TIB Data Manager Manual

Table of Content

Table of Content 2

Description 3

About 3

Impact on Scientific Data Management 3

Application of the Data Manager 4

How to install 4

Dependencies 4

TIB Data Manager 5

Jupyterhub 5

Updating existing Jupyter Notebooks 7

Customizing The Data Manager 7

Changing The Data Manager Logo 7

Changing The Data Manager Header Color 9

Description

About

The TIB Data Manager has been developed to support the aspect of better re-usability of research data.

The prototype supports the management and access to heterogeneous research data publications and assists researchers in the selection of relevant data sets for their respective disciplines.

The prototype currently offers the following functions for the visualization of research data:

- Supports data collections and publications with different formats
- Different views on the same data set (2D and 3D support)
- Visualization of Auto CAD files
- Jupyter Notebook(s) for demonstrating live code
- RDF Description of data collections

The file specific viewers were implemented using CKAN (Comprehensive Knowledge Archive Network) plug-ins to render existing viewers for the datasets included in the CKAN instance.

Impact on Scientific Data Management

In the research data landscape, there is a high demand for a sustainable and meaningful handling of a main product of scientific work - research data.

Since digital data production has increased rapidly in recent years and an end to this growth is not foreseeable, the availability of these growing volumes of data must be ensured not only for current research but also for future generations. The TIB Data Manager was developed to provide scientists with a **tool to improve the usability of research data**.

The TIB Data Manager provides a data management system that makes it possible to **check the contents of research data sets for their potential application** to the respective domain **- without having to download** them beforehand.

Therefore, the Data Manager enables the visualization of different research data formats and thus supports the **'screening' of data sets** for their potential benefits. As a visualization and management tool, TIB CKAN can be implemented on top of classical research data repositories, which often focus on the (long-term) archiving and publication of research data.

Application of the Data Manager

As an open source tool, the TIB Data Manager **offers**

- **developers,**
- **scientists and**
- **data curators in public and academic research as well as in industry** a wide range of possibilities for **expanding and connecting established and developing research data management systems**, such als local and discipline-specific research data repositories.

As the German National Library of Science and Technology, we advise universities, research institutions and industry on the use and implementation of the TIB Data Manager. Furthermore, we continue to develop and enhance the functionality of the system in the view of the constantly growing number of scientific file formats.

How to install

Dependencies

The TIB Data Manager is composed by different services:

- CKAN: Open-source DMS (data management system) for powering data hubs and data portals
- PostgreSQL: Open-source object-relational database management system
- SOLR: Open-source enterprise search platform
- Postfix: Open-source mail transfer agent (MTA) that routes and delivers electronic mail
- DataPusher:

In order to avoid having to manually install these dependencies, the distribution package, comes with dockerized instances of these dependencies, making it easy to get started with TIB Data Manager.

The distribution package also contains a docker-compose file, where

Docker
Docker-compose 1.18.0+

TIB Data Manager

Docker is necessary so that the TIB Data Manager can be used. To be able install it, the user must download the docker packets from Docker official website (https://docs.docker.com/install/), and afterwards follow the installation step established in the packets.

To be able use CKAN with all the services it is necessary to follow the these steps:

1. First, it is necessary change the url of the site in the docker-compose.yml file, in the line 32 of the file the user needs to put the Url of the server to get access to the CKAN through the server.

   ```
   CKAN_SITE_URL: "<URL Server>:5000"
   ```

   Additionally make sure the following ports are free in the server:

   - Port 5000 for data pusher.
   - Port 8000 for Jupyter Notebooks.

2. Build the container base of CKAN inside the docker folder.

   ```
   $ docker build ./ckan_base -t ckan_base
   ```

3. Next the user needs to run

   ```
   $ docker-compose build  ckan
   ```

4. At the end, you need to run the following commands:

   ```
   $ docker-compose up -d pusher redis solr db postfix
   $ docker-compose up ckan
   ```

Jupyterhub

With JupyterHub you can create a multi-user Hub which spawns, manages, and proxies multiple instances of the single-user Jupyter notebook server.

