Post Refactor

package com.StockTake;

(+) import java.io.IOException;

public class GraphActivity extends Activity {

StockManager myStockmanager;

private Spinner spinner\_stocks, spinner\_time;

private Button btnGenerateGraph;

private XYPlot plot;

TableLayout table;

TableRow errorRow;

TextView error1;

TableRow.LayoutParams params;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

// Get the StockManager

myStockmanager = ((StockManager)getApplicationContext());

setContentView(R.layout.graph);

update();

}

private void update() {

if (checkInternetConnection()) {

try {

populateSpinners();

} catch(Exception e) {}

} else {}

}

private void populateSpinners() throws IOException, JSONException {

spinner\_stocks = (Spinner) findViewById(R.id.stock\_spinner);

final String[] stockArray = {“S&M”,”Experian”,”M&S”,”HSBC”,”BP”};

ArrayAdapter<String> dataAdapter = new ArrayAdapter<String>

(this, android.R.layout.simple\_spinner\_item,stockArray);

dataAdapter.setDropDownViewResource

(android.R.layout.simple\_spinner\_dropdown\_item);

spinner\_stocks.setAdapter(dataAdapter);

spinner\_time = (Spinner) findViewById(R.id.time\_spinner);

final String[] timeArray = {“Weekly”,”Monthly”,”Yearly”};

ArrayAdapter<String> dataAdapter1 = new ArrayAdapter<String>

(this, android.R.layout.simple\_spinner\_item,timeArray);

dataAdapter.setDropDownViewResource

(android.R.layout.simple\_spinner\_dropdown\_item);

spinner\_time.setAdapter(dataAdapter1);

// Button click Listener

addListenerGenerateGraphBtn();

}

private void addListenerGenerateGraphBtn() {

spinner\_stocks = (Spinner) findViewById(R.id.stock\_spinner);

spinner\_time = (Spinner) findViewById(R.id.time\_spinner);

btnGenerateGraph = (Button) findViewById(R.id.btnSubmit);

btnGenerateGraph.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

String stockname = String.valueOf(spinner\_stocks.getSelectedItem());

String time = String.valueOf(spinner\_time.getSelectedItem());

String stockAb = “”;

if (stockname.equals(“S&M”)) {

stockAb = “SN”;

} else if (stockname.equals(“Experian”)) {

stockAb = “EXPN”;

} else if (stockname.equals(“M&S”)) {

stockAb = “MKS”;

} else if (stockname.equals(“HSBC”)) {

stockAb = “HSBA”;

} else if (stockname.equals(“BP”)) {

stockAb = “BP”;

}

getGraphData(stockAb, time, stockname);

}

});

}

private boolean checkInternetConnection() {

ConnectivityManager conMgr = (ConnectivityManager) getSystemService(Context.CONNECTIVITY\_SERVICE);

if (conMgr.getActiveNetworkInfo() != null && conMgr.getActiveNetworkInfo().isAvailable() && conMgr.getActiveNetworkInfo().isConnected()) {

return true;

} else {

return false;

}

}

private void getGraphData(String stockAb, String time, String stockName)

{

LinkedList<Float> HistoricList = new LinkedList<Float>();

FeedParser HistoricData = new FeedParser();

HistoricList = HistoricData.getHistoricFeed(stockAb, time);

int weekBoundMax = 5;

int monthBoundMax = 20;

int yearBoundMax = 365;

Number[] array = HistoricList.toArray(new Number[HistoricList.size()]);

// initialize our XYPlot reference:

plot = (XYPlot) findViewById(R.id.mySimpleXYPlot);

plot.clear();

formatGraph(time,stockName,array);

// Turn the above arrays into XYSeries’:

XYSeries series1 = new SimpleXYSeries(Arrays.asList(array), SimpleXYSeries.ArrayFormat.Y\_VALS\_ONLY, “Series1”);

LineAndPointFormatter series1Format = new LineAndPointFormatter(

Color.rgb(0, 200, 0), Color.rgb(0, 100, 0), null, null);

// add a new series’ to the xyplot:

plot.addSeries(series1, series1Format);

