Project 1-tbl SDK Design

Date	**05.06.2013**
Last modified by	**Roxan**

Introduction

What does the SDK provide to the client (programmer):

1. Get a list of all elements in one table (in project) and to filter that list (filtering coming later).
2. Get info on a specific element.
3. Add element to table (in project).
4. Using aggregation functions(MAX , MIN , COUNT , AVERAGE , SUM)
5. Modify element in table (in project).
6. Delete element from table (in project).
7. In case of no internet connection, all requests that failed will be sent as soon as the connection gets back.

- **Important: Please read the notes below.**

Change Management

Date	Author	Changes
06.05.2013	Fareed & Aziz	Started writing the template for the document
08.05.2013	Roxan	Started writing OneHelper class design , with the 4 methods
12.05.2013	Aziz & Rani	Modified the OneHelper class methods (where clause mechanism has been changed to be determined by manual user input) + added interpreter method
13.05.2013	Roxan , Fareed & Aziz, Rani	Modified methods , added a new factory class, added exception table
16.05.2013	Aziz	Added new classes(OneColumn ,OneColumnString,OneColumnInteger
26.05.2013	Fareed	Remove the logic of OnColumn from date 16.5.2013 Change the api to work with the server Add Action Listeners
28.05.2013	Roxan	Added notes , which are manifest changes that the programmer should add in order to use our sdk .
05.06.2013	Roxan	Added OneTableError , modified functions and variables definitions

SDK Design

OneHelper<R>

Public class

Class Overview

OneHelper class provides the programmer tools to access his tables in 1-tbl server, get information from them, change, and delete elements.

Each instance of OneHelper class have access to only one table in a specific project in 1-tbl server , and the variables (version , project token, userID ,table name) given at the constructor , are checked if valid at all calls to the server , and if one or more of them is invalid then an onError function is invoked with the corresponding OneTableError.

***Note : The class R given MUST have a default constructor.**

Summary

Public Constructors		
	OneHelper<R>(Context context,String token,String userID,String tableName)	

Public Methods	
void	insert(final R record, final OnInsertListener<R> listener)
void	delete(final Class<R> oneClass ,final String whereClause ,final String[] arguments , final OnDeleteListener<R> listener)
void	select(final Class<R> oneClass, final String whereClause, final String[] arguments, final OnSelectListener<R> listener)
Void	select(final Class<R> oneClass, final String whereClause, final String[] arguments,final OneAggregate aggObj, final OnSelectListener<R> listener)
void	update(final R obj, final String whereClause, final String[] arguments , final String[] columnsToUpdate , final OnUpdateListener<R> listener)

Public Constructors

*public **OneHelper**<R>(Context context, String token,String userID,String tableName)*

Constructs OneHelper object , given for public functions to deal with table(insert, delete, update, select)

***Note : The class R given MUST have a default constructor.**

Parameters:

context	The current context
token	The project token
userID	The User Id
tableName	The table name in the server

Public Methods

*Public void **insert**(final R record, final OnInsertListener<R> listener)*

Inserts given record fields to the table in 1-tbl server, where column names and values are the instance variables in the given record, the result which is a boolean indicating whether the progress was successful or not , is given to the listener's onInsert function , if an error occurred then it will be given to onError function according to the OneTableError object .

Parameters:

record The object to insert to the table in the server, values and columns names are according to the object's member variables.

OnInsertListener Interface that implements the functions onInsert(boolean success) and onError(OneTableError error) , the programmer should implement this interface and deal with the result as he wishes.

*Public void **delete**(final Class<R> oneClass ,String whereClause, String[] arguments, final OnDeleteListener<R> listener)*

Removes one object or more from the current object's table, according to the given where clause , the result which is number of rows affected , is given to the listener's onDelete function , if an error occurred then it will be given to onError function according to the OneTableError object .
****Note**: empty whereClause and arguments indicates a delete all query.