If the user wishes to use Jupyter Notebook within the Data Manager, it is necessary to follow the these steps to install Jupyterhub:

1. It is necessary to install python3 in the user computer (if the computer already has python3 installed skip this step)

   ```
   $ sudo apt-get install python3-pip
   ```

2. Install nodejs/npm

   ```
   $ sudo apt-get install npm nodejs
   ```

3. Install proxy with npm

   ```
   $ npm install -g configurable-http-proxy
   ```

4. Install Jupyterhub

   ```
   $ pip3 install jupyterhub
   ```

5. Install Jupyter notebook (/upgrade)

   ```
   $ pip3 install --upgrade notebook
   ```

6. Test Jupyterhub default configuration

   ```
   $ jupyterhub
   ```

   This session will start in localhost:8000
   ** Make sure that port isn't protected under Firewall of your system

7. It is recommended to use secure SSL certificate file for the public facing interface of the proxy. To produce personal security certificates commands are as follows:

   ```
   $ openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout mykey.key -out
   mycert.pem
   ```

   It is not necessary to fill in the credentials.

To configure Jupyterhub is necessary to create a configuration file, which is generated through the following command:

```
$ jupyterhub --generate-config
```

This will allow the user to configure the Jupyterhub server to act has desired.

Additionally, to be able to embed a Jupyterhub instance into the Data Manager it is necessary to add the following line to both configuration file of both Jupyterhub and Jupyter Notebook:

1. For Jupyterhub configuration file this line must be added:

```
c.JupyterHub.tornado_settings = {'headers':
{'Content-Security-Policy':"frame-ancestors*",'Access-Control-Allow-Origin': "*"}}
```

2. In the case that there is not a configuration file created for Jupyter Notebook, please execute the following command:

```
$ jupyter notebook --generate-config
```

3. For Jupyter Notebooks configuration file this line must be added:

```
c.NotebookApp.allow_origin = '*'

c.NotebookApp.tornado_settings = { 'headers': { 'Content-Security-Policy': "child-src
*" }}
```

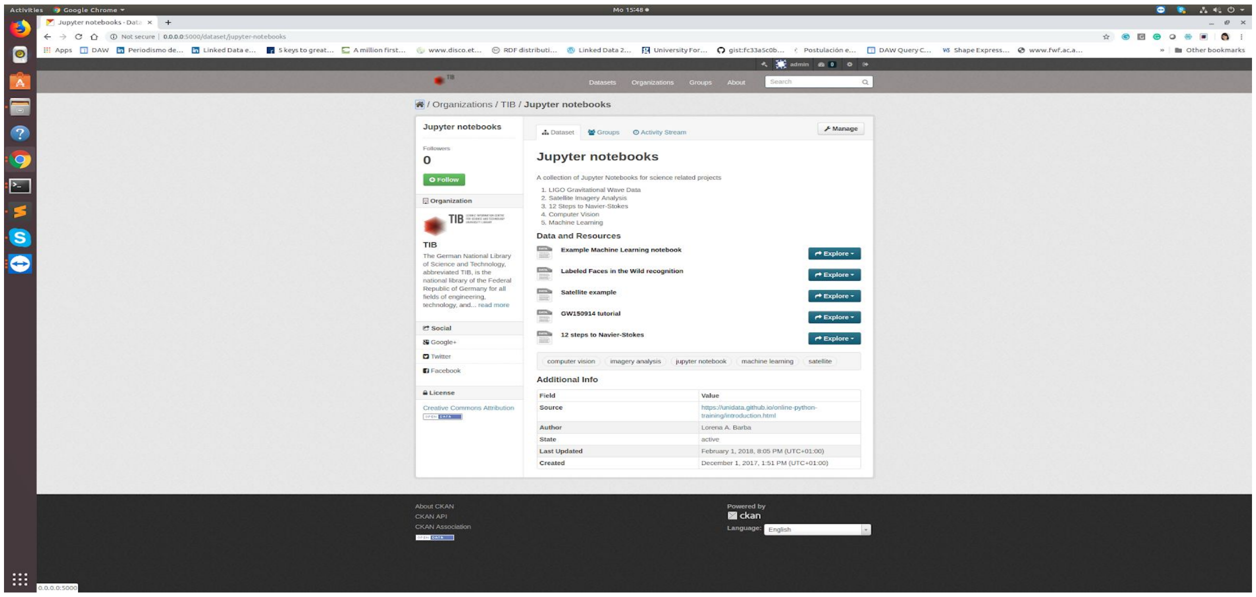
Note: TIB data manager source folder does have an example of a jupyterhub configuration file but it is recommended that the user generates their own following the previous steps.

Updating existing Jupyter Notebooks

The data manager does have examples of jupyter notebooks but these must be updated according to the computer hosting it.

1. Open Jupyterhub in the browser (localhost:8000).
2. Search for the TIB data manager source folder in the file system and enter the files folder.
3. Open the file that is going to be updated.
4. Copy the file link.
5. In the data manager click on the dataset tab and then choose “Jupyter Notebook” dataset.

