# 基本部件构建

## 网址循环

网址：

http://quotes.money.163.com/service/chddata.html?code=0600297&start=20001116&end=20161226&fields=TCLOSE;HIGH;LOW;TOPEN;LCLOSE;CHG;PCHG;TURNOVER;VOTURNOVER;VATURNOVER;TCAP;MCAP

private void button1\_Click(object sender, EventArgs e)

{

string stockID = null;

for (int i = 0; i < 10; i++)

{

stockID = i.ToString("D6");

MessageBox.Show(stockID);

string starttime ="20000101"; //开始时间

string endtime = DateTime.Now.ToString("yyyymmdd");

string wedad = string.Format(

@"http://quotes.money.163.com/service/chddata.html?code=0{0}&start={1}&end={2}&fields=TCLOSE;HIGH;LOW;TOPEN;LCLOSE;CHG;PCHG;TURNOVER;VOTURNOVER;VATURNOVER;TCAP;MCAP",

stockID,starttime,endtime);

textBox1.Text+= wedad+"\n";

}

}

## 循环下载

#region 函数

private DataTable DownloadCSVToTable(string x\_URL)

{

DataTable dt = new DataTable();

string Line\_str = ""; //读取行内容

string[] Line\_ary = null; //字符串用于Split行内容

bool IsFirst = true; //布尔值是否是读取的第一行

HttpWebRequest request = (HttpWebRequest)WebRequest.Create(x\_URL);

HttpWebResponse response = (HttpWebResponse)request.GetResponse();

Stream responseStream = response.GetResponseStream();

using (StreamReader respStreamReader = new StreamReader(responseStream, Encoding.Default))

{

while ((Line\_str = respStreamReader.ReadLine()) != null)

{

Line\_ary = Line\_str.Split(',');

if (IsFirst == true) //判断是否第一行

{

for (int j = 0; j < Line\_ary.Length; j++) //添加列和列名

{

DataColumn dc = new DataColumn(Line\_ary[j]);

dt.Columns.Add(dc);

}

IsFirst = false;

}

else //如果是第其他行

{

DataRow dr = dt.NewRow(); //添加行数据

for (int j = 0; j < Line\_ary.Length; j++)

{

dr[j] = Line\_ary[j];

}

dt.Rows.Add(dr);

}

}

}

responseStream.Close();

responseStream.Dispose();

return dt;

}

#endregion

private void button1\_Click(object sender, EventArgs e)

{

string stockID = null;

for (int i = 0; i < 10; i++)

{

dataGridView1.DataSource = null;

stockID = i.ToString("D6");

string starttime ="20000101"; //开始时间

string endtime = DateTime.Now.ToString("yyyymmdd");

string URL = string.Format(

@"http://quotes.money.163.com/service/chddata.html?code=0{0}&start={1}&end={2}&fields=TCLOSE;HIGH;LOW;TOPEN;LCLOSE;CHG;PCHG;TURNOVER;VOTURNOVER;VATURNOVER;TCAP;MCAP",

stockID,starttime,endtime);

textBox1.Text+= URL+"\n";

dataGridView1.DataSource = DownloadCSVToTable(URL);

MessageBox.Show(stockID+ dataGridView1.Rows.Count.ToString());

}

}

## 批量创建表

SQL=

create table DB\_Stock.dbo.cn000009

(

No Int Identity(1,1) Primary Key,

日期 date not null,

股票代码 char(6) not null,

名称 nchar(6) not null,

收盘价 decimal(10,4) not null,

最高价 decimal(10,4) not null,

最低价 decimal(10,4) not null,

开盘价 decimal(10,4) not null,

前收盘 decimal(10,4) not null,

涨跌额 decimal(9,4) not null,

涨跌幅 decimal(5,4) not null,

换手率 char(10),

成交量 bigint not null,

成交金额 float not null,

总市值 char (15),

流通市值 char(15)

)

private void button2\_Click(object sender, EventArgs e)

{

try

{

for (int i = 0; i < 9; i++)

{

string TableName = null;

TableName = "CN" + i;

if (IsTableExit("SIM-PC", "DB\_Stock", TableName) != true)

