

---

# Backbone Events

---

- ❖ They are not restricted to *Models*, *Collections*, *Views*, and *Routers*. You can attach events to any object
- ❖ To bind an event to an object you use Underscore's extend method: `_.extend(object, Backbone.Events)`
- ❖ The `.on` function accepts three parameters:
  - ❖ The event name
  - ❖ The handler function
  - ❖ An optional context where the event will be applied
- ❖ Multiple events can be accepted by the handler function, when they are separated by spaces

---

# Unbinding Events

---

- ❖ Used if you no longer need the event to fire
- ❖ Defined by the *.off* function
- ❖ The *.off* function can be used in the following ways:
  - ❖ To unbind all handlers of an event: `object.off("event")`
  - ❖ To unbind a specific handler: `object.off("event", handler)`
  - ❖ To unbind all events tied to a specific handler:  
`object.off(null, handler)`
  - ❖ To unbind all events and all handlers of an object:  
`object.off()`

---

# Listening Once

---

- ❖ Example usage: when you want to capture events during object initialization only
- ❖ Saves you from having to bind and unbind the event, if you only want to listen for the event once.
- ❖ Defined using the *.once* function

---

# Listening to events from other objects

---

- ❖ Example usage: when you want a view to listen and respond to changes in a model
- ❖ Defined using the *.listenTo(object,event,handler)*. Where *object* is the object where the event will happen
- ❖ Can be switched off using the *.stopListening(object)* method to stop listening to events of an object, or *.stopListening()* to stop listening to all events from all objects
- ❖ Will listen to only the first event using the *.listenToOnce* method

---

# Raising Events

---

- ❖ Used to manually raise an event without this event actually happening (faking an event)
- ❖ Invoked using the *object.trigger(event,{data:data})* method on an object
- ❖ You can optionally pass any data object(s) as the second parameter of the trigger function. This data becomes available to the event handler by being passed as a function parameter

---

# Creating your own events

---

- ❖ Example usage: You want to raise an event whenever a number of model attributes have been assigned specific values (the stock and the reserve of your store model have both reached zero)
- ❖ You can create a custom event on any object using *Backbone.on("object:event", handler)*
- ❖ Custom events have to be manually triggered
- ❖ They are normally triggered from within the handler functions of other built in events (often the *change* event)
- ❖ Any optional data passed in the trigger function is available to the handler by passing it as a parameter to the function