**Dynamic Tests**

**package** com.app.junitDemo;

**import** java.util.Arrays;

**import** java.util.Collection;

**import** **static** org.junit.jupiter.api.Assertions.\*;

**import** org.junit.jupiter.api.DynamicTest;

**import** org.junit.jupiter.api.TestFactory;

**public** **class** DynamicTests {

@TestFactory

**public** Collection<DynamicTest> dynamictests() {

**return** Arrays.*asList*(

DynamicTest.*dynamicTest*("Simple Test1", () ->*assertTrue*(**true**)),

DynamicTest.*dynamicTest*("Simple Test1", () ->*assertTrue*(**true**)),

DynamicTest.*dynamicTest*("Exception Handling", () ->{**throw** **new** Exception("Exception in Dynamic Test");})

);

}

}

**import** java.util.Arrays;

**import** java.util.List;

**import** java.util.stream.Stream;

**import** **static** org.junit.jupiter.api.DynamicTest.*dynamicTest*;

**import** org.junit.jupiter.api.DynamicContainer;

**import** org.junit.jupiter.api.DynamicNode;

**import** org.junit.jupiter.api.TestFactory;

**import** **static** org.junit.jupiter.api.Assertions.\*;

**public** **class** DynamicContainerDemo {

**private** List<String> inputString()

{

**return** Arrays.*asList*("race","mom","dad","test");

}

@TestFactory

**public** Stream<DynamicNode> dynamiccontmethod()

{

**return** inputString().stream().map(input ->

DynamicContainer.*dynamicContainer*("Container for "+input, Stream.*of*

(*dynamicTest*("not null",() -> *assertNotNull*(input)),

DynamicContainer.*dynamicContainer*("Properties Test", Stream.*of*(

*dynamicTest*("length>0", () -> *assertTrue*(input.length()>0)),

*dynamicTest*("not empty ", () -> *assertFalse*(input.isEmpty()))

)))

));

}

}