{

string Conn\_str = ConnectionText("SIM-PC", "DB\_Stock");

string SQL\_str = string.Format(

@"create table {0}

(

No Int Identity(1, 1) Primary Key,

日期 date not null,

股票代码 char(6) not null,

名称 nchar(6) not null,

收盘价 decimal(10, 4) not null,

最高价 decimal(10, 4) not null,

最低价 decimal(10, 4) not null,

开盘价 decimal(10, 4) not null,

前收盘 decimal(10, 4) not null,

涨跌额 decimal(9, 4) not null,

涨跌幅 decimal(5, 4) not null,

换手率 decimal(5, 4),

成交量 bigint not null,

成交金额 decimal(16, 4) not null,

总市值 decimal(16, 4),

流通市值 decimal(16, 4)

)",

TableName);

SqlCommand(Conn\_str, SQL\_str);

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

## 批量创建表（判断是否有现有表）

private bool IsTableExit(string x\_ServerName, string y\_DataBase,string z\_TableName) //判断数据表是否存在

{

bool line=false;

SqlConnection Conn = new SqlConnection(ConnectionText(x\_ServerName, y\_DataBase));

Conn.Open();

try

{

SqlCommand Cmd = new SqlCommand();

Cmd.CommandText = string.Format(@"select COUNT(\*) from sysobjects where ID = OBJECT\_ID(N'{0}')", z\_TableName);

Cmd.Connection = Conn;

using (SqlDataReader dr = Cmd.ExecuteReader())

{

dr.Read();

line = Convert.ToBoolean(dr[0]);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

Conn.Close();

Conn.Dispose();

}

return line;

}

private void button2\_Click(object sender, EventArgs e)

{

//textBox1.Text=IsTableExit("SIM-PC","DB\_Stock",textBox1.Text).ToString();

try

{

for (int i = 0; i < 9; i++)

{

string TableName = null;

TableName = "CN" + i;

if (IsTableExit("SIM-PC", "DB\_Stock", TableName) != true)

{

string Conn\_str = ConnectionText("SIM-PC", "DB\_Stock");

string SQL\_str = string.Format(

@"create table {0}

(

No Int Identity(1, 1) Primary Key,

日期 date not null,

股票代码 char(6) not null,

名称 nchar(6) not null,

收盘价 decimal(10, 4) not null,

最高价 decimal(10, 4) not null,

最低价 decimal(10, 4) not null,

开盘价 decimal(10, 4) not null,

前收盘 decimal(10, 4) not null,

涨跌额 decimal(9, 4) not null,

涨跌幅 decimal(5, 4) not null,

换手率 decimal(5, 4),

成交量 bigint not null,

成交金额 decimal(16, 4) not null,

总市值 decimal(16, 4),

流通市值 decimal(16, 4)

)",

TableName);

SqlCommand(Conn\_str, SQL\_str);

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

## 批量写入数据库（dataGridview）

private void AddDataGridViewToSql(DataGridView DGV, string x\_ServerName, string y\_DataBase, string z\_TableName) //将DGV写入SQL

{

for (int j = 0; j < DGV.Rows.Count - 1; j++)

{

string Conn\_str = ConnectionText(x\_ServerName, y\_DataBase);

int Col = DGV.Columns.Count;

string[] field = new string[Col];

string[] value = new string[Col];

for (int i = 0; i < Col; i++)

{

//获取字段名

field[i] = DGV.Columns[i].Name.ToString();

//获取数据值

if (DGV.Rows[0].Cells[i].Value.ToString() != "")

{

value[i] = "'" + DGV.Rows[j].Cells[i].Value.ToString() + "'";

}

else

{

value[i] = "null";

}

}

string CommandText = string.Format(

@"Insert Into {0} ({1}) Values ({2})",

z\_TableName,

string.Join(",", field),

string.Join(",", value));

SqlConnection Conn = new SqlConnection(Conn\_str);

Conn.Open();

try

{

SqlCommand Cmd = new SqlCommand(CommandText, Conn);

Cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

Conn.Close();

Conn.Dispose();

}

}

}

private void button3\_Click(object sender, EventArgs e)

{

for (int i = 0; i < dataGridView1.Rows.Count-1; i++)

