

```

package nthx.util.struts.navigation.tests;

import junit.framework.TestCase;
import org.apache.log4j.Logger;
import nthx.util.struts.navigation.NavigableData;
import nthx.ilybra.book.Books;
import nthx.ilybra.book.Book;

/** Tests for testing {@link NavigableData NavigableData} class.
 *
 * @version $Id: NavigableDataTest.java,v 1.2 2003/08/20 01:05:59 nthx Exp $
 * @author nthx@irc.pl
 */
public class NavigableDataTest extends TestCase
{
    Logger logTest = Logger.getLogger(nthx.util.log.Log4jTest.class);

    //--- Constructors -----
    public NavigableDataTest(String arg)
    {
        super(arg);
    }

    .....

    //--- Tests here.. -----
    public void testNavigation()
    {
        .....
    }

    public void testNavigationPages()
    {
        .....
    }

    public void testGoCustomPage()
    {
        Books books = new Books();
        for (int i=0; i<100; i++)
            books.add(new Book("123"+i, "S.2409", "Book: " + i));

        assertEquals(100, books.size());

        NavigableData navigable = new NavigableData(books.getCollection(), 10);
        navigable.goCustomPage(3);

        assertEquals("items size", 10, navigable.getNavigableList().size());
        assertEquals("items number", 10, navigable.getItemsNumber());
        assertEquals("Bad 'actualPage'", 3, navigable.getActualPage());
        assertEquals("Bad 'nextPage'", 4, navigable.getNextPage());
        assertEquals("Bad 'prevPage'", 2, navigable.getPrevPage());
        assertEquals("pages number", 10, navigable.getPagesNumber());
        assertEquals("items per page", 10, navigable.getItemsPerPage());
        assertEquals("Bad lastPage", 9, navigable.getLastPage());
        assertEquals("Pages error", 5, navigable.getPages().size());

        navigable.goForward();

        assertEquals("Bad 'actualPage'", 4, navigable.getActualPage());
    }
}

```

```

assertEquals("Bad 'nextPage'", 5, navigable.getNextPage());
assertEquals("Bad 'prevPage'", 3, navigable.getPrevPage());
assertEquals("pages number", 10, navigable.getPagesNumber());
assertEquals("items per page", 10, navigable.getItemsPerPage());
assertEquals("Bad lastPage", 9, navigable.getLastPage());
assertEquals("Pages error", 4, navigable.getPages().size());

navigable.goCustomPage(0);

assertEquals("Bad 'actualPage'", 0, navigable.getActualPage());
assertEquals("Bad 'nextPage'", 1, navigable.getNextPage());
assertEquals("Bad 'prevPage'", 0, navigable.getPrevPage());
assertEquals("pages number", 10, navigable.getPagesNumber());
assertEquals("items per page", 10, navigable.getItemsPerPage());
assertEquals("Bad lastPage", 9, navigable.getLastPage());
assertEquals("Pages error", 8, navigable.getPages().size());
}

public void testHumanPage()
{
    .....
}

public void testActualPage()
{
    .....
}
}

```