6. Select the jupyter notebook that will be updated.



/ Organizations / TIB / **Jupyter notebooks**

Jupyter notebooks

Followers
0

Organization

TIB

The German National Library of Science and Technology, abbreviated TIB, is the national library of the Federal Republic of Germany for all fields of engineering, technology, and... read more

Social
Google+
Twitter
Facebook

License
Creative Commons Attribution

Jupyter notebooks

A collection of Jupyter Notebooks for science related projects

1. LIGO Gravitational Wave Data
2. Satellite Imagery Analysis
3. 12 Steps to Navier-Stokes
4. Computer Vision
5. Machine Learning

Data and Resources

- **Example Machine Learning notebook**
- **Labeled Faces in the Wild recognition**
- **Satellite example**
- **GW150914 tutorial**
- **12 steps to Navier-Stokes**

computer vision · imagery analysis · jupyter notebook · machine learning · satellite

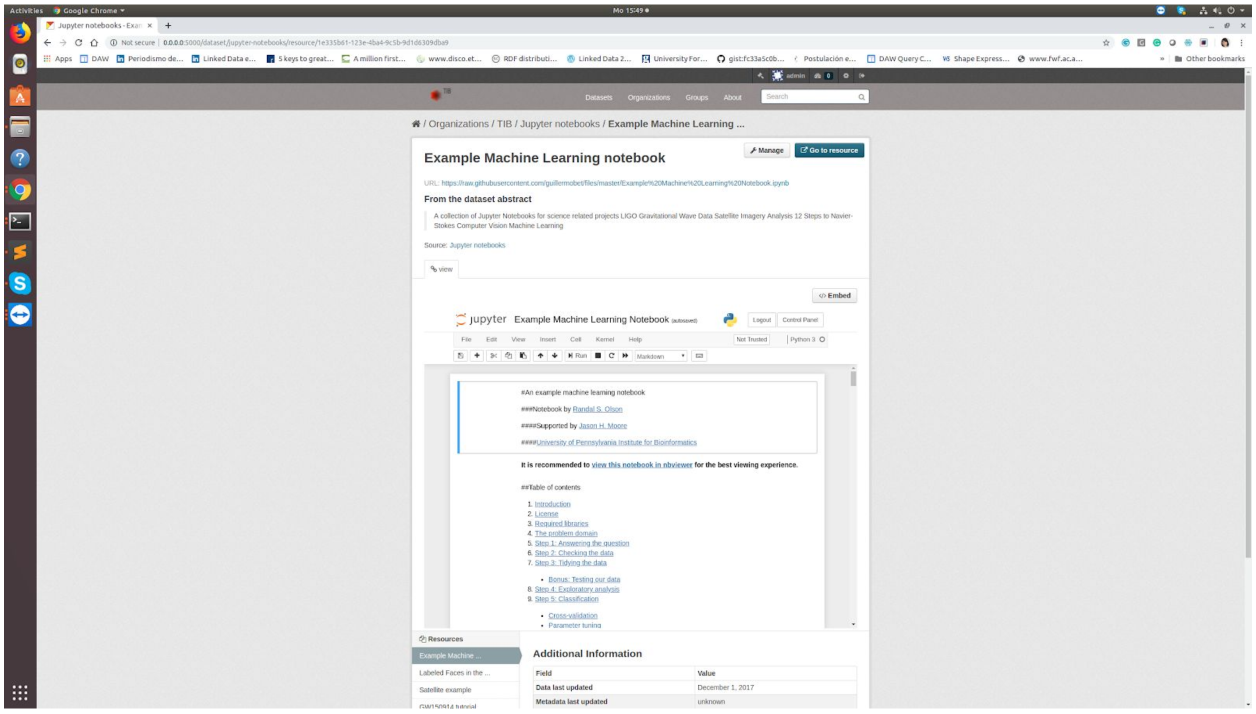
Additional Info

Field	Value
Source	https://unidata.github.io/online-python-training/introduction.html
Author	Lorena A. Barba
State	active
Last Updated	February 1, 2018, 8:05 PM (UTC+01:00)
Created	December 1, 2017, 1:51 PM (UTC+01:00)

About CKAN
CKAN API
CKAN Association

Powered by ckan
Language: English

7. Click the “manage” option.



/ Organizations / TIB / Jupyter notebooks / **Example Machine Learning ...**

Example Machine Learning notebook

URL: https://raw.githubusercontent.com/guillermobet/files/master/Example%20Machine%20Learning%20Notebook.ipynb

From the dataset abstract

A collection of Jupyter Notebooks for science related projects LIGO Gravitational Wave Data Satellite Imagery Analysis 12 Steps to Navier-Stokes Computer Vision Machine Learning

Source: Jupyter notebooks

jupyter Example Machine Learning Notebook (autosaved)

#An example machine learning notebook

###Notebook by Randal S. Olson

####Supported by Jason H. Moore

####University of Pennsylvania Institute for Bioinformatics

It is recommended to view this notebook in nbviewer for the best viewing experience.

