

COMPASS API

Matouš Jandek
matous.jandek@cern.ch

Faculty of Nuclear Sciences and Physical Engineering
Czech Technical University

12th July, 2018



COMPASS Data Storage

- COMPASS experiment utilizes two types of data storage.

COMPASS database

- MySQL database
- Designed by COMPASS members

Network folders and file servers

- Directories on COMPASS servers
- Services provided by CERN IT department



Example of usage

- Example of typical database query:

```
MariaDB [(none)]> SELECT nbspill, nbvt, recordingfg, shiftid, title, comments, daq_dt, veto_dt, veto_rdt,
-> DATE_FORMAT(starttime, '%Y-%M-%D %H:%i:%s') AS starttime,
-> DATE_FORMAT(stoptime, '%Y-%M-%D %H:%i:%s') AS stoptime,
-> rt.description AS rt_description, rt.keyword AS rt_keyword,
-> t.pbtypewd, t.errdetwd, bv.sm1, bv.sm2, bv.ionmuon,
-> DATE_FORMAT(bv.mestime, '%Y-%M-%D %H:%i:%s') AS bv_mestime,
-> bv.t6current, t6.desc AS t6head, sf.name AS sps_file, sf.mode AS sps_mode,
-> sf.energy AS sps_energy, tg.uppol, tg.centralpol, tg.downpol, tg.coil4, tg.solencur, tg.dipolcur,
-> DATE_FORMAT(tg.magtime, '%Y-%M-%D %H:%i:%s') AS magtime, DATE_FORMAT(tg.mestime, %s) AS pol_mestime,
-> ar.ROBelems, ae.EBelems, fi.filtername, tr.trigelems, ed.runnb AS errorDump,
-> coll1_jaw1, coll1_jaw2, coll2_jaw1, coll2_jaw2, coll3_jaw1, coll3_jaw2, coll4_jaw1, coll4_jaw2,
-> DATE_FORMAT(co.stamp, '%Y-%M-%D %H:%i:%s') AS coll_stamp, root.filename AS root_filename, t.period
-> FROM tb_run t
-> INNER JOIN tb_runtpe rt ON t.runtpeid = rt.runtpeid
-> INNER JOIN (tb_beamvalues bv INNER JOIN tb_t6heads t6 ON bv.t6head = t6.t6headid)
-> ON t.bvalueid = bv.bvalueid
-> INNER JOIN tb_spsfiles sf ON t.spsfileid = sf.spsfileid
-> INNER JOIN tb_target tg ON t.targetid = tg.targetid
-> INNER JOIN tb_actROB ar ON t.actROBId = ar.actROBId
-> INNER JOIN tb_actEB ae ON t.actEBid = ae.actEBid
-> INNER JOIN tb_filterinfo fi ON t.filterid = fi.filterid
-> INNER JOIN tb_trigger tr ON t.triggerid = tr.triggerid
-> LEFT JOIN tb_errorDump ed ON t.runnb = ed.runnb
-> LEFT JOIN tb_collimators co ON t.runnb = co.runnb
-> LEFT JOIN tb_rootfiles root ON t.runnb = root.runnb
-> WHERE t.runnb = 222222;
```



Database Access Methods

- Database provides no views or procedures
- Manually written database queries are used by applications
- Each application uses individually written queries
- Results in repetitive code
 - More time required to develop new features
 - More difficult maintenance of existing software
 - Higher probability of errors
 - Inability to change database structure without adapting all applications

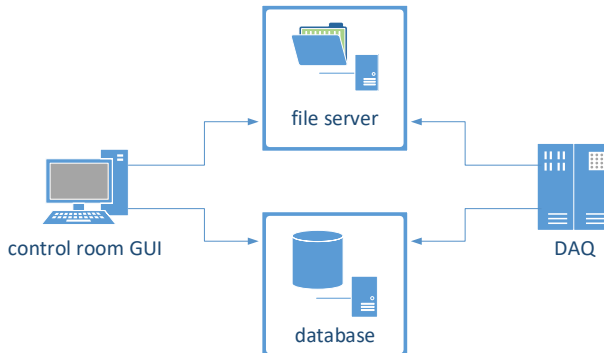


File Storage Directories

- Directory information is obtained from database
- This procedure is not unified
- Files are often bound to specific database records, but are stored separately
 - Risk of desynchronization between database and filesystem
 - File-related data SELECT SQL – risk of retrieving incomplete data
 - File-related data INSERT SQL – risk of storing incomplete data and loss of data integrity



COMPASS Data Storage



Possible Solution – Database Views

- Database problems may be solved by implementing SQL procedures and views
- Solves duplicity of code that is used to access database
- Drawbacks:
 - No control over file storage – does not solve file-database synchronization
 - Lack of ability to affect any other computer systems
- Advisable to implement, but complete solution should be universal



Possible solution – Application Programming Interface

- API acts as an abstraction layer between COMPASS data and applications
- Applications may use the API to operate on data
- Required data operations:
 - Create data
 - Modify existing data
 - Read data
 - Delete data



- COMPASS API is designed as RESTful API
- REST is a system architecture principle
- Properties of RESTful API-based systems
 - REST is centered around “resource” - single piece of information
 - Client-server structure
 - Request-response communication
 - Clients request resource from server
 - Server sends back resource representation in response



Implementation – HTTP

- Typical RESTful API implementation uses HTTP protocol
- HTTP request contains “method” keyword

