

# DataShield/Opal Notes on Deployment, Tools & Data

**Dr. Ulrich Harttig**  
Team Leader of the Human Study Center (HSZ) Data Center



ENPADASI Workshop WP3  
Bari, 08-09 June 2016



# TOC

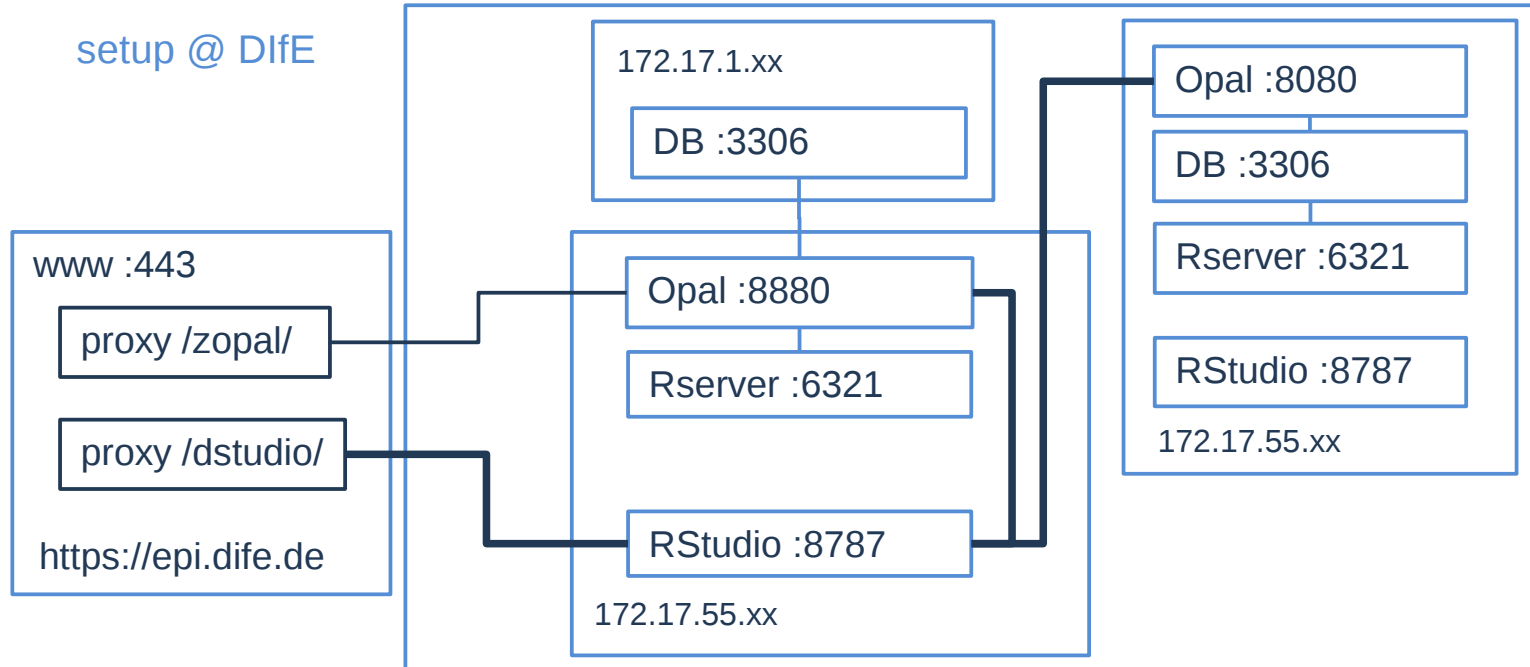
---



- Opal Deployment
  - Web access via proxy
  - Proxy setup example
- Data import
  - Data Dictionary
  - Data upload
- Opal Tools
  - CLI access
  - REST interface
- DataShield Proxy Demo setup

Access via proxy allows

- Deployment in standard environment, no opening of additional firewall holes
- Integration in existing infrastructure
- Flexible backend structure



<https://epi.dife.de/zopal/> (external access via proxy, IP unrestricted, OPAL accounts needed)

