

3D Touch

Show your apps another dimension








Agenda

- Overview of 3D Touch and why it is awesome 🖐️
- Integration into your apps ➡️📱
- Deep insight in the APIs ✅

History

- 3D Touch, comes in every iPhone after 6s (Sept. 2015)
- Detects the pressure exerted on screen
- iOS 9 - APIs made available to developers

Why integrate 3D Touch?

- It's cool, duh! 
- Users can fast track features  
- 400 million+ iPhones with 3D Touch compatibility 
- Users can go directly to an action from app icon
- App Store visibility 



Ease of Usage



Where can you use 3D Touch? 🤔🤔

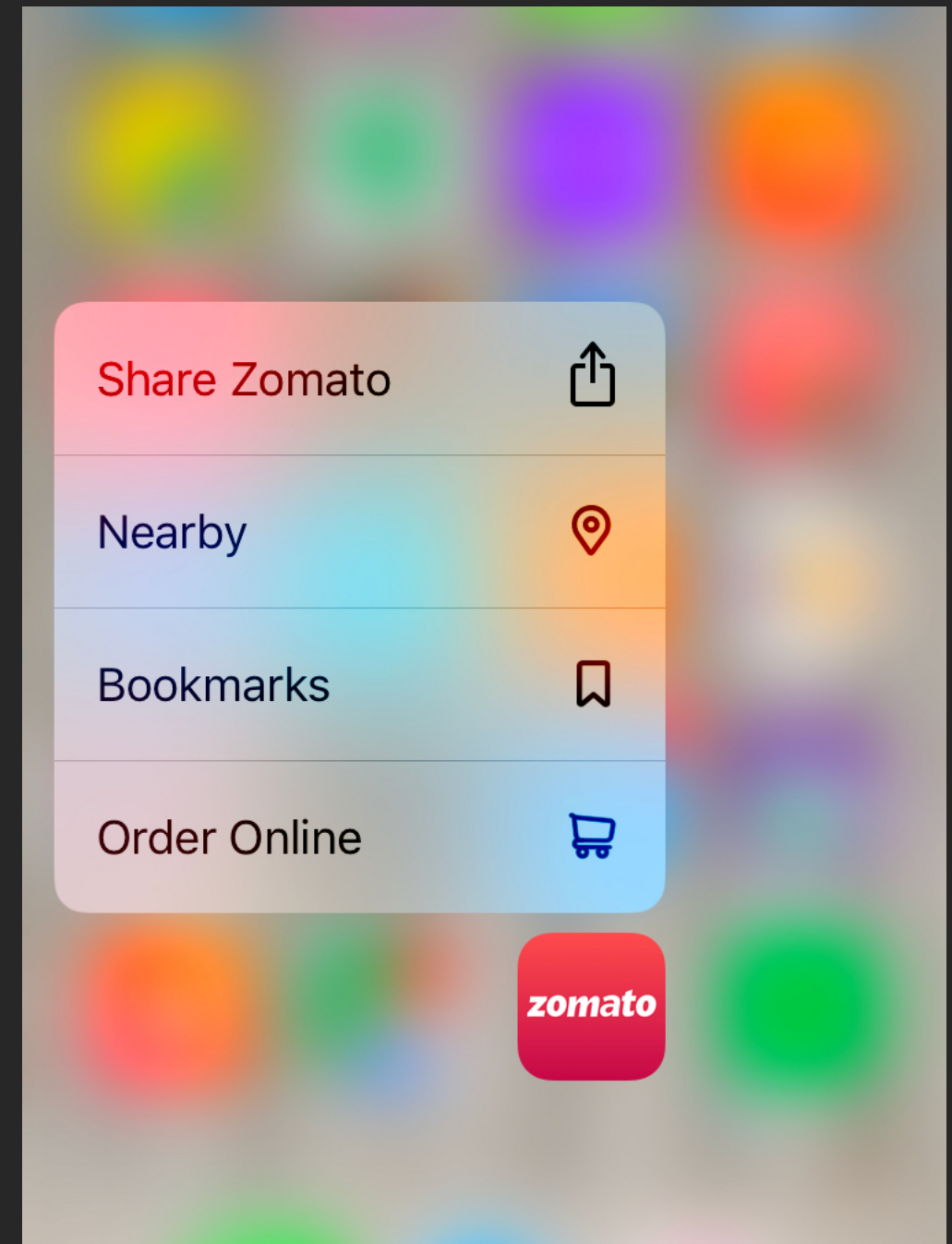
- Home Screen Quick Actions
- Peek and Pop
- Notification Content Extensions