Parameters:

oneClass The class which is corresponding to the table (Columns names in the table are the class's instance variables names)

whereClause A condition that's according to it, an element or more is deleted from the table.(e.g. "ID > ? or Name = ? ")

arguments Array of strings, each member represents a value in the where clause (e.g.

whereClause= “ ClientID> ? and productId = ? or amount = ? “
arguments[0] = “31” , **arguments**[1] = “399%clientID” ,
arguments[2] = “clientID * 499 “)

OnDeleteListener Interface that implements the functions onDelete(int numOfDeleted) and onError(OneTableError error) ,the programmer should implement this interface and deal with the result as he wishes.

*Public void **select** select(final Class<R> oneClass, final String whereClause, final String[] arguments, final OnSelectListener<R> listener)*

Select one or more rows from the table using the given where clause and arguments, the result which is the values returned put in a list of the given object, is given to the listener's onSelect function , if an error occurred then it will be given to onError function according to the OneTableError object.
*__Note__: empty whereClause and arguments indicates a select all query.
***Note : The class R given MUST have a default constructor.**

Parameters:

oneClass The class which is corresponding to the table, the values returned will be put in objects of the given class.

whereClause A condition that's according to it, an element or more is selected from the table. (e.g. “ID >? Or Name =? “)

arguments Array of strings, each member represents a value in the where clause (e.g.
whereClause= “ ClientID > ? And productId =? Or amount =? “
arguments[0] = “31” , **arguments**[1] = “399%clientID” ,
arguments[2] = “clientID * 499 “)

OnSelectListener Interface that implements the functions onSelect(List<R> results) and onError(OneTableError error)the programmer should implement this interface and deal with the result as he wishes.

*ublic void **select** select(final Class<R> oneClass, final String whereClause, final String[] arguments ,final OneAggregate aggObj, final OnSelectListener<R> listener)*

THIS FUNCTION PROVIDES SELECTION WITH AGGREGATE FUNCTION.
Select one or more rows from the table using the given where clause and arguments, the result which is the values returned put in a list of the given object, is given to the listener's onSelect function , if an error occurred then it will be given to onError function according to the OneTableError object.
All columns that are not in an aggregation function will be grouped by in the sql query in server database , so if you wish not to group by all columns , use a new OneHelper with an object corresponding to columns you wish to group by.
*__Note__: empty whereClause and arguments indicates a select all query.
***Note : The class R given MUST have a default constructor.**

Parameters:

oneClass The class which is corresponding to the table, the values returned will be put in objects of the given class.

whereClause A condition that's according to it, an element or more is selected from the table. (e.g. “ID >? Or Name =? “)

arguments Array of strings, each member represents a value in the where clause (e.g.
whereClause= “ ClientID > ? And productId =? Or amount =? “
arguments[0] = “31” , **arguments**[1] = “399%clientID” ,
arguments[2] = “clientID * 499 “)

aggObj object containing the aggregate function , and the column to be applied on.

OnSelectListener Interface that implements the functions onSelect(List<R> results) and onError(OneTableError error)the programmer should implement this interface and deal with the result as he wishes.

Public void ***update*** *(final R obj, final String whereClause, final String[] arguments , final String[] columnsToUpdate , final OnUpdateListener<R> listener)*

Updates one or more rows at the table, number of rows updated are according to the where clause. The object's variables represent the column names to update, and the values represents the value to set at that column,and if you wish not to update all columns in the object you can set the Array columnsToUpdate , if you want to update all columns you can keep columnsToUpdate null.
the result which is number of rows affected , is given to the listener's onUpdate function , if an error occurred then it will be given to onError function according to the OneTableError object .
***Note**: empty whereClause and arguments indicates an update all query.

Parameters:

obj Instances of an object that it's variable's names are the columns names at the corresponding table and variable's values are to set at each column.

whereClause A condition that's according to it, an element or more are updated in the table. (e.g. “ID > ? Or Name = ? “)

arguments Array of strings, each member represents a value in the where clause (e.g.
whereClause= “ ClientID > ? And productId =? Or amount =? “
arguments[0] = “31” , **arguments**[1] = “399%clientID” ,
arguments[2] = “clientID * 499 “)

columnsToUpdate	Columns names to update, if null all columns will be updated
OnUpdateListener	Interface that implements the functions onUpdate(int rowUpdated) and onError(OneTableError error) , the programmer should implement this interface and deal with the result as he wishes.