<https://epi.dife.de/dstudio/> (Rstudio, external access via proxy, restricted: special account needed)



# Opal Deployment - Proxy setup example



- using Apache 2.4 with modules: proxy proxy\_connect proxy\_fcgi proxy\_html proxy\_http proxy\_wstunnel, rewrite, deflate
- part of an existing web server configuration
- for a setup of a complete Virtual Host dedicated to Opal see <http://wiki.obiba.org/display/OPALDOC/Opal+Configuration+Guide>

```
# disable forward proxy
ProxyRequests Off

<Proxy "*">
    SetOutputFilter INFLATE;proxy-html;DEFLATE
    Require all granted
</Proxy>

# for opal
# this works for jehanne.dyndns.org/opal/ and https://epi.dife.de/zopal/
RewriteRule "^/ws/(.*)$" "http://zimt:8880/ws/$1" [P]
<Location "/zopal/">
    LogLevel info rewrite:trace5 proxy:trace4
    ProxyPass "http://zimt:8880/"
    ProxyPassReverse "http://zimt:8880/"
    ProxyPreserveHost On
    RequestHeader set X-Forwarded-HTTPS ON
    <RequireAny>
    ##         Require ip 172.17.55.31 193.175.234.39 192.168.112
    ##         Require host dife.de
    ##     </RequireAny>
    Require all granted
</Location>

# setup for the Rstudio/Server proxy
ProxyPassMatch ^/dstudio/p/([0-9]+)/((websocket|.*websocket)/$ ws://localhost:8787/p/$1/$2/
<Location "/dstudio/">
    ProxyPass http://localhost:8787/
    ProxyPassReverse http://localhost:8787/

    AuthType Basic
    AuthName "DSstudio"
    AuthBasicProvider file
    AuthUserfile "/etc/apache2/security/ds.auth"
    <RequireAll>
        Require all granted
        Require user harttig
        Require ip 193.175.234.39 172.17.55.31
    </RequireAll>
</Location>
```

tested on 2 separate instances

OPTIONAL for debugging rewrite and proxy modules

URL of actual opal instance

IP/host based access restriction (recommended) e.g. IP of the analysis machine(s)



# Data Dictionary

---



- **a structured documentation of detailed information for each variable in a study**
- enhance comprehension and highlight heterogeneity across different studies.
- can be created before or after data import
- create and update Data Dictionary information directly in Opal or via ex/import of an Excel template

The template consists of 2 spreadsheets

- **Variables Spreadsheet**
  - is used to define variable attributes.
- **Categories Spreadsheet**
  - is used to define categories for the categorical variables

see also

<http://wiki.obiba.org/display/OPALDOC/How+to+install+and+use+Opal+and+DataSHIELD+for+Data+Harmonization+and+Federated+Analysis>



# Data Dictionary



## Variables Spreadsheet

The **name** column is mandatory.

Other columns can be deleted; default values are used for certain columns (see below). Add any other columns to specify customized attributes for your variables. The built-in column names are reserved words and should not be used as customized attributes. Specify the attribute language by adding the language code to the end of the attribute name (e.g. label:en, label:fr)

Default columns	
table	The table name the variable will be added to. Default value is Table.
name	The variable name (mandatory).
valueType	The value type of the variable. Default value is text.
entityType	Opal can store data on different entities such as Participant, Instrument, Area, Drug, etc. Default value is Participant.
unit	The unit in which values expressed (e.g. cm, kg ...).
contentType	The mime type of the variable to help applications to display documents (e.g. image/jpeg, application/excel ...).
repeatable	1 if repeatable, 0 if not. (eg. Three measures of blood pressure). Default value is 0.
occurrenceGroup	Name of a repeatable variable group (e.g. The group [measure value, measure date] is a group of variables that can be repeated)
referencedEntityType	If the variable values are entity identifiers, this is the type of the entities that are referenced
index	Position or weight of the variable in the list of variables of the table for ordering. Default value is 0.
label	Label of the variable. Can be localized (e.g. label:en, label:fr ...).
alias	Alternative name for the variable, usually used for defining a shorter name for the variable

## Categories Spreadsheet

Columns **variable** and **name** are mandatory.