{

string cell = dataGridView1.Rows[i].Cells[1].Value.ToString();

cell = cell.Substring(1);

dataGridView1.Rows[i].Cells[1].Value = cell;

}

AddDataGridViewToSql(dataGridView1, "SIM-PC", "DB\_Stock", "cc");

}

## 逐行写入SQL

private void button4\_Click(object sender, EventArgs e) //打开连接逐行写入数据

{

string TableName = "ff";

string Conn\_str = ConnectionText("SIM-PC","DB\_Stock");

string URL = @"http://quotes.money.163.com/service/chddata.html?code=0000009&start=20161201&end=20161227&fields=TCLOSE;HIGH;LOW;TOPEN;LCLOSE;CHG;PCHG;TURNOVER;VOTURNOVER;VATURNOVER;TCAP;MCAP";

string Line\_str = ""; //读取行内容

string[] Line\_ary = null; //字符串用于Split行内容

string field = null; //写入的字段

string value = null; //写入的数据

bool IsFirst = true; //布尔值是否是读取的第一行

int Col = 0; //字段列数

int Row = 0; //数据行数

string Con\_str = ConnectionText("SIM-PC","DB\_Stock");

string SQL = null;

HttpWebRequest request = (HttpWebRequest)WebRequest.Create(URL);

HttpWebResponse response = (HttpWebResponse)request.GetResponse();

Stream responseStream = response.GetResponseStream();

using (StreamReader respStreamReader = new StreamReader(responseStream, Encoding.Default))

{

while ((Line\_str = respStreamReader.ReadLine()) != null)

{

Row += 1;

if (IsFirst == true) //判断是否第一行

{

field = Line\_str;

Line\_ary = Line\_str.Split(','); //生成数组

Col = Line\_ary.Length;

IsFirst = false;

}

else

{

//去除“'”符号

value = Line\_str;

Line\_ary = value.Split('\'');

value = string.Join("", Line\_ary);

Line\_ary = null;

Line\_ary = value.Split(',');

value = "'" + string.Join("','", Line\_ary) + "'";

SQL = "";

SQL = string.Format(

@"Insert Into {0} ({1}) Values ({2})",

TableName,

field,

value);

SqlCommand(Conn\_str, SQL); //写入数据

}

}

}

responseStream.Close();

responseStream.Dispose();

MessageBox.Show("已经写入[" + Row + "]行数据！");

}

## 使用datetimepicker比较选择时间

private void button5\_Click(object sender, EventArgs e)

{

string starttime = dateTimePicker1.Value.ToShortDateString();

DateTime dt1 = Convert.ToDateTime(starttime);

starttime = string.Join("", starttime.Split('/'));

string endtime= dateTimePicker2.Value.ToShortDateString();

DateTime dt2 = Convert.ToDateTime(endtime);

endtime = string.Join("", endtime.Split('/'));

if (DateTime.Compare(dt1, dt2) < 0)

{

MessageBox.Show("<");

}

else

{

MessageBox.Show(">");

}

}

## 批量插入表

1. 读取CSV到dt
2. 将dt插入表

private DataTable DownloadCSVToTable(string x\_URL)

{

DataTable dt = new DataTable();

string Line\_str = ""; //读取行内容

string[] Line\_ary = null; //字符串用于Split行内容

bool IsFirst = true; //布尔值是否是读取的第一行

HttpWebRequest request = (HttpWebRequest)WebRequest.Create(x\_URL);

HttpWebResponse response = (HttpWebResponse)request.GetResponse();

Stream responseStream = response.GetResponseStream();

using (StreamReader respStreamReader = new StreamReader(responseStream, Encoding.Default))

{

while ((Line\_str = respStreamReader.ReadLine()) != null)

{

Line\_str = string.Join("", Line\_str.Split('\''));

Line\_ary = Line\_str.Split(',');

if (IsFirst == true) //判断是否第一行

{

for (int j = 0; j < Line\_ary.Length; j++) //添加列和列名

{

DataColumn dc = new DataColumn(Line\_ary[j]);

dt.Columns.Add(dc);

}

IsFirst = false;