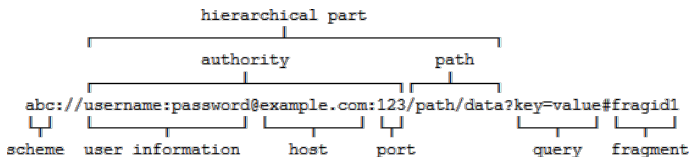
```
GET http://wwwcompass.cern.ch/  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8  
Accept-Encoding: gzip, deflate  
Accept-Language: en-US,en;q=0.5  
Cache-Control: max-age=0  
Connection: keep-alive  
Host: wwwcompass.cern.ch  
If-Modified-Since: Tue, 06 Mar 2018 06:49:23 GMT  
Upgrade-Insecure-Requests: 1  
User-Agent: Mozilla/5.0
```

- HTTP methods may be mapped to request types



Implementation – URI

- Identification of resources done by URI



- example: `http://www.jinr.ru`
- Resource is identified by “path” section of URI
- examples:
`http://apiserver/resource,`
`http://apiserver/other_resource`

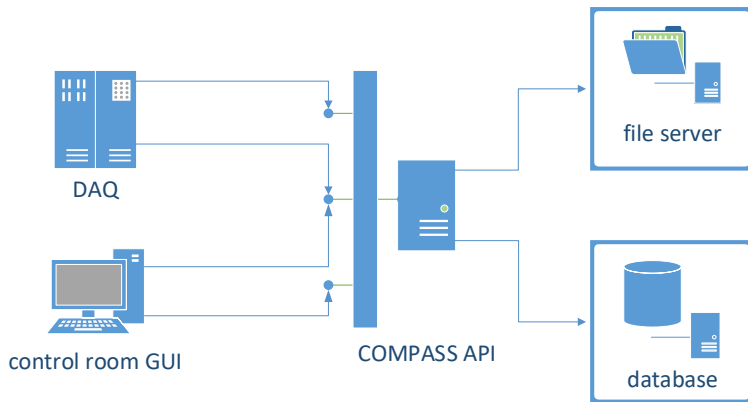


Implementation

- COMPASS API is a RESTful API server
- Server implemented in C++ with the use of Qt framework – same technologies as COMPASS iFDAQ software
- Modular structure – request type definitions are loaded at runtime from plugin modules
- Server routes all requests to routines in corresponding plugin
- Plugin is then responsible for executing necessary database queries and file operations, and constructs the response to the client



COMPASS data storage with API



- Unification of data interaction procedures
 - Background logic in single program
 - Easier to maintain and implement new features
- The design allows to extend the functionality of COMPASS API to execute variety of tasks



Example of usage

- Example of database query:

```
MariaDB [(none)]> SELECT nbspill, nbvt, recordingfg, shiftid, title, comments, daq_dt, veto_dt, veto_rdt,
-> DATE_FORMAT(starttime, '%Y-%M-%D %H:%i:%s') AS starttime,
-> DATE_FORMAT(stoptime, '%Y-%M-%D %H:%i:%s') AS stoptime,
-> rt.description AS rt_description, rt.keyword AS rt_keyword,
-> t.pbtypewd, t.errdetwd, bv.sm1, bv.sm2, bv.ionmuon,
-> DATE_FORMAT(bv.mestime, '%Y-%M-%D %H:%i:%s') AS bv_mestime,
-> bv.t6current, t6.desc AS t6head, sf.name AS sps_file, sf.mode AS sps_mode,
-> sf.energy AS sps_energy, tg.uppol, tg.centralpol, tg.downpol, tg.coil4, tg.solencur, tg.dipolcur,
-> DATE_FORMAT(tg.magtime, '%Y-%M-%D %H:%i:%s') AS magtime, DATE_FORMAT(tg.mestime, %s) AS pol_mestime,
-> ar.ROBelems, ae.EBelems, fi.filtername, tr.trigelems, ed.runnb AS errorDump,
-> coll1_jaw1, coll1_jaw2, coll2_jaw1, coll2_jaw2, coll3_jaw1, coll3_jaw2, coll4_jaw1, coll4_jaw2,
-> DATE_FORMAT(co.stamp, '%Y-%M-%D %H:%i:%s') AS coll_stamp, root.filename AS root_filename, t.period
-> FROM tb_run t
-> INNER JOIN tb_runtpe rt ON t.runtpeid = rt.runtpeid
-> INNER JOIN (tb_beamvalues bv INNER JOIN tb_t6heads t6 ON bv.t6head = t6.t6headid)
-> ON t.bvalueid = bv.bvalueid
-> INNER JOIN tb_spsfiles sf ON t.spsfileid = sf.spsfileid
-> INNER JOIN tb_target tg ON t.targetid = tg.targetid
-> INNER JOIN tb_actROB ar ON t.actROBid = ar.actROBid
-> INNER JOIN tb_actEB ae ON t.actEBid = ae.actEBid
-> INNER JOIN tb_filterinfo fi ON t.filterid = fi.filterid
-> INNER JOIN tb_trigger tr ON t.triggerid = tr.triggerid
-> LEFT JOIN tb_errorDump ed ON t.runnb = ed.runnb
-> LEFT JOIN tb_collimators co ON t.runnb = co.runnb
-> LEFT JOIN tb_rootfiles root ON t.runnb = root.runnb
-> WHERE t.runnb = 222222;
```



Example of usage

- Solution:

- Move the SQL query to COMPASS API
- In the computer program, send an HTTP request
- Example using C++ and Qt

```
1 QNetworkAccessManager manager;  
2 QUrl url("http://compassapi/runs?id=222222");  
3 QNetworkRequest request(url);  
4 QNetworkReply reply;  
5  
6 reply = manager.get(request);
```



Thank you for your attention