Default columns	
table	The table name the variable will be added to. Default value is Table.
variable	The variable name the category belongs to (mandatory).
name	The category name (mandatory).
missing	Some categories are interpreted as missing answers (e.g. 'Don't know', 'Prefer not to answer'). Use 1 for missing and 0 for not. missing (normal answer). Default value is 0.
label	Label of the category. Can be localized (e.g. label:en, label:fr ...).



# Data Import



1. step: upload to OPAL file system - into personal (per user) or project filespace
2. step: import from OPAL file space into Project/Table  
includes indexing = might take a while,  
progress of longer running tasks can be  
monitored via 'Tasks' tab

ID	Type	User	Start	End	Status	Actions
1	import	demo01	Jun 6 2016 4:22 PM	Jun 6 2016 4:24 PM [2 minutes]	●	<a href="#">Log</a>

## Data sources & Types

- **CSV Datasource**  
"delimiter separated values" format (default delimiter being comma). The first column will represent the entity identifiers and the subsequent column names will identify variables. Each row of the file (except the first row) are the values for one entity. The entity identifier must be unique: there cannot be two rows starting with the same identifier.
- **Opal Archive Datasource**  
This datasource comes as a .zip file containing a folder for each table having: the full data dictionary in a XML file, a XML data file per entity. This is the file format used when exporting data from Onyx.
- **SPSS Datasource**  
The SPSS source file must be a valid non-compressed binary file with a .sav extension. In Opal an SPSS file represents a table and its variables are used as the table's data dictionary.
- **Excel Datasource**  
Opal supports both Excel 97 and Excel 2007 formats. Only for data dictionary import, NOT data import
- **Opal Datasource**  
Opal datasource allows one Opal server to connect to a *remote Opal server*. This can be useful when syncing datasources in different Opal instances. (No working example yet!)

Examples see in CLI Section

see also <http://wiki.obiba.org/display/OPALDOC/Datasource+Types>



# Opal Tools - CLI access



## Commandline access to Opal functions via Python Client - all functions of UI (with REST)

```
ulrich@zimt:~/projects/datashield/opal-datashield-vm$ opal -h
```

```
usage: opal [-h]
            {dict,data,entity,file,import-opal,import-csv,import-xml,import-spss,import-limesurvey,import-sql,import-ids,import-ids-map,export-xml,export-csv,export-sql,copy-table,delete-table,user,group,perm-project,perm-datasource,perm-table,perm-variable,perm-r,perm-datashield,perm-system,system,rest,encrypt,decrypt}
            ...
```

Opal command line.

optional arguments:

-h, --help show this help message and exit

sub-commands:

```
{dict,data,entity,file,import-opal,import-csv,import-xml,import-spss,import-limesurvey,import-sql,import-ids,import-ids-map,export-xml,export-csv,export-sql,copy-table,delete-table,user,group,perm-project,perm-datasource,perm-table,perm-variable,perm-r,perm-datashield,perm-system,system,rest,encrypt,decrypt}
```

```
dict Available sub-commands. Use --help option on the sub-command for more details.
data Query for data dictionary.
entity Query for data.
file Query for entities (Participant, etc.).
import-opal Manage Opal file system.
import-csv Import data from a remote Opal server.
import-xml Import data from a CSV file.
import-spss Import data from a ZIP file.
import-limesurvey Import data from a SPSS file.
import-sql Import data from a LimeSurvey database.
import-ids Import data from a SQL database.
import-ids-map Import system identifiers.
import-ids-map Import identifiers mappings.
export-xml Export data to a zip of Opal XML files.
export-csv Export data to a folder of CSV files.
export-sql Export data to a SQL database.
copy-table Copy a table into another table.
delete-table Delete some tables.
user Manage users.
group Manage groups.
perm-project Apply permission on a project.
perm-datasource Apply permission on a datasource.
perm-table Apply permission on a set of tables.
perm-variable Apply permission on a set of variables.
perm-r Apply R permission.
perm-datashield Apply DataSHIELD permission.
perm-system Apply system permission.
system Query for system status and configuration
rest Request directly the Opal REST API, for advanced users.
encrypt Encrypt string using Opal's secret key.
decrypt Decrypt string using Opal's secret key.
```