//Domain = X-axis || Range = Y-axis

plot.setTicksPerRangeLabel(1);

plot.setTicksPerDomainLabel(1);

if(time.equals(“Weekly”)) {

plot.setDomainBoundaries(1, weekBoundMax, BoundaryMode.FIXED);

}

else if(time.equals(“Monthly”)) {

plot.setDomainBoundaries(1, monthBoundMax, BoundaryMode.FIXED);

}

else if(time.equals(“Yearly”)) {

plot.setDomainBoundaries(1, yearBoundMax, BoundaryMode.FIXED);

}

plot.getGraphWidget().setDomainLabelOrientation(-45);

plot.redraw();

}

private void formatGraph(String time, String stockName, Number[] array) {

//Set Graph Title according to selection

String graph\_title = “” + time + “ Graph for Stock: “ + stockName;

plot.setTitle(graph\_title);

plot.setDomainLabel(“Time Period: “ + time); // X-axis label

plot.setRangeLabel(“Share Value”); // Y-axis label

plot.getLegendWidget().setVisible(false);

plot.setDomainValueFormat(new DecimalFormat(“0”));

plot.setRangeValueFormat(new DecimalFormat(“0”));

}

}

Pre Refactor

package com.StockTake;

(+) import java.io.IOException;

public class GraphActivity extends Activity {

StockManager myStockmanager;

private Spinner spinner1;

private Button btnSubmit;

private Spinner spinner2;

Bundle stateForGraph;

private XYPlot plot;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

stateForGraph = savedInstanceState;

// Get the StockManager

myStockmanager = ((StockManager)getApplicationContext());

setContentView(R.layout.graph);

update();

}

private void update(){

if (checkInternetConnection()) {

try {

test();

} catch(Exception e) {

/\* Parse Error \*/

//error1.setText(Html.fromHtml(" <big>Oops!</big><br/><br/> Something went wrong when we tried to retrieve your share portfolio.<br/><br/> Please try again later."));

//errorRow.addView(error1, params);

//table.addView(errorRow);

}

} else {

/\* No Internet Connection \*/

//error1.setText(Html.fromHtml(" <big>Oops!</big><br/><br/> It seems there is a problem with your internet connection."));

//errorRow.addView(error1, params);

//table.addView(errorRow);

}

}

private void test() throws IOException, JSONException

{

//String stockName = "SN";

//getData(stockName);

spinner1 = (Spinner) findViewById(R.id.stock\_spinner);

List<String> list = new ArrayList<String>();

list.add("S&M");

list.add("Experian");

list.add("M&S");

list.add("HSBC");

list.add("BP");

ArrayAdapter<String> dataAdapter = new ArrayAdapter<String>

(this, android.R.layout.simple\_spinner\_item,list);

dataAdapter.setDropDownViewResource

(android.R.layout.simple\_spinner\_dropdown\_item);

spinner1.setAdapter(dataAdapter);

spinner2 = (Spinner) findViewById(R.id.time\_spinner);

List<String> list2 = new ArrayList<String>();

list2.add("Weekly");

list2.add("Monthly");

list2.add("Yearly");

ArrayAdapter<String> dataAdapter1 = new ArrayAdapter<String>

(this, android.R.layout.simple\_spinner\_item,list2);

dataAdapter.setDropDownViewResource

(android.R.layout.simple\_spinner\_dropdown\_item);

spinner2.setAdapter(dataAdapter1);

// Button click Listener

addListenerOnButton();

}

//get the selected dropdown list value

public void addListenerOnButton() {

spinner1 = (Spinner) findViewById(R.id.stock\_spinner);

spinner2 = (Spinner) findViewById(R.id.time\_spinner);

btnSubmit = (Button) findViewById(R.id.btnSubmit);

btnSubmit.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

String stockname;

String time;

String stockAb ="";

/\*Toast.makeText(GraphActivity.this,

"On Button Click : " +

"\n" + String.valueOf(spinner1.getSelectedItem()) + "\n" + String.valueOf(spinner2.getSelectedItem()),

