测试用例列表：

|  |  |
| --- | --- |
| 测试用例ID | 测试用例描述 |
| StockDSImpl\_Test001 | StockDSImpl中方法的测试 |
| BenchMarkDSImpl\_Test002 | BenchMarkDSImpl中方法的测试 |
| OptionalStockDSImpl\_Test003 | OptionalStockDSImpl中方法的测试 |
| StockBLImpl\_Test004 | StockBLImpl中方法的测试 |
| BenchMarkBLImpl\_Test005 | BenchMarkBLImpl中方法的测试 |
| OptionalStockBLmpl\_Test006 | OptionalStockBLmpl中方法的测试 |

相关的Junit源码文件：

1. StockDSImpl\_Test

**public** **class** StockDSImplTest {

StockDataService stockDataService;

@Before

**public** **void** setUp() **throws** Exception {

stockDataService = StockDSImpl.*getStockDSImpl*();

}

@After

**public** **void** tearDown() **throws** Exception {

stockDataService = **null**;

}

@Test

**public** **void** testGetAllStocks() {

List<String> **codeStrings** = stockDataService.getAllStocks();

**if** (codeStrings == **null**) {

*fail*("testGetAllStocks()--null");

} **else** **if** (codeStrings.size() < 10) {

*fail*("testGetAllStocks()--size");

}

}

@Test

**public** **void** testGetAllStocksInt() {

**for** (**int** **i** = 2010; i < 2017; i++) {

List<String> **codeStrings** = stockDataService.getAllStocks(i);

**if** (codeStrings == **null**) {

*fail*("testGetAllStocks()--null");

} **else** **if** (codeStrings.size() < 10) {

*fail*("testGetAllStocks()--size");

}

}

}

@Test

**public** **void** testGetAllStocksExchange() {

List<String> **codes** = stockDataService.getAllStocks(Exchange.***sh***);

**if** (codes == **null**) {

*fail*("testGetAllStocks()--null");

} **else** **if** (codes.size() < 10) {

*fail*("testGetAllStocks()--size");

}

List<String> **codeStrings** = stockDataService.getAllStocks(Exchange.***sz***);

**if** (codeStrings == **null**) {

*fail*("testGetAllStocks()--null");

} **else** **if** (codeStrings.size() < 10) {

*fail*("testGetAllStocks()--size");

}

}

@Test

**public** **void** testGetAllStocksIntExchange() {

List<String> **codes**;

**for** (**int** **i** = 2015; i < 2018; i++) {

**for** (**int** **j** = 0; j < 2; j++) {

**if** (j % 2 == 0) {

codes = stockDataService.getAllStocks(Exchange.***sh***);

} **else** {

codes = stockDataService.getAllStocks(Exchange.***sz***);

}

**if** (codes == **null**) {

*fail*("testGetAllStocks()--null");

} **else** **if** (codes.size() < 10) {

*fail*("testGetAllStocks()--size");

}

}

}

}

@Test

**public** **void** testGetStockMesString() {

String [] **codes** = {"sh600000","sh600216","sz002588","sz002310","sh00012","sh5000001"};

List<String> **errorCode** = **new** ArrayList<String>();

List<StockPO> **stockPOs** = **new** ArrayList<>();

**for**(**int** **i**=0; i<codes.length;i++){

StockPO **stockpo** = stockDataService.getStockMes(codes[i]);

**if**(stockpo==**null**){

errorCode.add(codes[i]);

}**else**{

stockPOs.add(stockpo);

}

}

**if**(!errorCode.get(0).equals(codes[4])||!errorCode.get(1).equals(codes[5])){

*fail*("testGetStockMesString()---errorCodes can't be properly handled");

}**else** **if**( !stockPOs.get(0).getCode().equals(codes[0])){

*fail*("testGetStockMesString()---stockpo code changed");

}

}

@Test

**public** **void** testGetStockMesStringMyDate() {

MyDate **date** =DateCalculator.*getAnotherDay*(-1);

String **code** = "sh600216";

String **nameString** = "浙江医药";

String **dateString** = date.DateToString();