# Opal Tools - CLI access - examples



# download folder as ZIP file

```
opal file -v -o http://zimt:8880 -u administrator -p xxxxx -dl  
/home/administrator/testdata/CNSIM > CNSIM.zip
```

# upload data file to project directory - NOT data import

```
opal file -j -v -o https://epi.dife.de/zopal/ -u administrator -p xxxxx -up  
data/opal/testdata/CNSIM/CNSIM3.csv /projects/CNSIM/
```

# import data from project directory to table space - data import

```
opal import-csv -j -v -o https://epi.dife.de/zopal/ -u demo01 -p xxxxxx -pa /projects/CNSIM/CNSIM3.csv -s ','  
-d CNSIM -t CNSIM3 -ty Participant
```

```
{  
  "command": "import",  
  "commandArgs": "import --destination CNSIM",  
  "id": 1,  
  "messages": [  
    {  
      "msg": "Job started.",  
      "timestamp": 1465222955297  
    },  
    {  
      "msg": " Importing tables [CNSIM3] in CNSIM ...\\n",  
      "timestamp": 1465222955298  
    }  
  ],  
  "name": "import",  
  "owner": "demo01",  
  "project": "CNSIM",  
  "startTime": "2016-06-06T16:22:35.297+0200",  
  "status": "IN_PROGRESS"  
}
```

further progress can be checked  
via 'Task' tab of UI



## # transfer data from one opal instance to another

```
opal import-opal -j -v -o https://jehanne.dyndns.org/opal/ -u demo01 -p aileeGhe -ro  
https://epi.dife.de/zopal/ -ru demo01 -rp aileeGhe -rd CNSIM -d CNSIM
```

2016-06-06 23:46:24,665 [qtp2025390719-27481] INFO org.obiba.opal.rest.client.magma.OpalJavaClient -  
Connecting to Opal: https://epi.dife.de/zopal/ws/  
2016-06-06 23:46:24,971 [qtp2025390719-27481] ERROR org.obiba.opal.rest.client.magma.RestDatasource -  
Unexpected error while communicating with Opal server: Host name may not be null -> Cause: ???

see also

<http://wiki.obiba.org/display/OPALDOC/Opal+Python+User+Guide>



# Opal REST Interface



## Programmatic access to Opal - any web client - advanced users

```
ulrich@zimt:~/projects/datashield/opal-datashield-vm$ opal rest --help
```

```
usage: opal rest [-h] [--opal OPAL] [--user USER] [--password PASSWORD]
                [--ssl-cert SSL_CERT] [--ssl-key SSL_KEY] [--verbose]
                [--method METHOD] [--accept ACCEPT]
                [--content-type CONTENT_TYPE] [--json]
                ws
```

### positional arguments:

```
ws                Web service path, for instance:
                  /datasource/[PROJECT]/table/[TABLE NAME]/variable/[VARIABLE]
```

### optional arguments:

```
-h, --help                show this help message and exit
--opal OPAL, -o OPAL      Opal server base url
--user USER, -u USER    User name
--password PASSWORD, -p PASSWORD
                           User password
--ssl-cert SSL_CERT, -sc SSL_CERT
                           Certificate (public key) file
--ssl-key SSL_KEY, -sk SSL_KEY
                           Private key file
--verbose, -v             Verbose output
--method METHOD, -m METHOD
                           HTTP method (default is GET, others are POST, PUT,
                           DELETE, OPTIONS)
--accept ACCEPT, -a ACCEPT
                           Accept header (default is application/json)
--content-type CONTENT_TYPE, -ct CONTENT_TYPE
                           Content-Type header (default is application/json)
--json, -j               Pretty JSON formatting of the response
```