Toast.LENGTH\_LONG).show();\*/

stockname = String.valueOf(spinner1.getSelectedItem());

time = String.valueOf(spinner2.getSelectedItem());

if(stockname.equals("S&M"))

{

stockname = "SN";

stockAb = "S&M";

}

else if (stockname.equals("Experian"))

{

stockname = "EXPN";

stockAb = "Experian";

}

else if (stockname.equals("M&S"))

{

stockname = "MKS";

stockAb = "M&S";

}

else if (stockname.equals("HSBC"))

{

stockname = "HSBA";

stockAb = "HSBC";

}

else if (stockname.equals("BP"))

{

stockname = "BP";

stockAb = "BP";

}

getData(stockname, time, stockAb);

}

});

}

private boolean checkInternetConnection() {

ConnectivityManager conMgr = (ConnectivityManager) getSystemService(Context.CONNECTIVITY\_SERVICE);

if (conMgr.getActiveNetworkInfo() != null && conMgr.getActiveNetworkInfo().isAvailable() && conMgr.getActiveNetworkInfo().isConnected()) {

return true;

} else {

return false;

}

}

public void getData(String stockname, String time, String stockAb)

{

LinkedList<Float> HistoricList = new LinkedList<Float>();

FeedParser HistoricData = new FeedParser();

HistoricList = HistoricData.getHistoricFeed(stockname, time);

int weekBoundMax = 5;

int monthBoundMax = 20;

int yearBoundMax = 365;

String[] week\_vals = {"Mon", "Tues", "Wed", "Thur", "Fri"};

String[] month\_vals = {""};

String[] year\_vals = {"Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sept", "Oct", "Nov", "Dec"};

for(int i = 0; i < HistoricList.size(); i++) {

Log.v("booyah", "4 : " + HistoricList.get(i));

}

Number[] array = HistoricList.toArray(new Number[HistoricList.size()]);

// initialize our XYPlot reference:

plot = (XYPlot) findViewById(R.id.mySimpleXYPlot);

plot.clear();

//Set Graph Title according to selection

String graph\_title = "" + time + " Graph for Stock: " + stockAb;

plot.setTitle(graph\_title);

plot.setDomainLabel("Time Period: " + time); //X-axis label

plot.setRangeLabel("Share Value"); //Y-axis label

plot.getLegendWidget().setVisible(false);

plot.setDomainValueFormat(new DecimalFormat("0"));

plot.setRangeValueFormat(new DecimalFormat("0"));

// Turn the above arrays into XYSeries':

XYSeries series1 = new SimpleXYSeries(

Arrays.asList(array), // SimpleXYSeries takes a List so turn our array into a List

SimpleXYSeries.ArrayFormat.Y\_VALS\_ONLY, // Y\_VALS\_ONLY means use the element index as the x value

"Series1"); // Set the display title of the series

LineAndPointFormatter series1Format = new LineAndPointFormatter(

Color.rgb(0, 200, 0),

Color.rgb(0, 100, 0),

null,

null);

// add a new series' to the xyplot:

plot.addSeries(series1, series1Format);

//Domain = X-axis || Range = Y-axis

plot.setTicksPerRangeLabel(1);

plot.setTicksPerDomainLabel(1);

if(time.equals("Weekly"))

{

plot.setDomainBoundaries(1, weekBoundMax, BoundaryMode.FIXED);

//set domain labels as string [x-axis]

//plot.getGraphWidget().setDomainValueFormat(new GraphXLabelFormat());

}

else if(time.equals("Monthly"))

{

plot.setDomainBoundaries(1, monthBoundMax, BoundaryMode.FIXED);

//set domain labels as string [x-axis]

//plot.getGraphWidget().setDomainValueFormat(new GraphXLabelFormat());

}

else if(time.equals("Yearly"))

{

plot.setDomainBoundaries(1, yearBoundMax, BoundaryMode.FIXED);

//set domain labels as string [x-axis]

//plot.getGraphWidget().setDomainValueFormat(new GraphXLabelFormat());

}

plot.getGraphWidget().setDomainLabelOrientation(-45);

plot.redraw()

}

}