}

else //如果是第其他行

{

DataRow dr = dt.NewRow(); //添加行数据

for (int j = 0; j < Line\_ary.Length; j++)

{

dr[j] = Line\_ary[j];

}

dt.Rows.Add(dr);

}

}

}

responseStream.Close();

responseStream.Dispose();

return dt;

}

private static void DataTableToSQLServer(DataTable dt, string connectString,string Table) //将DataTable写入SQL

{

using (SqlConnection Conn = new SqlConnection(connectString))

{

Conn.Open();

using (SqlBulkCopy bulkCopy = new SqlBulkCopy(Conn))

{

try

{

bulkCopy.DestinationTableName = Table;//要插入的表的表名

bulkCopy.BatchSize = dt.Rows.Count;

bulkCopy.ColumnMappings.Add("日期", "日期");//映射字段名 DataTable列名 ,数据库 对应的列名

bulkCopy.ColumnMappings.Add("股票代码", "股票代码");

bulkCopy.ColumnMappings.Add("名称", "名称");

bulkCopy.ColumnMappings.Add("收盘价", "收盘价");

bulkCopy.ColumnMappings.Add("最高价", "最高价");

bulkCopy.ColumnMappings.Add("最低价", "最低价");

bulkCopy.ColumnMappings.Add("开盘价", "开盘价");

bulkCopy.ColumnMappings.Add("前收盘", "前收盘");

bulkCopy.ColumnMappings.Add("涨跌额", "涨跌额");

bulkCopy.ColumnMappings.Add("涨跌幅", "涨跌幅");

bulkCopy.ColumnMappings.Add("换手率", "换手率");

bulkCopy.ColumnMappings.Add("成交量", "成交量");

bulkCopy.ColumnMappings.Add("成交金额", "成交金额");

bulkCopy.ColumnMappings.Add("总市值", "总市值");

bulkCopy.ColumnMappings.Add("流通市值", "流通市值");

bulkCopy.WriteToServer(dt);

//System.Windows.Forms.MessageBox.Show("插入成功");

}

catch (Exception ex)

{

MessageBox.Show (ex.Message);

}

finally

{

}

}

}

}

private void button6\_Click(object sender, EventArgs e)

{

string URL = @"http://quotes.money.163.com/service/chddata.html?code=0000009&start=20161201&end=20161227&fields=TCLOSE;HIGH;LOW;TOPEN;LCLOSE;CHG;PCHG;TURNOVER;VOTURNOVER;VATURNOVER;TCAP;MCAP";

DataTable dt = new DataTable();

dt = DownloadCSVToTable(URL);

//dataGridView1.DataSource = dt.DefaultView;

string Con\_str = ConnectionText("SIM-PC", "DB\_Stock");

DataTableToSQLServer(dt, Con\_str,"ff");

}

**2、封装批量插入数据SQL Server数据的方法**

1 public static void DataTableToSQLServer(DataTable dt,string connectString)

2 {

3 string connectionString = connectString;

4

5 using (SqlConnection destinationConnection = new SqlConnection(connectionString))

6 {

7 destinationConnection.Open();

8

9 using (SqlBulkCopy bulkCopy = new SqlBulkCopy(destinationConnection))

10 {

11

12

13 try

14 {

15

16 bulkCopy.DestinationTableName = "checkinout";//要插入的表的表名

17 bulkCopy.BatchSize = dt.Rows.Count;

18 bulkCopy.ColumnMappings.Add("ID", "ID");//映射字段名 DataTable列名 ,数据库 对应的列名

19 bulkCopy.ColumnMappings.Add("TIME", "TIME");

20

21 bulkCopy.WriteToServer(dt);

22 System.Windows.Forms.MessageBox.Show("插入成功");

23 }

24 catch (Exception ex)

25 {

26 Console.WriteLine(ex.Message);

27 }

28 finally

29 {

30

31

32 }

33 }

34

35

36 }

37

38 }

# 优化一：

## 开始时间结束时间选择

选择时间：

private void dateTimePicker1\_CloseUp(object sender, EventArgs e)

{

startDate\_txb.Text = dateTimePicker1.Value.ToString("yyyyMMdd");

}

private void dateTimePicker2\_CloseUp(object sender, EventArgs e)