see also <http://wiki.obiba.org/display/OPALDOC/REST+API+Command>



# Opal REST Interface - query information



```
opal rest -v -o https://epi.dife.de/zopal/ -u demo01 -p xxxxxx -m GET -j /datasource/CNSIM/table/CNSI
```

```
{
  "datasourceName": "CNSIM",
  "entityType": "Participant",
  "link": "/datasource/CNSIM/table/CNSIM",
  "name": "CNSIM",
  "timestamps": {
    "created": "2016-01-31T15:30:06.000+0100",
    "lastUpdate": "2016-01-31T15:33:28.000+0100"
  }
}
```

```
opal rest -v -o https://epi.dife.de/zopal/ -u demo01 -p xxxxx -m GET -j \
/datasource/mytpr/table/eveAge/variable/PROBAND_ALTER
```

```
{
  "attributes": [
    {
      "name": "script",
      "value": "if($('PROBAND_ALTER').any(-99).not().value()){\\n  $('PROBAND_ALTER').value();\\n }\\nelse {\\n  ;\\n}"
    },
    {
      "name": "derivedFrom",
      "namespace": "opal",
      "value": "/datasource/mytpr/table/BMBFeve/variable/PROBAND_ALTER"
    }
  ],
  "categories": [
    {
      "isMissing": true,
      "name": "miss"
    }
  ],
  "entityType": "Participant",
  "index": 0,
  "isRepeatable": false,
  "mimeType": "",
  "name": "PROBAND_ALTER",
  "parentLink": {
    "link": "/datasource/mytpr/table/eveAge",
    "rel": "eveAge"
  },
  "referencedEntityType": "",
  "unit": "",
  "valueType": "integer"
}
```



# Opal REST Interface - manipulate Opal



These are calls from the installation scripts preparing **databases** and default **projects** from pre-defined \*.json files

```
opal rest -o http://localhost:8080 \  
-u administrator -p $OPAL_PWD \  
-m POST /system/databases \  
--content-type "application/json" < "${DATABASES_DIR}/${i}.json"
```

```
opal rest \  
-o https://localhost:8883 \  
-u administrator \  
-p $OPAL_PWD \  
-m POST /projects \  
--content-type "application/json" < CNSIM.json
```

```
{  
  "name": "CNSIM",  
  "title": "CNSIM",  
  "description": "Simulated data",  
  "database": "sqlldb"  
}
```

CNSIM.json

see also

<https://wikis.bris.ac.uk/display/DSDEV/Importing+data+into+Opal+with+the+API>



# Proxy Demo - single instance



**Rstudio** - <https://epi.dife.de/dstudio/>

**DS Tutorial** - <https://wikis.bris.ac.uk/display/DSDEV/DataSHIELD+users+tutorial>

*#load opal/datashield libraries*

```
library(opal)
```

```
library(dsBaseClient)
```

```
library(dsStatsClient)
```

```
library(dsGraphicsClient)
```

```
library(dsModellingClient)
```

*# login details*

```
server <- c("zimtproxy")
```

*# access via proxy to be accessible from Rstudio Server or other client*

```
url <- c("https://epi.dife.de/zopal/")
```

```
user <- c("demo01")
```

```
password <- c("aileeGhe")
```

```
table <- c("CNSIM.CNSIM")
```

```
logindata <- data.frame(server,url,user,password,table)
```

*# Create an 'opals' object by passing the 'logindata' data frame to the datashield.login function*

```
opals <- datashield.login(logins=logindata, assign = TRUE)
```

```
ds.dim(x='D') # dimension of data
```

```
ds.quantileMean(x='D$LAB_HDL') # Quantiles of the data
```

```
ds.table1D(x='D$GENDER')
```

```
ds.histogram(x='D$LAB_HDL')
```

```
ds.heatmapPlot(x='D$LAB_TSC', y='D$LAB_HDL')
```

## Errors:

```
> ds.histogram(x='D$LAB_HDL', type='split')
```

Fehler: Command 'rangeDS(D\$LAB\_HDL)' failed on 'zimtproxy': Error while evaluating 'dsGraphics::rangeDS( D\$LAB\_HDL)'

-> check the R SERVER instance ([OPAL-URL]/ui/index.html#!admin/!datashield) if ds packages have been loaded -> here , the package **dsGraphics** is missing



# Q & A

---

DIFE