StockPO **po** = stockDataService.getStockMes(code,date);

**if**(po==**null**){

*fail*("testGetStockMesStringMyDate() --NULL");

}**else** **if**(!po.getCode().equals(code)){

*fail*("testGetStockMesStringMyDate() --code");

}**else** **if**(!po.getName().equals(nameString)){

*fail*("testGetStockMesStringMyDate() --name");

}**else** **if**(!po.getDate().equals(dateString)){

*fail*("testGetStockMesStringMyDate() --date");

}

}

@Test

**public** **void** testGetStockMesStringMyDateMyDate() {

MyDate **endDate** =DateCalculator.*getToDay*();

MyDate **startDate** = DateCalculator.*getAnotherDay*(-30);

String **code** = "sh600216";

String **nameString** = "浙江医药";

String **dateString** = startDate.DateToString();

List<StockPO> **pos** = stockDataService.getStockMes(code,startDate,endDate);

**if**(pos==**null**){

*fail*("testGetStockMesStringMyDateMyDate() --NULL");

}**else** **if**(!pos.get(0).getCode().equals(code)){

*fail*("testGetStockMesStringMyDateMyDate()--code");

}**else** **if**(!pos.get(0).getName().equals(nameString)){

*fail*("testGetStockMesStringMyDateMyDate() --name");

}**else** **if**(!pos.get(0).getDate().equals(dateString)){

*fail*("testGetStockMesStringMyDateMyDate() --date");

}

}

@Test

**public** **void** testGetAllStockMes() {

List<StockPO> **pos** = stockDataService.getAllStockMes();

**if**(pos==**null**){

*fail*("testGetAllStockMes() --NULL");

}

}

@Test

**public** **void** testGetTimeSharingPOs() {

MyDate **today** = DateCalculator.*getToDay*();

String **code** = "sh600216";

String **dateString** = today.DateToString();

List<TimeSharingPO> **pos** = stockDataService.getTimeSharingPOs(code);

**if**(pos==**null**){

*fail*("testGetTimeSharingPOs() --NULL");

}**else** **if**(!pos.get(0).nowTime.DateToString().equals(dateString)){

*fail*("testGetTimeSharingPOs() --date");

}

}

@Test

**public** **void** testUpdateAllMes() {

MyDate **yestoday** = DateCalculator.*getAnotherDay*(-1);

String **dateString** = yestoday.DateToString();

stockDataService.updateAllMes();

List<StockPO> **pos** = stockDataService.getAllStockMes();

**if**(pos==**null**){

*fail*("testUpdateAllMes()--NULL");

}**else** **if**(!pos.get(0).getDate().equals(dateString)){

*fail*("testUpdateAllMes()--date");

}

}

}

2.BenchMarkDSImpl\_Test

**public** **class** BenchMarkDSImplTest {

BenchMarkDataService dataService ;

@Before

**public** **void** setUp() **throws** Exception {

dataService=BenchMarkDSImpl.*getBenchMarkDSImpl*();

}

@After

**public** **void** tearDown() **throws** Exception {

dataService=**null**;

}

@Test

**public** **void** testGetAllBenchMarks() {

String **code** = "000001";

List<String> **benchcodeStrings** = dataService.getAllBenchMarks();

**if**(benchcodeStrings==**null**){

*fail*("testGetAllBenchMarks()---null");

}**else** **if**(benchcodeStrings.size()!=14){

*fail*("testGetAllBenchMarks()---size");

}**else** **if**(!benchcodeStrings.get(0).equals(code)){

*fail*("testGetAllBenchMarks()---first code");

}

}

@Test

**public** **void** testGetBenchMesString() {

String [] **codes** = {"000001","399001","399005","00000","1111111"};

List<String> **errorCode** = **new** ArrayList<String>();

List<BenchMarkPO> **benchPOs** = **new** ArrayList<>();

**for**(**int** **i**=0; i<codes.length;i++){

BenchMarkPO **po** = dataService.getBenchMes(codes[i]);

**if**(po==**null**){

errorCode.add(codes[i]);

}**else**{

benchPOs.add(po);