##Table of contents

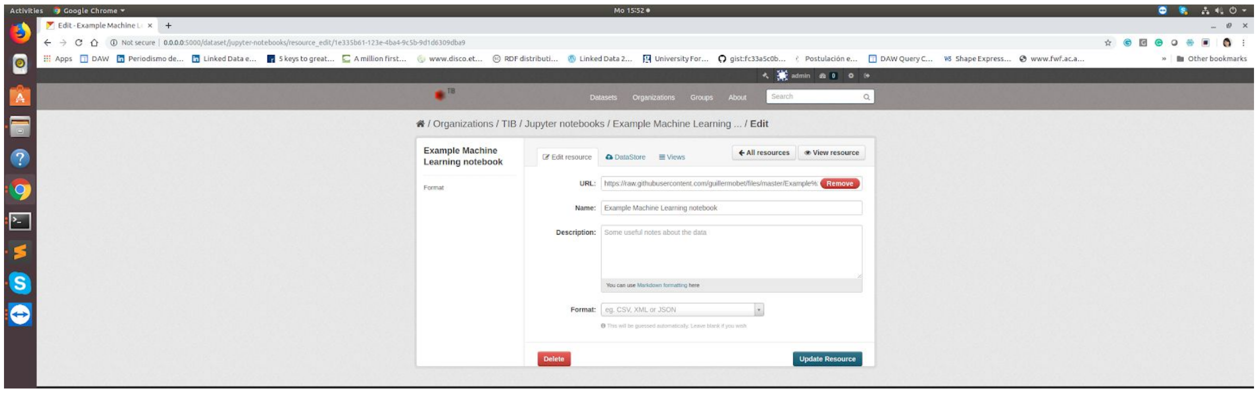
1. Introduction
2. License
3. Required libraries
4. The problem domain
5. Step 1: Answering the question
6. Step 2: Checking the data
7. Step 3: Tidying the data
    - Bonus: Testing our data
8. Step 4: Exploratory analysis
9. Step 5: Classification
    - Cross-validation
    - Parameter tuning

Resources
Example Machine ...
Labeled Faces in the ...
Satellite example
GW150914 tutorial

Additional Information

Field	Value
Data last updated	December 1, 2017
Metadata last updated	unknown

8. Click on “Views” and then click on “view”.

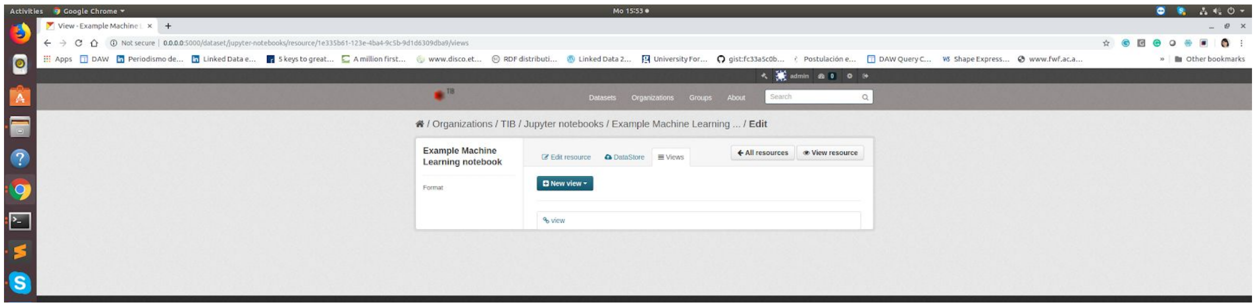

/ Organizations / TIB / Jupyter notebooks / Example Machine Learning ... / Edit
Example Machine Learning notebook
Edit resource
DataStore
Views
All resources
View resource
Remove
Delete
Update Resource

