry web pages. Advanced search options are available for more specific and customised searches.
Browse Tracks
Copyright © EMBL-EBI Privacy | Cookies | Terms of use
wellcome trust sanger institute
EMBL-EBI