}

}

**if**(!errorCode.get(0).equals(codes[3])||!errorCode.get(1).equals(codes[4])){

*fail*("testGetBenchMesString()---errorCodes can't be properly handled");

}**else** **if**( !benchPOs.get(0).getCode().equals(codes[0])){

*fail*("testGetBenchMesString()---benchmarkpo code changed");

}

}

@Test

**public** **void** testGetBenchMesStringMyDate() {

MyDate **date** =DateCalculator.*getAnotherDay*(-1);

String **code** = "000001";

String **nameString** = "上证综指";

String **dateString** = date.DateToString();

BenchMarkPO **po** = dataService.getBenchMes(code,date);

**if**(po==**null**){

*fail*("testGetBenchMesStringMyDate() --NULL");

}**else** **if**(!po.getCode().equals(code)){

*fail*("testGetBenchMesStringMyDate() --code");

}**else** **if**(!po.getName().equals(nameString)){

*fail*("testGetBenchMesStringMyDate() --name");

}**else** **if**(!po.getDate().equals(dateString)){

*fail*("testGetBenchMesStringMyDate()--date");

}

}

@Test

**public** **void** testGetBenchMesStringMyDateMyDate() {

MyDate **endDate** =DateCalculator.*getAnotherDay*(-1);

MyDate **startDate** = DateCalculator.*getAnotherDay*(-30);

String **code** = "000001";

String **nameString** = "上证综指";

String **dateString** = startDate.DateToString();

List<BenchMarkPO> **pos** = dataService.getBenchMes(code,startDate,endDate);

**if**(pos==**null**){

*fail*("testGetBenchMesStringMyDateMyDate() --NULL");

}**else** **if**(!pos.get(0).getCode().equals(code)){

*fail*("testGetBenchMesStringMyDateMyDate()--code");

}**else** **if**(!pos.get(0).getName().equals(nameString)){

*fail*("testGetBenchMesStringMyDateMyDate() --name");

}**else** **if**(!pos.get(0).getDate().equals(dateString)){

*fail*("testGetBenchMesStringMyDateMyDate() --date");

}

}

@Test

**public** **void** testGetAllBenchMes() {

List<BenchMarkPO> **pos** = dataService.getAllBenchMes();

**if**(pos==**null**){

*fail*("testGetAllBenchMes()--null");

}

}

}

3. OptionalStockDSImpl\_Test003

**public** **class** OptionalStockDSImplTest {

OptionalStockDataService dataservice;

@Before

**public** **void** setUp() **throws** Exception {

dataservice = OptionalStockDSImpl.*getOptionalStockDSImpl*();

}

@After

**public** **void** tearDown() **throws** Exception {

dataservice = **null**;

}

@Test

**public** **void** testGetSelectedStockCodes() {

List<String> **codes** = dataservice.getSelectedStockCodes();

**if** (codes == **null**) {

*fail*(" testGetSelectedStockCodes()--null");

}

}

@Test

**public** **void** testGetOptionalStocks() {

Iterator<StockPO> **poIterator** = dataservice.getOptionalStocks();

**if** (poIterator == **null**) {

*fail*("testGetOptionalStocks()--null");

} **else** **if** (!poIterator.hasNext()) {

*fail*("testGetOptionalStocks()--size");

}

}

@Test

**public** **void** testDeleteOptionalStock() {

String **codes**[] = { "sh600023", "sh600021", "sh50012" };

**if** (!dataservice.deleteOptionalStock(codes[0])) {

*fail*("testDeleteOptionalStock()--can't delete code t" + "hat does exist!");

} **else** {

**if** (dataservice.deleteOptionalStock(codes[0])) {

*fail*("testDeleteOptionalStock()--try to delete code that doesn't exist!");

}

}

dataservice.addOptionalStock(codes[0]);

**if** (dataservice.deleteOptionalStock(codes[2])) {

*fail*("testDeleteOptionalStock()-- try to delete code that doesn't exist!");

}

}

@Test

**public** **void** testAddOptionalStock() {

String **code** = "sz002592";

dataservice.addOptionalStock(code);

