# Capitolo 3

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

Being able to provide a way to extend or replace functionality at runtime is going to prove very powerful

I am going to introduce two methods that I want to be invoked when the order reaches two different states

Quando ha finite l’inizializzazione devo chiamare il messaggio SendMessageToWarehouse e quando l’ordine è finito SendConfirmationEmail

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

Providing a way to get a callback when an order reaches a certain state, allow the consumer to act on that information.

This could be to populate the UI with some information about which order was most recently processed.

When a order reach a certain state we want to delegate the implementation to someone else

This could be to populate the UI with some information about which order was most recentily processed

The method initialize and Process will both try to invoke two corresponding delegates

In other words, they will try to invoke methods that we are simply pointing to, but do not belong to this class

The Idea is that the delegate that we create, SendMessageToWarehouse and SendConfirmationEmail, matches those method signatures.

When the order processor is create, we need to provide a pointer to those methods.

This class can then invoke the delegate, which will run the methods we have pointed to.

Immagine che contiene testo

Descrizione generata automaticamente

The whole point of the delegate is to allow it to point to a method that lives somewhere else and might not even be know when this portion of the code is compiled.

OnOrderInitialized can reference any void method without parameters

This property could now point to any method that correspond with the meghod signature that we defined when declaring the delegate

The method you point to does have the signature of the delegate

Immagine che contiene testo

Descrizione generata automaticamente

Si veda come , dalla figura sotto che il delegato viene invocato e , ciò che viene eseguito è una funzione che è esterna alla classe in cui è presente il delegato

Immagine che contiene tavolo

Descrizione generata automaticamente

# Capitolo 4

Nel capitolo precedente abbiamo utilizzato il delegato per comunicare che il processamento dell’ordine ha raggiunto un certo stato

L’idea di segnalare che qualcosa ha raggiunto un certo stato oppure qualcosa è successo può essere visto come publishing or raising an event

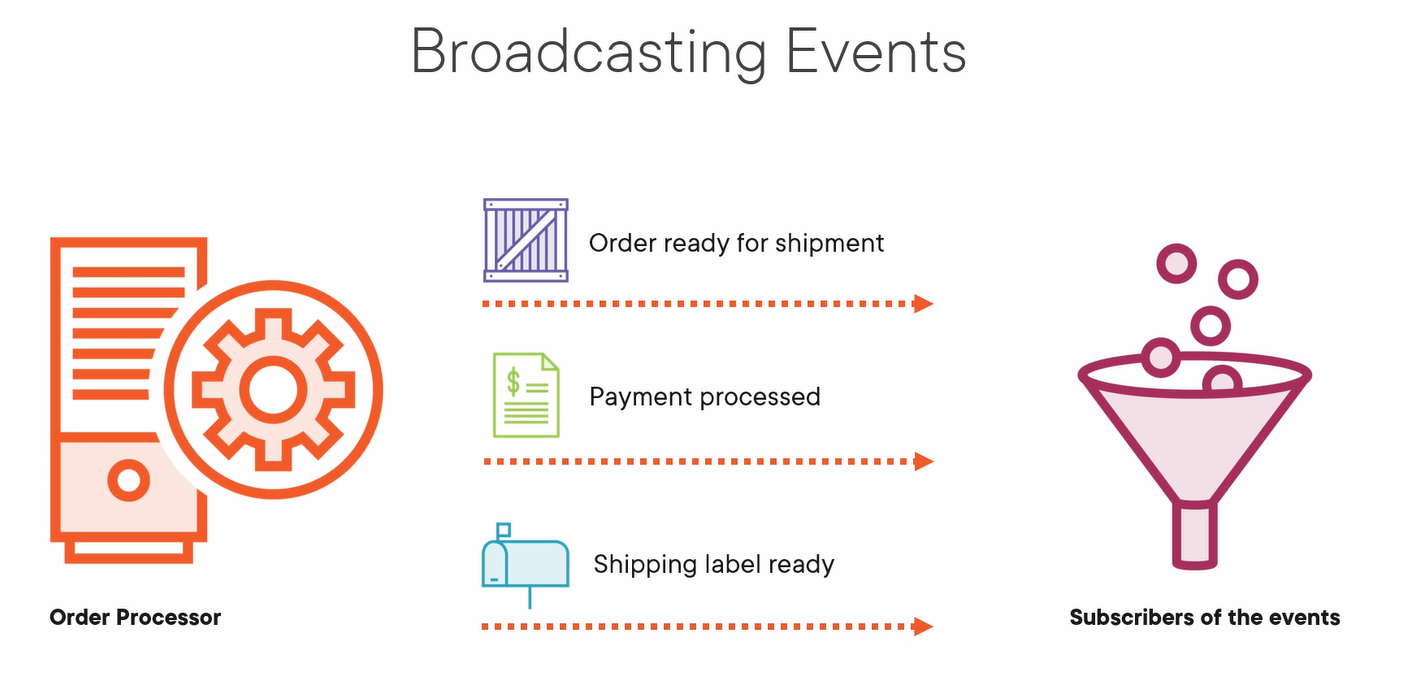


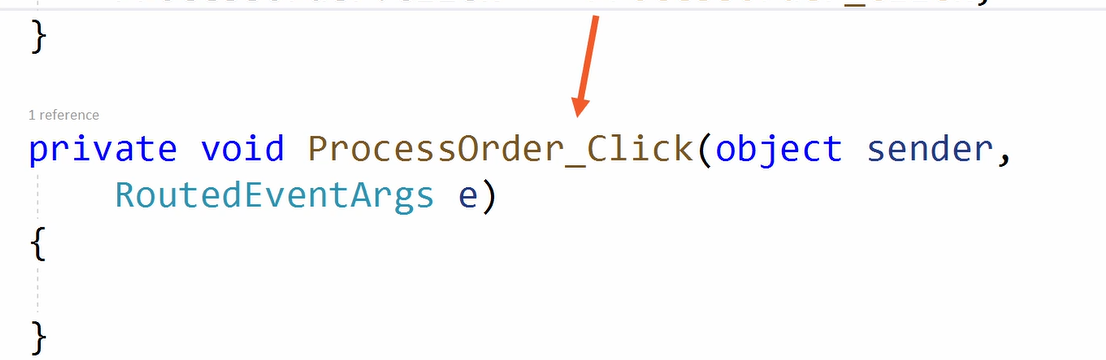
Immagine che contiene testo

Descrizione generata automaticamente

Immagine che contiene testo

Descrizione generata automaticamente

It uses a delegate to allow you to reference a method that is executed when you press the button



The only part of the system that should be in charge of invoking that particular delegate is the button control itself

It would be quite odd( abbstanza chiaro) if any code throughout the application were able to invoke the click delegate

You only want that delegate to be invoked when that actual button on the UI was pressed