Collection Best Practices

C# Tips

SWIPE >>>

Richard Crane

Collection Best Practices

Use `IEnumerable`, `IReadOnlyCollection`, or `IReadOnlyList`:

Choose IEnumerable<T> if the consumer only needs to iterate through the collection, IReadOnlyCollection<T> for access with a known count, or IReadOnlyList<T> for indexed access.

Avoid returning null:

Return an empty list or enumerable instead of null to prevent null reference exceptions.

Return Immutable Collections:

For safety and to prevent modifications, consider using immutable collections. This ensures the data you expose cannot be altered.

Consider Thread Safety:

Ensure safety for enumerations in multi-threaded environments where other threads may alter the collection.

SWIPE >>>

Richard Crane

Bad Example

```
public List<Employee> GetEmployeesAndAlumni()
{
    var employees = new List<Employee> () { ... };

    return employees;
}
```

SWIPE >>>

Richard Crane

Good Example

```
public IReadOnlyList<Employee> GetEmployeesAndAlumni()
{
    var employees = new List<Employee> () { ... };

    return employees.AsReadOnly();
}
```

SWIPE >>>

Richard Crane

Get more tips at WickedProgrammer.com