Where can you use 3D Touch? 🤔🤔

- Home Screen Quick Actions
- Peek and Pop
- Notification Content Extensions

Home Screen Quick Actions

- Introduced in iOS 9
- Can have a total of 4 Actions
- Are of 2 types : Static and Dynamic



Static Home Screen Actions

- Declared in Info.plist
- Set once, Populated @ Compile Time
- Available as soon as the app is installed

▼ UIApplicationShortcutItems	⌵	Array	(1 item)
▼ Item 0		Dictionary	(4 items)
UIApplicationShortcutItemTitle		String	Search
UIApplicationShortcutItemType		String	\$(PRODUCT_BUNDLE_IDENTIFIER).search
UIApplicationShortcutItemIconFile		String	search
UIApplicationShortcutItemSubtitle		String	Search Recipes

Static Home Screen Actions

required

`UIApplicationShortcutItemType`

`UIApplicationShortcutItemTitle`

optional

`UIApplicationShortcutItemSubtitle`

`UIApplicationShortcutItemIconType`

`UIApplicationShortcutItemIconFile`

Dynamic Home Screen Actions

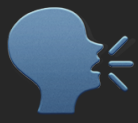


- Populated @ Runtime
- Will not appear until the user has used that feature
- Add to UIApplication shortcutsItems, when relevant.

Dynamic Home Screen Actions

```
let recentRecipe = UIApplicationShortcutType(type: "Recent",
                                             localizedTitle: recent.name,
                                             localizedSubtitle: recent.subname,
                                             icon: UIApplicationShortcutIconType(type: .time),
                                             userInfo: nil)
```

```
UIApplication.shared.shortcutItems = [recentRecipe]
```

How do Quick Actions work?