URL: https://raw.githubusercontent.com/guillermobet/files/master/Example%

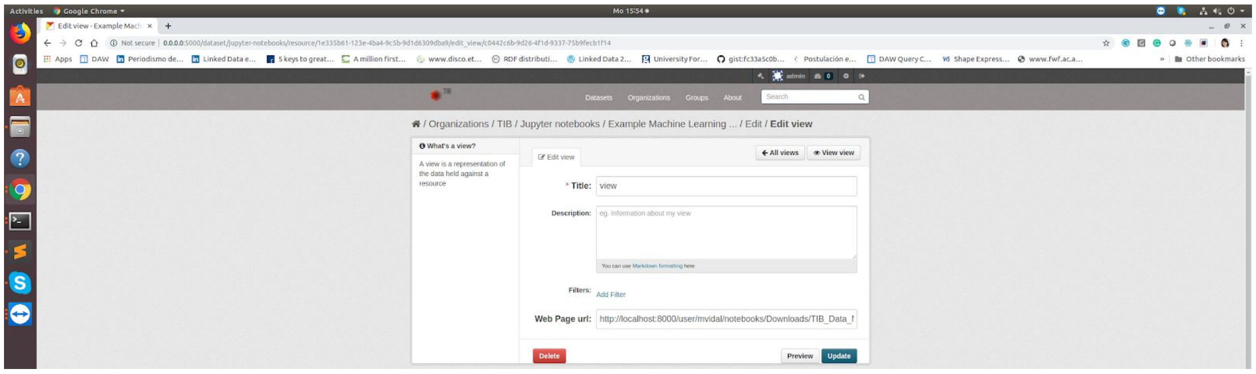
Name: Example Machine Learning notebook

Description: Some useful notes about the data

Format: eg. CSV, XML or JSON


/ Organizations / TIB / Jupyter notebooks / Example Machine Learning ... / Edit
Example Machine Learning notebook
Edit resource
DataStore
Views
All resources
View resource
New view
view

9. Update the file link with the new link.


/ Organizations / TIB / Jupyter notebooks / Example Machine Learning ... / Edit / Edit view
What's a view?
A view is a representation of the data held against a resource
Edit view
All views
View view
Add Filter
Delete
Preview
Update

Title: view

Description: eg. Information about my view

Web Page url: http://localhost:8000/user/mvidal/notebooks/Downloads/TIB_Data_

Note: The user must be logged in to do these changes (default user username:admin password:admin).

Customizing The Data Manager

Changing The Data Manager Logo

To be able to change the logo of the header of the Data Manager, the user must modify or create the `header.html` file:

```
ckanext-example_theme/
  ckanext/
    example_theme/
      templates/
        header.html
```

Insert the extension of the file containing the new logo:

```
{% ckan_extends %}

{% block header_logo %}

        &nbsp;
        &nbsp;
        &nbsp;
        &nbsp;
        <img src="/base/images/ckan-logo.png" style="margin-top:10px" alt="CKAN" 
title="CKAN">
        &nbsp;
        &nbsp;
        <img 
src="https://www.tib.eu/typo3conf/ext/tib_tmpl_bootstrap/Resources/Public/images/TIB_
Logo_en.png" style="margin-top:10px;max-width:30%" alt="TIB" title="TIB">

{% endblock %}

{% block header_site_search %}
  <form class="section site-search simple-input" action="{% url_for 
controller='package', action='search' %}" method="get">
    <div class="field">
```

```
      <label for="field-sitewide-search">{% block header_site_search_label %}{{
_('Search Datasets') }}{% endblock %}</label>
      <input id="field-sitewide-search" type="text" name="q" placeholder="{{
_('Search') }}" />
      <button class="btn-search" type="submit" style="top:16px"><i class="fa
fa-search"></i></button>
    </div>
  </form>
{% endblock %}
```

Changing The Data Manager Header Color

To be able to change the color of the header of the Data Manager, head to the extensions directory and modify or create a new CSS file in the public directory.

```
ckanext-example_theme/
  ckanext/
    example_theme/
      public/
        example_theme.css
```

Add this CSS into the example_theme.css file, to change the color of Data Manager header:

```
.account-masthead {
    background-color: rgb(40, 40, 40);
}

*Adjust the values to obtain the desired color.
```

To make the Data Manager use the custom CSS it is necessary to override the `base.html` template, this is the base template which the templates for all Data Manager pages extend, so if we include a CSS file in this base template then the file will be included in every page of the Data Manager site. Create or the file:

```
ckanext-example_theme/
  ckanext/
    example_theme/
      templates/
        base.html
```

and put this Jinja code in it:

```
{% ckan_extends %}

{% block styles %}
  {{ super() }}
  <link rel="stylesheet" href="/example_theme.css" />
{% endblock %}
```