# CDS Libraries for UWS, ADQL and TAP implementation

**Author:** Grégory Mantelet
**Presented by:** André Schaaff

# Overview

**TAP Library**

- execute ADQL queries
- describe table metadata
- let uploading tables

+

**ADQL Library**

- parse ADQL
- transform ADQL
- translate in SQL

+

**UWS Library**

- execute jobs in background
- manage execution queue
- list and summarize jobs

# UWS Library

- Version 3.0 -> 3.1 *(mainly for bug corrections)*
- 2 things to do to use it:
  - Extend AbstractJob: *to say what a job is supposed to do (i.e. cross-match, executing adql query, ...)*
  - Write a servlet which forwards all requests to UWS

# Example

```
public class UWSTimers extends HttpServlet {
  protected BasicUWS<JobChrono> uws = null;

  public void init(ServletConfig config) {
    super.init(config);
    try{
      // Create the UWS [required]:
      uws = new BasicUWS<JobChrono>(JobChrono.class);
      // Create the job list "timers" [required]:
      uws.addJobList(new JobList<JobChrono>("timers"));
    }catch(UWSException ex){
      throw new ServletException(ex);
    }
  }


  protected void service(HttpServletRequest req,
  HttpServletResponse resp) {
    try{
      // Forward the request to the uws [required]:
      uws.executeRequest(req, resp);
    }catch(UWSException uwsEx){
      // Display properly the caught UWSException:
      resp.sendError(uwsEx.getHttpErrorCode(),
      uwsEx.getMessage());
    }
  }
```

```
public class JobChrono extends AbstractJob {
 protected int time = 0;
 public JobChrono(Map<String, String> lstParam) {
    super(lstParam);
 }

 protected boolean loadAdditionalParams() {
    time = Integer.parseInt(additionalParameters.get("time"));
    ...
 }

 protected synchronized void jobWork() {
    // 1. EXECUTION TASK = to wait {time} seconds:
    int count = 0;
     while(!thread.isInterrupted() && count++ < time)
        Thread.sleep(1000);
    // If the task has been canceled/interrupted:
    if (thread.isInterrupted())
       throw new InterruptedException();

    // 2. WRITE THE RESULT FILE:
    ...
 }

 protected void clearResources() {
    // 1. Stop the job (if running):
    super.clearResources();
    // 2. Delete the result file (if any):
    ...
 }
```

# ADQL Library

- Version 1.0 **beta**
- ADQL 2.0 only
- SQL translation possible only for PostgreSQL+PgSphere for the moment
- Not yet managed:
  - STC-S
  - REGION function
  - Coordinate system conversions

# Example

Check with DB
*(optional)*

**Parse ADQL**

Modify ADQL query
*(optional)*

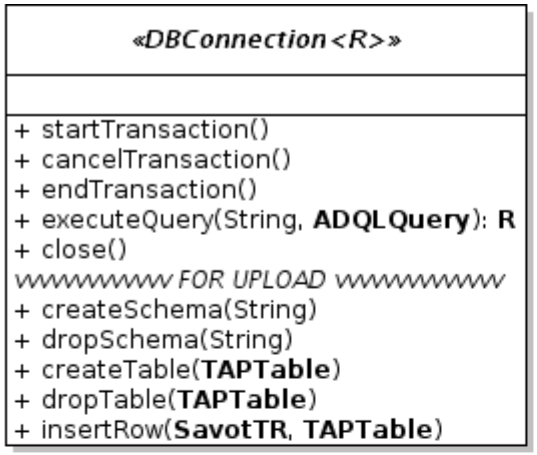
Translate into SQL
*(optional)*

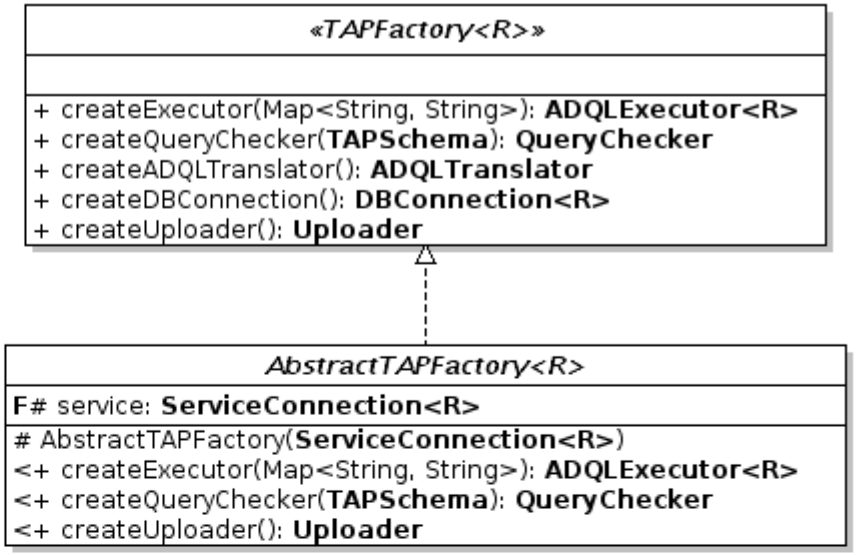
```
Collection<DBTable> tables = ...;
QueryChecker checker = new DBChecker(tables);

ADQLParser parser = new ADQLParser(checker);
try{
    // Parse the ADQL query:
    ADQLQuery query = parser.parseQuery(queryStr);

    // Manipulate the ADQL tree (change the column name case):
    ISearchHandler handler = new SearchColumnHandler();
    SearchResult results = handler.search(query);
    for(ADQLObject match : results){
        ADQLColumn column = (ADQLColumn)match;
        column.setColumnName(column.getColumnName().toUpperCase());
    }

    // Translate into SQL:
    ADQLTranslator translator = new PgSphereTranslator();
    String sql = translator.translate(query);
}catch(ParseException pe){
    System.err.println("Errors between [l."+pe.getBeginLine()+" ;
    c."+pe.getEndLine()+"] "+ pe.getMessage());
}
```

