HAMCREST FRAMEWORK

Link [reference](http://hamcrest.org/JavaHamcrest/javadoc/2.2/) API:

1. What is hamcrest

* Hamcrest is framework for writing matcher object
* The matcher object can be used to add validation in the unit-tests
* Hamcrest framework provide a lot of in-built matcher object
* You can also create your own custom matcher object

2. Other test framework

Hamcrest has been designed from the outset to integrate with different unit testing frameworks.

Hamcrest can also be used with mock objects frameworks by using adaptors to bridge from the mock objects framework's concept of a matcher to a Hamcrest matcher Hamcrest comes with a library of useful matchers. Here are some of the most important ones.

* Core
  + anything - always matches, useful if you don't care what the object under test is
  + describedAs - decorator to adding custom failure description
  + is - decorator to improve readability - see "Sugar", below

Demo

Text

Description automatically generated

* Logical
  + allOf - matches if all matchers match, short circuits (like Java &&)
  + anyOf - matches if any matchers match, short circuits (like Java ||)
  + not - matches if the wrapped matcher doesn't match and vice versa

Text

Description automatically generated

* Object
  + equalTo - test object equality using Object.equals
  + hasToString - test Object.toString
  + instanceOf, isCompatibleType - test type
  + notNullValue, nullValue - test for null
  + sameInstance - test object identity

Demo

* + Graphical user interface, text

    Description automatically generated
* Beans
  + hasProperty - test JavaBeans properties

Text

Description automatically generated

* Collections
  + array - test an array's elements against an array of matchers
  + hasEntry, hasKey, hasValue - test a map contains an entry, key or value
  + hasItem, hasItems - test a collection contains elements
  + hasItemInArray - test an array contains an element

Demo

Text

Description automatically generated

* Number
  + closeTo - test floating point values are close to a given value
  + greaterThan, greaterThanOrEqualTo, lessThan, lessThanOrEqualTo - test ordering

Demo

Text

Description automatically generated

* Text
  + equalToIgnoringCase - test string equality ignoring case
  + equalToIgnoringWhiteSpace - test string equality ignoring differences in runs of whitespace
  + containsString, endsWith, startsWith - test string matching

Demo

Text

Description automatically generated

* **Sugar**
* Hamcrest strives to make your tests as readable as possible. For example, the is matcher is a wrapper that doesn't add any extra behavior to the underlying matcher. The following assertions are all equivalent:

Example:

assertThat(theBiscuit, equalTo(myBiscuit));

assertThat(theBiscuit, is(equalTo(myBiscuit)));

assertThat(theBiscuit, is(myBiscuit));

* The last form is allowed since is(T value) is overloaded to return is(equalTo(value)).