{

endDate\_txb.Text = dateTimePicker2.Value.ToString("yyyyMMdd");

}

判断时间：

private bool IsDateValid(string x\_date,string y\_date) //判断两个时间是否合规

{

bool Valid=false;

DateTime dt1 = DateTime.ParseExact(startDate\_txb.Text, "yyyyMMdd", System.Globalization.CultureInfo.CurrentCulture);

DateTime dt2 = DateTime.ParseExact(endDate\_txb.Text, "yyyyMMdd", System.Globalization.CultureInfo.CurrentCulture);

//DateTime dt1 = Convert.ToDateTime(x\_date);

//DateTime dt2 = Convert.ToDateTime(y\_date);

Valid=DateTime.Compare(dt1, dt2) > 0 ?true:false;

return Valid;

}

private void button7\_Click(object sender, EventArgs e)

{

if (startDate\_txb.Text == "" || endDate\_txb.Text == "") //判断日期是否输入

{

MessageBox.Show("时间不能为空");

}

else

{

if (IsDateValid(startDate\_txb.Text, endDate\_txb.Text))

{

MessageBox.Show("开始时间不能大于结束时间");

}

else

{

MessageBox.Show("AA");

}

}

}

## URL组合（时间）

private string FormURL(string code, string date\_start, string date\_end) //生成URL字符串

{

string URL=null;

date\_start = string.Join("", date\_start.Split('/'));

date\_end = string.Join("", date\_end.Split('/'));

URL = string.Format(@"http://quotes.money.163.com/service/chddata.html?code=0{0}&start={1}&end={2}&fields=TCLOSE;HIGH;LOW;TOPEN;LCLOSE;CHG;PCHG;TURNOVER;VOTURNOVER;VATURNOVER;TCAP;MCAP",

code, date\_start, date\_end);

return URL;

}

private void button7\_Click(object sender, EventArgs e)

{

int start = 0;

int end = 999;

string URL = null;

if (startDate\_txb.Text == "" || endDate\_txb.Text == "") //判断日期是否输入

{

MessageBox.Show("时间不能为空");

}

else

{

if (IsDateValid(startDate\_txb.Text, endDate\_txb.Text))

{

MessageBox.Show("开始时间不能大于结束时间");

}

else

{

MessageBox.Show("AA");

URL= FormURL(end.ToString("000000"),startDate\_txb.Text,endDate\_txb.Text);

textBox1.Text=URL;

}

}

## URL组合（代码Code循环）

private void button7\_Click(object sender, EventArgs e)

{

int start = 0;

int end = 3;

string URL = null;

if (startDate\_txb.Text == "" || endDate\_txb.Text == "") //判断日期是否输入

{

MessageBox.Show("时间不能为空");

}

else

{

if (IsDateValid(startDate\_txb.Text, endDate\_txb.Text))

{

MessageBox.Show("开始时间不能大于结束时间");

}

else

{

int codeID = start;

do

{

URL = FormURL(codeID.ToString("000000"), startDate\_txb.Text, endDate\_txb.Text);

//MessageBox.Show(codeID.ToString());

//MessageBox.Show(URL);

codeID += 1;

}

while (codeID<end+1);

}

}

## 下载CSV测试

private void button7\_Click(object sender, EventArgs e)

{

int start = 0;

int end = 10;

string URL = null;

if (startDate\_txb.Text == "" || endDate\_txb.Text == "") //判断日期是否输入

{

MessageBox.Show("时间不能为空");

}

else

{

if (IsDateValid(startDate\_txb.Text, endDate\_txb.Text))

{

MessageBox.Show("开始时间不能大于结束时间");

}

else

{

int codeID = start;

do

{

URL = FormURL(codeID.ToString("000000"), startDate\_txb.Text, endDate\_txb.Text);

dataGridView1.DataSource=DownloadCSVToTable(URL);

MessageBox.Show(codeID.ToString());

codeID += 1;

}

while (codeID<end+1);

}

}

}

# 优化二：

## 状态显示

this.toolStripStatusLabel1.Text = "正在处理：" + codeID.ToString("000000"); //状态栏显示

## 进度条显示进度

循环前设置：

this.toolStripProgressBar1.Maximum = end-start+1; //进度条初始化

this.toolStripProgressBar1.Value = start;