- AppDelegate receives a callback when an action is selected 
- Check UIApplicationShortcutItemType 
- Perform Action 

```
func application(_ application: UIApplication, performActionFor shortcutItem:
    UIApplicationShortcutItem, completionHandler: @escaping(Bool) -> Void) {

    switch shortcutItem.type {

        case "search":

            //deeplink to search page

        case "favourite":

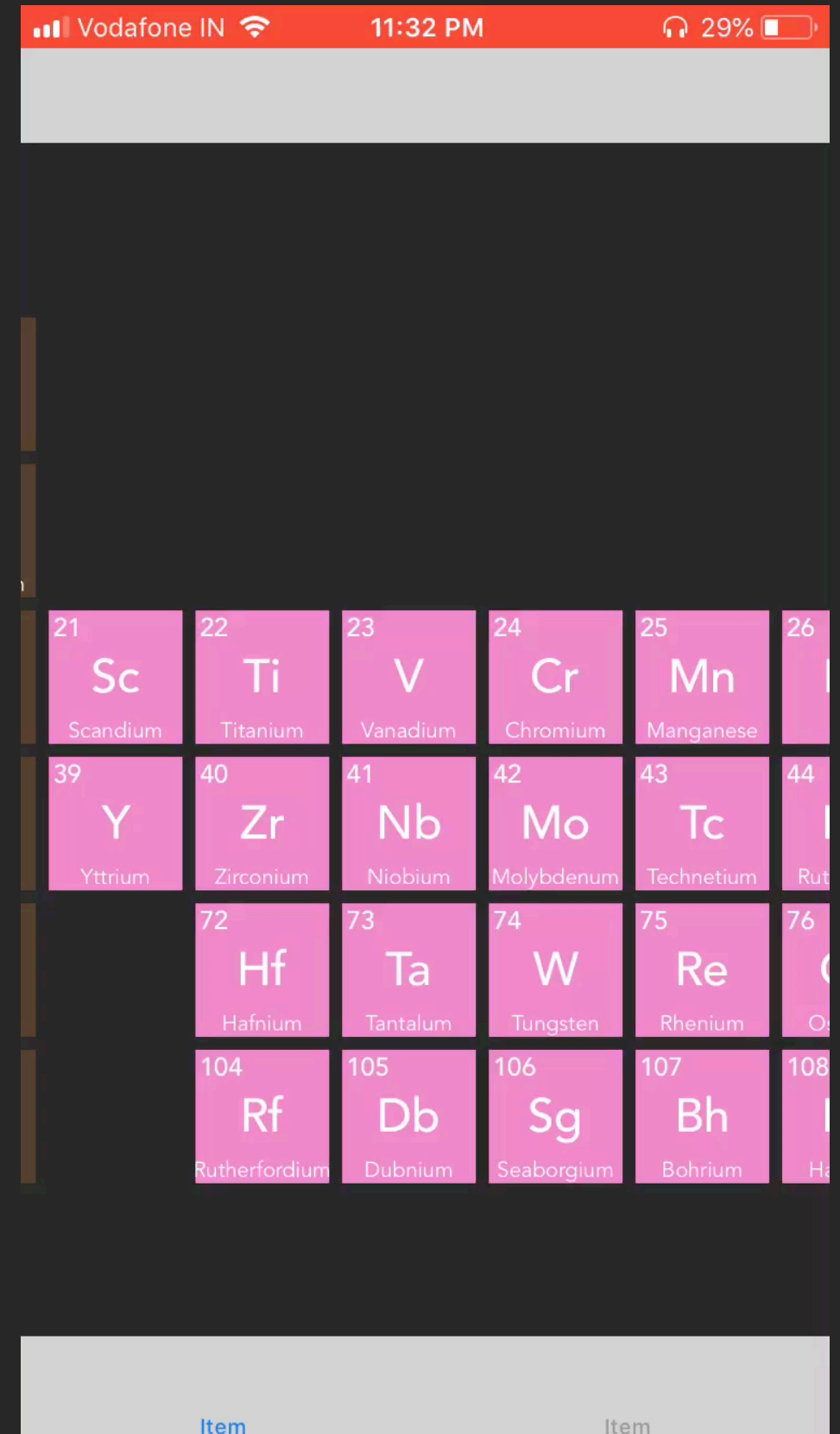
            //deeplink to favourite page

    }

}
```

Peek and Pop

- Useful for previewing content before navigation
- Should be implemented wherever content can be tapped to navigate - Recommended by Apple



Peek and Pop

Protocol Overview

`UIViewControllerPreviewingDelegate`

```
func previewingContext(viewControllerForLocation:)
```

```
func previewingContext(viewControllerToCommit:)
```

```
var previewActionItems
```


Check for 3D Touch 100



- Check @ runtime for 3D Touch

```
override func viewDidLoad() {  
    super.viewDidLoad()  
    if traitCollection.forceTouchCapability == .available {  
        registerForPreviewing(with: self, sourceView: collectionView)  
    }  
}
```

- Check @ runtime for 3D Touch

```
override func viewDidLoad() {  
    super.viewDidLoad()  
    if traitCollection.forceTouchCapability == .available {  
        registerForPreviewing(with: self, sourceView: collectionView)  
    }  
}
```

Back to Peek and Pop 🙈 🙈

Peek and Pop

```
func previewingContext(_ previewingContext: UIViewControllerForPreviewing,  
    viewControllerForLocation location: CGPoint) -> UIViewController? {  
  
    //return viewController for peek  
  
}
```

```
func previewingContext(_ previewingContext: UIViewControllerForPreviewing,  
    commit viewControllerToCommit: UIViewController) {  
  
    show(viewControllerToCommit, sender: self)  
  
}
```

Demo



Alternatives ?

- All devices do not support 3D Touch
- Reasons - iOS Version, iPhone Model, Settings
- Always provide an alternative if you can
- `UILongPressGestureRecognizer` (press and hold)

Quick Wins 🏆

- Save time for User 🚗
- Easy af
- Seems fancy and intuitive
- Grabs App Store Attention

Thank You! 😊

 **@soulful_swift**

 **@xorforce**

