- I got this lib to work with Oracle with only one minor modification in the sql generation in `JDBCoDataBase::createSql` (Oracle doesn't have the limit-clause so I switched to Oracle's `fetch next / offset <n> rows`)

```
//      sql.append(" limit ").append(top);
//      if (skip != null) {
//          sql.append(" offset ").append(skip);
        if (skip != null) {
            sql.append(" offset ").append(skip).append(" rows");
        }
        sql.append(" fetch next ").append(top).append(" rows only");
```
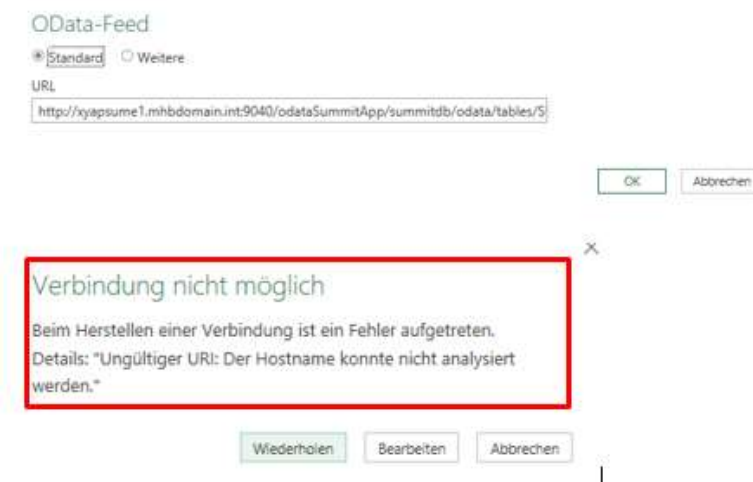
- While a request such as

http://xyapsume1.mhbdomain.int:9040/odataSummitApp/summitdb/odata/tables/SUMMIT/DMBOOK/RS?$top=5&$format=json

works just fine from a browser…

… it does not work when using the same request in PowerBI:

1. Reason seems to be that the odata.context in the returned message does not contain an absolute path (in this case "$metadata#RS").

   A similar issue was observed in olingo, see https://issues.apache.org/jira/browse/OLINGO-758 ("regardless of the spec, there are some consumers of OData that require the URL to be

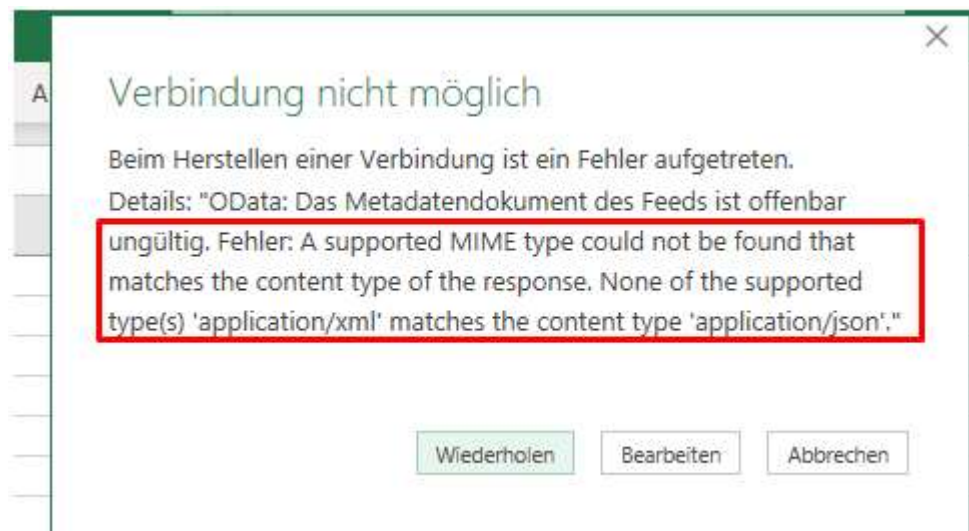absolute and not relative. Rhymes with Bike-ro-soft. 🙂")

2. In my next attempt, I kind of faked an absolute path for my example in the ODataResultSet constructor, so see where that gets me:

```java
public ODataResultSet() {
    // this.put(ODataUtils.ODATACONTEXT, ODataUtils.METADATA + "#" + ODataIdentifier.ENTITYSETNAME);
    this.put(ODataUtils.ODATACONTEXT, "http://xyapsume1.mhbdomain.int:9040/odataSummitApp/summitdb/odata/tables/SUMMIT/DMBOOK/" + ODataUtils.METADATA + "?$
    this.put(ODataUtils.VALUE, value);
}
```

Request via browser still works, different odata.context gets returned (absolute path):


```
JSON   Rohdaten   Kopfzeilen

Speichern  Kopieren  Einheitlich formatieren

"@odata.context":"http://xyapsume1.mhbdomain.int:9040/odataSummitApp/summitdb/odata/tables/SUMMIT/DMBOOK/$metadata?$format=json","value":
[{"AUDIT_ID":"0000009M","AUDIT_TABLE":"BOOK","AUDIT_VERSION":"1","AUDIT_CURRENT":"N","AUDIT_ENTITYSTATE":"VER","AUDIT_AUTHORIZED":"Y","AUDIT_PEN
","AUDIT_ISHISTORY":"Y","AUDIT_USER":"summit","AUDIT_ACTION":"IMLIVE","AUDIT_DATETIME":"2013-04-17T09:50:26Z","BOOK":"Z_HY_A_BE_22","DESCRIP":"H
","BONDAVGMATCHMETHOD":"NONE","FXREVALMETHOD":" ","FXREVALSWEEPTOBASE":"N","SECCONVMTHD":" ","BONDPREMDISCACCMETHOD":" ","DEFPOSITIONSTYLE":" ",
","ADJCAPGL":"N","PREPAYMENTS":" ","FLOATRATEPROJECTIONS":" ","SUBSEQUENTFEES":" ","ACCTPOSOPTIMIZATION":"P","SPLITFEERECOGNITION":"N","FEEAMORT
","USEEQYIELD":"N","DAILYCEY":" ","SEPFEEPOST":"N","USEDUR":"MAT","FUTURESAVGMATCHMETHOD":"NONE","ISSHADOW":"N","IFRSACCTTYPE":"
```

Using the original request again as an OData-Feed in PowerBI then led to a different error:

The one request when issued from PowerBI leads to two calls to my tomcat:

```
:12:08:28 +0100] "GET /odataSummitApp/summitdb/odata/tables/SUMMIT/DMBOOK/RS?$top=5&$format=json HTTP/1.1" 200 6853CRLF
:12:08:28 +0100] "GET /odataSummitApp/summitdb/odata/tables/SUMMIT/DMBOOK/$metadata?$format=json HTTP/1.1" 200 168092CRLF
```

So Microsoft uses the returned odata.context to issue another call (the one with $metadata), but seems to expect XML – data and not json?

⇨ Do you see any chance that issues 1 and 2 can get addressed so your lib could get used for odata data feed connections from PowerBI or Excel (I got the same errors in Excel)?