What is IEnumerable in C#?

IEnumerable in C# is an interface that defines one method, GetEnumerator which returns an IEnumerator interface. This allows readonly access to a collection then a collection that implements IEnumerable can be used with a for-each statement.

Key Points

1. IEnumerable interface contains the System.Collections.Generic namespace.
2. IEnumerable interface is a generic interface which allows looping over generic or non-generic lists.
3. IEnumerable interface also works with linq query expression.
4. IEnumerable interface Returns an enumerator that iterates through the collection.

Let us implement the IEnumerable interface in a class as:

1. **public** **class** Customer : IEnumerable
2. {
4. **public** IEnumerator GetEnumerator()
5. {
6. **throw** **new** NotImplementedException();
7. }
8. }

In the above example, you have seen that after implementing the IEnumerable Interface there is method called GetEnumerator along with interface IEnumerator which helps to get current elements from the collection.

Methods of IEnumerator Interface

 IEnumerator is an interface which helps to get current elements from the collection, it has the following two methods

1. MoveNext()
2. Reset()

**MoveNext()**

Sets the enumerator to the next element of the collection; it Returns true if the enumerator was successfully set to the next element and false if the enumerator has reached the end of the collection.

**Reset()**

Sets the enumerator to its initial position, which is before the first element in the collection.

Properties of IEnumerator Interface

IEnumerator Interface has a property named  Current which returns the current element in the collection.

Let us implement the IEnumerator Interface in class as:

1. **public** **class** Customer : IEnumerator
2. {
4. **public** **object** Current
5. {
6. **get** { **throw** **new** NotImplementedException(); }
7. }
9. **public** **bool** MoveNext()
10. {
11. **throw** **new** NotImplementedException();
12. }
14. **public** **void** Reset()
15. {
16. **throw** **new** NotImplementedException();
17. }
18. }

In the above class we have implemented the IEnumerator interface which shows the above two methods and one property as we have already explained.

IEnumerable vs IEnumerator interface

While reading these two names, it can be confusing, so let us understand the difference between these two.

1. IEnumerable and IEnumerator are both interfaces.
2. IEnumerable has just one method called GetEnumerator. This method returns another type which is an interface that interface is IEnumerator.
3. If we want to implement enumerator logic in any collection class, it needs to implement IEnumerable interface (either generic or non-generic).
4. IEnumerable has just one method whereas IEnumerator has two methods (MoveNext and Reset) and a property Current.

For our understanding, we can say that  IEnumebale is a  box that contains IEnumerator inside it.