循环内步进：

toolStripProgressBar1.Value++; //进度条步进

## 判断数据建立数据表

private void button7\_Click(object sender, EventArgs e)

{

int start = 0;

int end = 5;

if (startDate\_txb.Text == "" || endDate\_txb.Text == "") //判断日期是否输入

{

MessageBox.Show("时间不能为空");

}

else

{

if (IsDateValid(startDate\_txb.Text, endDate\_txb.Text))

{

MessageBox.Show("开始时间不能大于结束时间");

}

else

{

int codeID = start;

do

{

this.toolStripStatusLabel1.Text = "正在处理：" + codeID.ToString("000000"); //状态栏显示

string URL = FormURL(codeID.ToString("000000"), startDate\_txb.Text, endDate\_txb.Text);

using (DataTable dt = DownloadCSVToTable(URL))

{

if (dt.Rows.Count != 0) //判断是否有数据

{

string TableName = null;

TableName = "CN" + codeID.ToString("000000");

string ConnectionTxt = ConnectionText("SIM-PC", "DB\_Stock");

if (IsTableExit(ConnectionTxt, TableName) != true) //如果判断表是否存在，如果不存在创建表

{

CreateTable(ConnectionTxt, TableName); //创建新表

DataTableToSQLServer(dt, ConnectionTxt, TableName); //添加数据

}

else //如果已经有表，载入数据

{

DataTableToSQLServer(dt, ConnectionTxt, TableName); //添加数据

}

}

}

codeID += 1; //步进

}

while (codeID<end+1);

}

}

}

## 批量删除数据表

private void DeleteTable(string ConnectionText, string Table) //删除数据表

{

string SQL\_str = string.Format(@"Drop Table {0}", Table);

SqlCommand(ConnectionText, SQL\_str);

}

private void button1\_Click(object sender, EventArgs e)

{

int start = 6;

int end = 10;

int codeID = start;

do

{

string Table = "CN" + codeID.ToString("000000");

MessageBox.Show(Table);

DeleteTable(ConnectionText("SIM-PC", "DB\_Stock"), Table);

codeID += 1;

}

while (codeID < end + 1);

}

## 状态显示（TXTbox）：

循环内显示：

textBox1.AppendText("处理完毕：" + codeID.ToString("000000") + " " + dt1.ToLongTimeString() + "\n");

完成时显示：

textBox1.AppendText("运行完毕，共处理" + (end - start).ToString() +"共用时："+ TimerDiff(dt1, dt2) + "\n");

## 创建数据表（设定更新）

string SQL\_str = string.Format(

@"create table {0}

(

No Int Identity(1, 1) Primary Key,

日期 date not null,

股票代码 char(6) not null,

名称 nchar(10) not null,

收盘价 decimal(10, 4) not null,

最高价 decimal(10, 4) not null,

最低价 decimal(10, 4) not null,

开盘价 decimal(10, 4) not null,

前收盘 char (10) not null,

涨跌额 char (10) not null,

涨跌幅 char (10) not null,

换手率 char(10),

成交量 char(20),

成交金额 char(20),

总市值 char (15),

流通市值 char(15)

)",

Table);

## 批量删除数据表（返回bool值）

# 测试

## 测试1：1年数据，5个代码

运行完毕，共处理5共用时：0分00秒

## 测试2：10年数据，5个代码

运行完毕，共处理5共用时：0分04秒

## 测试3：10年数据，10个代码

运行完毕，共处理10共用时：0分07秒

## 测试4：10年数据，100个代码

运行完毕，共处理99共用时：0分57秒

## 测试4：基年数据，50个代码

运行完毕，共处理50共用时：0分44秒

# 优化三

## 总览数据表设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | 代码 | 名称 | 更新时间 | 创建时间 |
|  |  |  |  |  |
|  |  |  |  |  |

create table DB\_Stock.dbo.total

(

No Int Identity(1, 1) Primary Key,

代码 char(10) not null,

名称 char(10) not null,

更新时间 date not null,

创建时间 date not null,

备注 text

)

# End