Creating Generic Methods and Delegates



Thomas Claudius Huber
Software Developer

@thomasclaudiush www.thomasclaudiushuber.com

Module Outline



- Work with generic methods
 - Start with non-generic method
 - Create a generic method
 - Build a generic extension method
- Work with generic delegates
 - Start with non-generic delegate
 - Create a generic delegate
 - Use the Action<T> delegate
 - Create events with EventHandler<T>





Add a non generic method



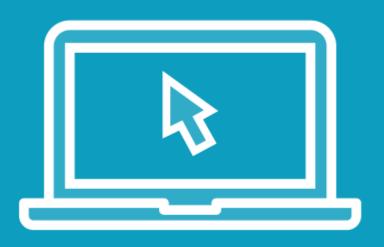
Create a generic method



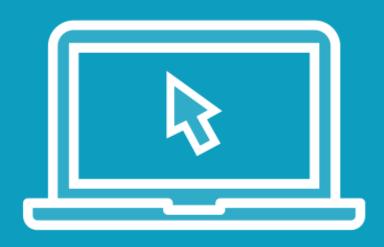


Build a generic extension method





Write a generic method with return value



Add a non generic delegate





Create a generic delegate



Understand variance with generic delegates

Use the Action<T> Delegate

```
public delegate void Action<in T>(T arg);
public delegate void Action<in T1, in T2>(T1 arg1, T2 arg2);

public delegate TResult Func<out TResult>();
public delegate TResult Func<in T, out TResult>(T arg);

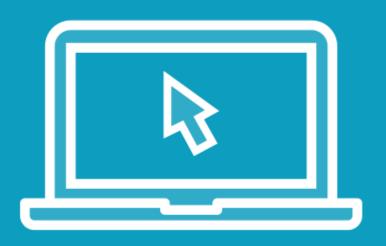
public delegate TResult Func<in T1, in T2, out TResult>(
    T1 arg1, T2 arg2);
```



Use the Action<T> Delegate

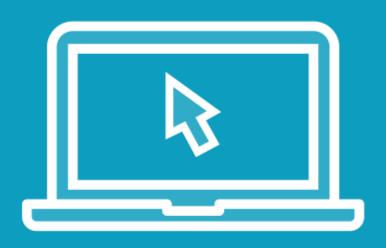
```
public delegate void Action<in T>(T arg);
public delegate void ItemAdded<in T>(T item);
```





Use the Action<T> delegate





Create events with EventHandler<T>

Summary



- Create and use a generic method
 - Build a generic extension method for a generic interface
 - Use type constraints
- Create and use a generic delegate
 - Understand covariance and contravariance
- Work with existing delegates like Action<T> and EventHandler<T>



Up Next:

Knowing the Special Cases with Generics

