

CKAN User Meetup Berlin June 2013

Attendees

- Thomas Scheel (FOKUS), API, Metadata-Transformer
- Fabian Kirstein (FOKUS), extensions, ocdirectory.org
- Dominik (OKF)
- Anja (Hitec HH), Transparency-Register
- Jens Wartenberg (Dataport HH, schleswig-holstein.de), Computer center MS, 1M Dataset, Archive
- Simon Dutkowski (FOKUS), Connection to JAVA
- Dirk Eiche (HU Berlin), Linked Data, Harvesting Library Data
- Michael Kurtz (DKRZ - Deutsches Klimarechenzentrum)
- Heinrich Widmann (DKRZ) OAI Harvesting
- Sean Hammond (OKF) CKAN Core Dev
- Knud Möller (daten.berlin.de) Berlin CKAN
- Hendrik Bunke (ZBW Hamburg/Kiel), Research Data
- Philipp Lämmel (FOKUS) Harvesting

Questions/Topics:

https://docs.google.com/spreadsheet/ccc?key=0AimvNUb_Dv91dFpQSC00amE1aHAwRG9EZGJORnN6Rmc

Sean: Writing Extension

Writing Extensions is tricky right now due to missing documentation.

<http://seanh.sdf.eu/writing-ckan-extensions/>

and the not yet merged docs: <https://github.com/okfn/ckan/pull/943>

The OKF are working on improving the extension docs. In his presentation Sean shows how to write an extension with the new authenticator feature.

CKAN extensions are normal Python libraries that one installs into CKAN virtualenv.

One extension contains one or more plugins.

All plugin interfaces: <https://github.com/okfn/ckan/blob/master/ckan/plugins/interfaces.py>

CKAN 1.8's rights, roles, CLI are all gone in 2.x. Now you have a set of simple Boolean properties in config.ini: <http://docs.ckan.org/en/ckan-2.0/authorization.html#organizations>. More

complication authorization has to be done programmatically in an extension

CRUD functions in

<https://github.com/okfn/ckan/tree/master/ckan/logic/action>

call corresponding functions in a

<https://github.com/okfn/ckan/tree/master/ckan/logic/auth>

to determine authorization.

Groups can hold packages and users, Organizations can hold packages and users. Organizations are more geared toward authorization, though. Like several members of a governmental department who are able to access all datasets of that department. In turn, groups are more for putting datasets together that have a e.g. a special focus. Group members normally can't edit the groups packages.

Rely on functions in `ckan.plugins.toolkit`, these are promised to be maintained.

<http://docs.ckan.org/en/ckan-2.0/toolkit.html>

`get_converter()` not documented yet, for now see

<https://github.com/okfn/ckan/blob/master/ckan/logic/converters.py>

Debugging: Most convenient is `import ipdb; ipdb.set_trace()`

Pull request with new writing extensions tutorial and example `IAuthFunctions`

extension: <https://github.com/okfn/ckan/pull/943>

More example extensions:

https://github.com/okfn/ckan/tree/master/ckanext/example_idatasetform

https://github.com/okfn/ckan/tree/master/ckanext/example_itemplatehelpers

Java Middleware for CKAN (Simon)

Presentation: <http://open-data.fokus.fraunhofer.de/wp-content/uploads/2013/07/ODR-Client.pdf>

Focussed on GovData Schema (<https://github.com/fraunhoferfokus/ogd-metadata>)

Any Java app can access CKAN functions without doing REST and JSON.

Active in GovData.de and opendata.hamburg.de

Something much thinner/more dynamic exists for Python (ckanclient is old school and heavy, now: ckanapi <https://github.com/open-data/ckanapi>)

For now this one does not SOLR client classes directly.

Can quite easily be adapted to API changes or Metadata schema or extra fields.

Datastore (Dominik)

Brand new implementation in 2012

Needs separate Database

Datastore Resources belong to exactly one package resource

Right now slightly confusing: CSV file + datastore resource

Random SQL queries possible

http://docs.ckan.org/en/latest/datastore-api.html#ckanext.datastore.logic.action.datastore_search_sql

Preview for JSON, XML, GeoJSON, Images. Extensible.

Metadaten-Schema-Transformer (Thomas)

Presentation:

http://open-data.fokus.fraunhofer.de/wp-content/uploads/2013/07/open_metadata_transformer.pdf

Implemented in Java

Rule based transformation of JSON document (JSONT, which is analogous to XSLT)

Can export also to Linked Data.

Publishing is being discussed. For now please inquire via eMail.

Quick questions

Demo section groups on the start page don't update. Broken links. - Are cached.

Everything is discussed on the ckan-dev list. Things that are sure to be bugs should go to the github issues.

Is deletion of harvesting sources possible? Might be that not. A harvesting source is dataset, one can purge that, e.g. with paster.

Mandatory fields and validation in harvesting.

<https://github.com/fraunhoferfokus/ckanext-govdatade>

Package UUIDs are random. Collisions would result in errors.

Database backend: Nothing but Postgres has been done successfully. SQLAlchemy is used less and less, some postgres-specific functions are used. So using e.g. MySQL or Oracle are very unlikely to work.

CMS Integration

- CKAN behind CMS (daten.berlin.de govdata.de)
 - loose cool ckan features
 - be upgradable
 - a lot of work to get in running in the first place
- CKAN next to CMS (data.gov, data.gov.uk, EC-Portal)
 - recommended way of CKAN-CMS integration
 - get all features of CKAN
 - two set of user administration
 - upgrading templates is more difficult if there are a lot of customizations

Linked Data

Not a heavy topic right now in ckan dev

In works for European commission some Linked Data is done on Metadata level.