ToIntFunction Functional Interface

Before we begin to discuss about ToIntFunction interface we need to know about Function interface. Function interface takes argument T as input and return result R. BiFunction is two-arity version of Function interface where it can take two different arguments T and U as input and returns result as R. I would highly recommend you to read it. I also have written on [higher order functions](http://data-structure-learning.blogspot.com/2015/07/higher-order-functions-using-function.html).

ToIntFunction interface represents a function that accepts argument T and returns int valued result.

|  |
| --- |
| **ToIntFunction Interface Declaration**  **public** **interface** ToIntFunction<T>  ToIntFunction interface represents a function that accepts input of argument T and returns int valued result. This is int producing primitive specialization of Function interface. |
| **applyAsInt()** **method**  **int** applyAsInt(T value);  T is the input for the method applyAsInt(). This function applies the given argument and returns int result. Let us take simple example of accepting a string and return its length.  ToIntFunction<String> toIntFunc = (str) -> str.length();  **int** result = toIntFunc.applyAsInt("Monday");  System.***out***.println(result); //Output 6 |

That’s all on ToIntFunction interface.

Read about important java.util.function package’s interface [here](http://data-structure-learning.blogspot.com/p/functional-programming-in-java.html). [Consumer](http://data-structure-learning.blogspot.com/2015/07/java-lambda-consumer-functional.html), [Function](http://data-structure-learning.blogspot.com/2015/07/java-lambda-function-functional.html), [Supplier](http://data-structure-learning.blogspot.com/2015/07/java-lambda-supplier-functional.html), [BinaryOperator](http://data-structure-learning.blogspot.com/2015/07/java-lambda-binaryoperator-functional.html) & [Predicate](http://data-structure-learning.blogspot.com/2015/07/java-lambda-predicate-functional.html) Functional Interfaces. I have also written on [High Order functions](http://data-structure-learning.blogspot.com/2015/07/higher-order-functions-using-function.html) using Function functional interface.