List<String> **codes** = dataservice.getSelectedStockCodes();

**if** (!codes.contains(code)) {

*fail*("testAddOptionalStock()--fail to add stock");

} **else** {

dataservice.deleteOptionalStock(code);

}

}

@Test

**public** **void** testClearOptionalStocks() {

dataservice.clearOptionalStocks();

List<String> **codes** = dataservice.getSelectedStockCodes();

**if** (codes == **null**) {

*fail*("testClearOptionalStocks()--null");

} **else** **if** (codes.size() != 0) {

*fail*("testClearOptionalStocks()--size");

}

String **oldCodes**[] = { "sh600012", "sh600019", "sh600028", "sh600048", "sh600007", "sh600018", "sh600026",

"sh600016", "sh600021", "sh600023" };

**for**(String **code**:oldCodes){

dataservice.addOptionalStock(code);

}

}

}

4. StockBLImpl\_Test004

**public** **class** StockBLImplTest {

**private** StockBLService bl;

**enum** testValue {};

@Before

**public** **void** setUp() **throws** Exception {

bl = BusinessFactory.*getStockBLService*();

}

@Test

**public** **void** getAllStocks() **throws** Exception {

// test in other method

}

@Test

**public** **void** getSortStocks() **throws** Exception {

Iterator<StockVO> **vos** = bl.getSortStocks(**true** , Stock\_Attribute.***low***);

**double** **last** = Double.***MIN\_VALUE***;

last = vos.next().low;

**while**(vos.hasNext()){

**double** **temp** = vos.next().low;

**if**(temp < last){

*fail*("Can not pass the sort of low");

}**else**{

last = temp;

}

}

}

/\*\*

\* 本测试主要测试对600XXX的股票进行升序排序是否有问题

\* **@throws** Exception

\*/

@Test

**public** **void** getSortStocksInScope() **throws** Exception {

Iterator<StockVO> **vos** = bl.getStocksByStockCode("600");

List<String> **vostr** = **new** ArrayList<>();

**while**(vos.hasNext()){

vostr.add(vos.next().code);

}

vos = bl.getSortStocksInScope(**true** , Stock\_Attribute.***high*** , vostr);

**double** **last** = Double.***MAX\_VALUE***;

last = vos.next().high;

**while**(vos.hasNext()){

**double** **temp** = vos.next().high;

**if**(temp > last){

*fail*("Can not pass the sort of high in given scope");

}**else**{

last = temp;

}

}

}

@Test

**public** **void** getRecentStocks() **throws** Exception {

// not need to test

}

@Test

**public** **void** getTodayStockVO() **throws** Exception {

// not need to test

}

@Test

**public** **void** getStocksByTime() **throws** Exception {

// not need to test

}

@Test

**public** **void** getStocksByStockCode() **throws** Exception {

}

@Test

**public** **void** getDayOHLC\_Data() **throws** Exception {

Iterator<StockVO> **vos** = bl.getStocksByTime("sh600300", MyTime.*getAnotherDay*(-30), MyTime.*getAnotherDay*(0));

List<Double> **lows** = **new** ArrayList<Double>();

**while**(vos.hasNext()){

lows.add(vos.next().low);

}

List<OHLC\_VO> **tmp** = bl.getDayOHLC\_Data("sh600300", MyTime.*getAnotherDay*(-30), MyTime.*getAnotherDay*(0));

**for** (**int** **i** = 0; i < 15; i++) {

**if**(!lows.get(i).equals(tmp.get(i).low)){

*fail*("Fail to get OHLC\_DATA correctly.");

}

}

}

@Test

**public** **void** getWeekOHLC\_Data() **throws** Exception {

}

@Test

**public** **void** getMonthOHLC\_Data() **throws** Exception {

}

@Test

**public** **void** getSharingVOs() **throws** Exception {

}

@Test

**public** **void** getDayDealVOs() **throws** Exception {

}

@Test

**public** **void** getWeekDealVOs() **throws** Exception {

}

@Test

**public** **void** getMonthDealVOs() **throws** Exception {

}