

Class

# UICollectionViewLayoutInvalidationContext

A context object that declares which parts of your layout need to be updated when the layout is invalidated.

## Declaration

```
class UICollectionViewLayoutInvalidationContext : NSObject
```

## Overview

Layout objects that are designed to support invalidation contexts can use the information in a `UICollectionViewLayoutInvalidationContext` object to optimize their behavior during the invalidation cycle. You can create an invalidation context object as a precursor to invalidating a layout object. After configuring the invalidation context object, pass it to the layout object's `invalidateLayout(with:)` method, which is responsible for using the context object to update the layout efficiently. The collection view also creates invalidation contexts in response to specific changes. For example, it creates an invalidation context when you change the layout or data source object, when you insert or delete items, and when you call the `reloadData()` method.

## Subclassing Notes

If you create your own custom layout objects, you can subclass `UICollectionViewLayoutInvalidationContext` and add properties to specify which aspects of your layout data can be invalidated separately. You must then design your layout object to check for these properties and update the layout appropriately.

For more information about how to support custom invalidation contexts in your layout objects, see [UICollectionViewLayout](#).

## Topics

### Invalidating the Collection View Data

var `invalidateEverything`: Bool

A Boolean that indicates that all layout data should be marked as invalid.

var `invalidateDataSourceCounts`: Bool

A Boolean that indicates whether the layout should ask for new section and item counts.

## Invalidating the Content Area

var `contentOffsetAdjustment`: CGPoint  
The delta value to be applied to the collection view's content offset.

var `contentSizeAdjustment`: CGSize  
The delta value to be applied to the collection view's content size.

## Invalidating Specific Items

func `invalidateItems(at: [IndexPath])`  
Adds the cells at the specified index paths to the list of invalid items.

func `invalidateSupplementaryElements(ofKind: String, at: [IndexPath])`  
Adds the supplementary views at the specified index paths to the list of invalid items.

func `invalidateDecorationElements(ofKind: String, at: [IndexPath])`  
Adds the decoration views at the specified index paths to the list of invalid items.

var `invalidatedItemIndexPaths`: [IndexPath]?  
An array of index paths representing the cells that were invalidated.

var `invalidatedSupplementaryIndexPaths`: [String : [IndexPath]]?  
A dictionary that identifies the supplementary views that were invalidated.

var `invalidatedDecorationIndexPaths`: [String : [IndexPath]]?  
A dictionary that identifies the decoration views that were invalidated.

## Invalidating the Order of Items

var `previousIndexPathsForInteractivelyMovingItems`: [IndexPath]?  
An array of index paths representing the previous location of moving items in the collection view.

var `targetIndexPathsForInteractivelyMovingItems`: [IndexPath]?  
An array of index paths representing the new location of moving items in the collection view.

var `interactiveMovementTarget`: CGPoint  
The current point used to determine the placement of moving items.

## Relationships

---

**Inherits From**      [NSObject](#)

---

**Conforms To**      [CVarArg](#), [Equatable](#), [Hashable](#)