OnDeleteListener<R>

Public interface

Class Overview

Interface definition for a callback to be invoked when a row/s deleted.

Summary

Public Methods	
void	onDelete(int numOfDeleted) Called when a row/s has been deleted, With the number of rows deleted.
void	onError(OneTableError error)OnDeleteListener<R> listener) Called when delete has failed.

OnInsertListener<R>

Public interface

Class Overview

Interface definition for a callback to be invoked when a row/s inserted.

Summary

Public Methods	
void	onInsert(boolean success) Called when a row/s has been inserted. With boolean indicating whether the insert was successful.
void	onError(OneTableError error)OnDeleteListener<R> listener) Called when an insert has failed.

OnSelectListener<R>

Public interface

Class Overview

Interface definition for a callback to be invoked when a row/s selected.

Summary

Public Methods	
void	onSelect(List<R> results) Called when a row/s has been selected. With a list if selected rows.
void	onError(OneTableError error)OnDeleteListener<R> listener) Called when a select has failed.

OnUpdateListener<R>

public interface

Class Overview

Interface definition for a callback to be invoked when a row/s updated.

Summary

Public Methods	
void	onUpdate(int rowsUpdated) Called when a row/s has been updated. With the number of rows updated.
void	onError(OneTableError error)OnDeleteListener<R> listener) Called when an update has failed.

OneTableError

Public class

Class Overview

This class indicates the errors that can occur during sending, processing and receiving data from the server, it includes all types of errors in an enum, and this error is passed to onError call back function to be implemented by the programmer .

Summary

Nested Classes

Enum	ErrorType	Represents the error types that can happen. { *InvalidCredentials* , *ColumnNotFound* , *ParameterParseException*, *IllegalCondition* , *IllegalArgument*, *IllegalAccess*, *DatabaseError* , *ServerError* , *NoSuchVersion* }

Instance Variables		
ErrorType	_type	Represents the error type
String	_message	Represents the error message

Public Constructors
OneTableError(ErrorType type , String message) Constructs an error.

Public Methods	
ErrorType	getType()
String	getMessage()

OneAggregate

Public class

Class Overview

This class is the aggregation representation which is a pair of aggregation function and a column to apply the aggregation on.

Summary

Nested Classes		
Enum	AggType	Represents the aggregation function to apply { *COUNT* , *SUM* , *MAX* , *MIN* ,*AVG* }

Instance Variables		
AGGTYPE	aggFunction	Represents the aggregation function to aply.
String	aggColumn	Represents the column to apply the aggregation on.

Public Constructors

OneAggregate(String aggColumn , AGGTYPE aggFunction)
Constructs an aggregation function on the column given.

Public Methods	
AggType	getAggFunction()
String	getAggColumn()

Notes

- In order to connect to the server and watch network connectivity change you **MUST** add the following permission to your application Manifest :

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission  android:name="android.permission.INTERNET"/>
```

- And you **MUST** add the following receiver and intent filter to your application Manifest , the following receiver watches internet connection in order to send and receive from the server , and watch network change:

```
<receiver
android:name="com.one_table.httpclient.ConnectionChangeReceiver" >

          <intent-filter>
                <action
                android:name="android.net.conn.CONNECTIVITY_CHANGE" />
                <action
                android:name="android.net.wifi.WIFI_STATE_CHANGED" />
          </intent-filter>
</receiver>
```

- The class R given at the object OneHelper **MUST** has a default constructor.

If you have further questions, we'll be happy to help, you may contact us on:

Roxan Barakat : roxan.onetablesdk@gmail.com

Aziz Kaddan : aziz.onetablesdk@gmail.com

Fareed Abu Saleh : fareed.onetablesdk@gmail.com