

Java class **AbstractDataSource**

- `public abstract String getDataSourceName();`
- `public abstract Collection<AbstractDataSet> getDataSets();`
- `public abstract JDBCMetaData getDataSourceMetaData();`
- `public abstract boolean authenticate() throws Exception;`
- `public abstract void disconnect();`
- `public abstract Map<String, Object> testConnection() throws Exception`

- `public ScheduleDefinition getScheduleDefinition();`
- `public boolean autoRun();`

- `protected final byte[ ] loadBlob(String key);`
- `protected final boolean saveBlob(String key, byte[ ] data);`
- `protected final boolean areBlobsAvailable();`
- `public final Object getAttribute(String key);`
- `public final Integer getSourceId();`

**public abstract String getDataSourceName();**

DataSourceNameString

**public abstract Collection<AbstractDataSet> getDataSets();**

collection

**public abstract JDBCMetaData getDataSourceMetaData();**

collection

**public abstract boolean authenticate() throws Exception;**

truefalsetrue

**public abstract void disconnect();**

disconnect()

public abstract Map<String, Object> testConnection() throws Exception

Yellowfin UI

ERROR

public ScheduleDefinition getScheduleDefinition();

ScheduleDefinitionScheduleDefinition

```
public ScheduleDefinition(FrequencyTypeCode frequencyTypeCode, String frequencyCode, Integer frequencyUnit);
```

frequencyTypeCode nfrequencyUnit

frequencyUnit	
MINUTES	n
DAILY	
WEEKLY	n
FORTNIGHTLY	frequencyCodeONETWOn
MONTHLY	n
ENDOFMONTH	
QUARTERLY	frequencyCodeONETWOTHREEEn
BIANNUAL	frequencyCodeONETWOTHREEFOURFIVESIXn
ANNUAL	frequencyCodeJANUARYFEBRUARYMARCHAPRILMAYJUNEJULYAUGUSTSEPTEMBEROCTOBERNOVEMBERDECEMBERn

```
public ScheduleDefinition getScheduleDefinition() {  
  
    return new ScheduleDefinition("WEEKLY", null, 1);  
  
}
```

```
public ScheduleDefinition getScheduleDefinition() {  
  
    return new ScheduleDefinition("MINUTES", null, 60);  
  
}
```

**public boolean autoRun();**

**autoRun**getScheduleDefinition()

**protected final byte[ ] loadBlob(String key);**

**loadBlob()**blob(byte[])BlobareBlobsAvailable()blob

**protected final boolean saveBlob(String key, byte[ ] data);**

**saveBlob()** blob(byte[])byte[]nullBlobareBlobsAvailable()blob

**protected final boolean areBlobsAvailable();**

**BlobareBlobsAvailable()**blobBlob

**public final Object getAttribute(String key);**

**getAttribute()**Yellowfin UIAPI

**public final Integer getSourceId();**

**getSourceId()**ID

**saveBlob()**loadBlob()

blob

## **Blob**

**Blob**Yellowfinblob

blob100,000**ALL\_TWEETS**blob

201601\_TWEETS

201602\_TWEETS

201603\_TWEETS

201604\_TWEETS

BlobBlob2016-02-05 2016-03-05201602\_TWEETS201603\_TWEETS23

blobblobblob

blobYellowfinblobDBMSYellowfin

Yellowfin

getColumnns()

getColumnns()getFilters()

Yellowfin UIThirdPartyException()

```
throw new ThirdPartyException("Unable to connect to the Twitter API at this time.");
```

Oh Noexecute()