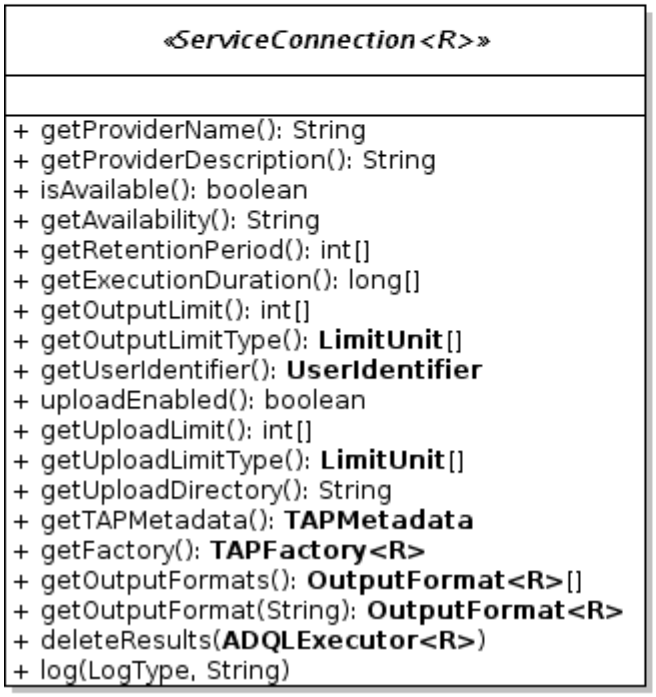
# TAP Library

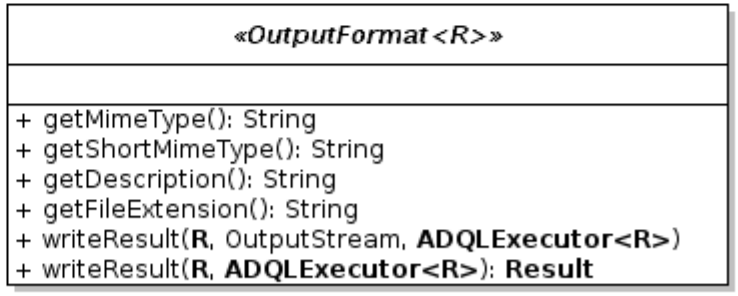
- Version 1.0 **beta**
- 4 short interfaces to implement:
  - DBConnection
  - ServiceConnection
  - TAPFactory
  - 1 OutputFormat per output format
- + 1 servlet to write as for UWS

# Interfaces to implement



# Example

```
public class MyTAP extends HttpServlet {

   private TAP tap = null;

   public void init(ServletConfig config) throws ServletException {
      try{
         ServiceConnection<ResultSet> service = new MyServiceConnection();
         tap = new TAP(service);
         tap.setHomePage("file:///home/myName/tapHome.html");       // or a URL
         tap.init(config);
      }catch(UWSException ue){
         throw new ServletException(ue);
      }
   }

   public void executeRequest(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
      tap.executeRequest(req, resp);
   }
}
```

# Work in progress...

- TAP
  - Update the library with the last TAPRegExt version
  - Improve the Upload API
    - May the Upload functionnality be an additionnal jar *(so, a kind of plugin)* ?
    - Or always integrated to the TAP library *(with Savot)* ?
- ADQL
  - More SQL translators *(for MySQL, SQLLite, ...)*
  - Impose the declaration of User Defined Functions *(which will be particularly usefull to full-fill TAPRegExt)*

# Some links

- CDS UWS Library
  - http://cdsportal.u-strasbg.fr/uwstuto
- CDS ADQL Library
  - http://cdsportal.u-strasbg.fr/adqltuto
- CDS TAP Library
  - http://cdsportal.u-strasbg.fr/taptuto

- Mail:
  - gregory.mantelet@astro.unistra.fr