**NBA STATISTIC SYSTEM**

**单元测试用例**



南京大学软件学院Skip to my lou小组

2015-4-20

**测试用例列表：**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试用例ID | 测试用例描述 | 输入数据 | 预期输出 | 实际输出 | Pass/Fail | 备注 |
| TestData\_Handler-001 | Data\_Handler中方法的测试 |  |  |  |  |  |
| TestHeapSortByInt-001 | HeapSortByInt中方法的测试 |  |  |  |  |  |
| TestPlayer\_Handler-001 | Player\_Handler中方法的测试 |  |  |  |  |  |
| TestTeam\_Handler-001 | Team\_Handler中方法的测试 |  |  |  |  |  |

**相关的Junit源码文件：**

**public** **class** Data\_HandlerTest {

Data\_Handler dh;

@Before

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

dh = Data\_Handler.*getInstance*();

}

@Test

**public** **void** testGetPlayers() {

*assertTrue*(dh.getPlayers()!=**null**);

}

@Test

**public** **void** testGetTeams() {

*assertTrue*(dh.getTeams()!=**null**);

}

}

**public** **class** HeapSortByIntTest {

HeapSortByInt hsbi;

@Before

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

hsbi = **new** HeapSortByInt();

}

@Test

**public** **void** testHeapSort() {

**int** a[] [] = {{1,0},{3,1},{53,2},{32,3}};

**int** b[] [] = {{53,2},{32,3},{3,1},{1,0}};

HeapSortByInt.*heapSort*(a);

**for**(**int** i =0;i<4;i++)

{

*assertTrue*(a[i][0]==b[i][0]&&a[i][1]==b[i][1]);

}

}

}

**public** **class** Player\_HandlerTest {

Player\_Handler player\_handler;

@Before

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

player\_handler = **new** Player\_Handler();

}

@Test

**public** **void** testGetPlayerByName() {

*assertTrue*(player\_handler.getPlayerByName("Aaro")==**null**);

*assertTrue*(player\_handler.getPlayerByName("Aaron Gray")!=**null**);

}

@Test

**public** **void** testGetAllPlayer() {

*assertTrue*(player\_handler.getAllPlayer().size()==448);

*assertTrue*(player\_handler.getAllPlayer().size()!=221);

}

@Test

**public** **void** testGetPlayersByTeam() {

*assertTrue*(player\_handler.getPlayersByTeam("ATL")!=**null**);

}

@Test

**public** **void** testFilterPlayersBy() {

*assertTrue*(player\_handler.filterPlayersBy("G", "All", "All", 20)!=**null**);

}

@Test

**public** **void** testHotPlayerDaily() {

*assertTrue*(player\_handler.hotPlayerDaily("score")!=**null**);

*assertTrue*(player\_handler.hotPlayerDaily("sdad")==**null**);

}

@Test

**public** **void** testProgressFastPlayerForTest() {

*assertTrue*(player\_handler.progressFastPlayerForTest("score")!=**null**);

}

@Test

**public** **void** testProgressFastPlayer() {

*assertTrue*(player\_handler.progressFastPlayer("score")!=**null**);

*assertTrue*(player\_handler.progressFastPlayer("ss")==**null**);

}

@Test

**public** **void** testHotPlayerSeason() {

*assertTrue*(player\_handler.hotPlayerSeason("score")!=**null**);

*assertTrue*(player\_handler.hotPlayerSeason("ss")==**null**);

}

@Test

**public** **void** testGetPlayerRecentPerformance() {

*assertTrue*(player\_handler.getPlayerRecentPerformance("Aaron Gray")!=**null**);

*assertTrue*(player\_handler.getPlayerRecentPerformance("ss")==**null**);

}

@Test

**public** **void** testSortDailyPerformance() {

*assertTrue*(player\_handler.sortDailyPerformance("Aaron Gray")!=**null**);

*assertTrue*(player\_handler.sortDailyPerformance("ss")==**null**);

}

}

**public** **class** Team\_HandlerTest {

Team\_Handler team\_handler;

@Before

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

team\_handler = **new** Team\_Handler();

}

@Test

**public** **void** testGetTeamByAbbr() {

*assertTrue*(team\_handler.getTeamByAbbr("ATL")!=**null**);

*assertTrue*(team\_handler.getTeamByAbbr("AT")==**null**);

}

@Test

**public** **void** testGetAllTeam() {

*assertTrue*(team\_handler.getAllTeam().size()==30);

}

@Test

**public** **void** testGetTeamRecentPerformance() {

*assertTrue*(team\_handler.getTeamRecentPerformance("ATL")!=**null**);

}

@Test

**public** **void** testHotTeamSeason() {

*assertTrue*(team\_handler.hotTeamSeason("score")!=**null**);

}

}

**public** **class** Game\_HandlerTest {

Game\_Handler game\_handler;

@Before

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

game\_handler = **new** Game\_Handler();

}

@Test

**public** **void** testGetAllGames() {

*assertTrue*(game\_handler.getAllGames()!=**null**);

}

@Test

**public** **void** testGetGameDaily() {

*assertTrue*(game\_handler.getGameDaily()!=**null**);

}

@Test

**public** **void** testGetGameByDate() {

*assertTrue*(game\_handler.getGameByDate("2013-02-03")!=**null**);

}

@Test

**public** **void** testGetGameByLabel() {

*assertTrue*(game\_handler.getGameByLabel("13-14 02-03 AVL-LAK")==**null**);

}

@Test

**public** **void** testGetPerformance() {

*assertTrue*(game\_handler.getPerformance("13-14 02-03 AVL-LAK", "avl")==**null**);

}

}