返回当前控制的所有子控制集合、用来搜索出unwind目标控制器 首先对自己调用 childViewControllerContainingSegueSource方法查询,然后 对子控制器进行查询;容器控制器可以重写改变搜索顺序以及搜索结果 allowedChildViewControllersForUnwindingFromSource(_: 如果只重写了过期的方法 UIStoryboardUnwindSegueSource) -> [UIViewController] viewControllerForUnwindSegueAction:fromViewContro ller:sender:没有重写此方法,该方法就不会被调用 想要让控制器成为unwind目标控制器,则重写下面方法 canPerformUnwindSequeAction:fromViewCont roller:withSender: 新增API 返回包含有segue source对象信息的子控制器 自定义控制器应该在自己重写的方法 childViewControllerContainingSegueSource(_: allowedChildViewControllersForUnwindingF UIStoryboardUnwindSegueSource) -> UIViewController? romSource实现中调用该方法 不要重写该方法,注意从模态视图unwind的情况处理 控制器在segue unwinding期间收到该消息,默认实现为对 canPerformUnwindSegueAction(_: Selector, fromViewController: 当前对象的respondsToSelector的结果返回,并且允许重 UIViewController, withSender _: AnyObject) -> Bool 写,进行额外的检查 Unwind Segue 容器控制器应该重写该方法,设置在进行的unwind unwindForSegue(_: UIStoryboardSegue, segue;后面的控制器是与unwind segue相关联,且离目 towardsViewController: UIViewController) 标控制器层次关系最近的控制器 已过期...返回一个将unwind的segue对象,通过方法 unwindForSegue:towardViewController可以获 segueForUnwindingToViewController(_: UIViewController, 得目标控制器,若执行storyboard中的unwind segue, fromViewController: UIViewController, identifier: String?) -> 重写此方法并不返回nil,在运行时就会执行所给定返回 UIStoryboardSegue? 的segue对象的perform,不会执行IB界面设置的Segue 类的对象. 废弃的API 返回能执行该方法返回Yes的直接子控制器对象或自身 viewControllerForUnwindSegueAction(_: Selector, canPerformUnwindSegueAction:fromViewCont fromViewController: UIViewController, withSender _: roller:withSender.该方法被重写则不会对其子控 制器进行unwind segue查询,直接对自己调用 AnyObject?) -> UIViewController? canPerformUnwindSequeAction:结束. 在目标控制器手动实现IBAction方 法,参数必为segue对象 使用 让在Storyboard界面上的源控制器 的操作控件与自定义的unwind方法 进行关联.

给关联的控制器设置转场代理

调用super.perform执行

Custom Segue

What's new in Storyboard