

Code Feedback :)

Code Refactoring Suggestions:

- **Separation is key:** Separate the data fetching logic from the view controller. Create a dedicated data management class or use a ViewModel to handle data-related tasks. 😊
- **Remove Unused Code:**
 - Remove `@objc func dismissController () {}` , if you need to dismiss a controller you can do so using `dismiss(animated: true, completion: nil)`. Bear with me :), the name of the function had a typo. It should be `dismissController () {}`.
- **Use Constants:** Instead of hardcoding values like "Done" or cell name, use constants or enums to make the code more maintainable.
- **Consistent Naming:** Maintain consistency in naming convention chosen (Camel case naming) throughout the code. Specifically, `@IBAction func closeshowDetails(){}` , the name doesn't follow the naming convention as the remainder of the application; the first letter of the second word has to be capitalized.
- **Error Handling:** Implement proper error handling for data fetching in `fetchData()` Currently, there is no error handling in the code.

- **Bugs:**

- In fetchData(), specifically here:

```
let url = URL(string: "testreq")!
```

you force unwrap a URL which might cause a crash (and in this case it will as the text given to the URL is not an actual url). I would recommend using a guard statement as follows:

```
guard let url = URL(string: "www.actualurl.com/jsonData") else {
```

```
//Code to handle no returned data case
```

```
return
```

```
}
```

Also, the URL parameter label was missing.

- In this function:

```
func collectionView(_ collectionView: UICollectionView, cellForItemAt  
indexPath: IndexPath) -> UICollectionViewCell {
```

```
return UICollectionViewCell()
```

```
}
```

An empty cell is returned, i would recommend using the data array to populate the collection view cells.

- **Computed variable:** instead of using isIphone() function, I suggest using a computed value; this way it is more efficient as a computed value computes it once and it is faster to retrieve it in subsequent references.

- **Avoid strong references when possible:** In `fetchData()`, ammend this task:

```
let task = URLSession.shared.dataTask(with: url) { [weak self](data, response,  
error) in
```

```
    self.dataArray = data as? [Any]
```

```
    self.collectionView.reloadData()
```

```
}
```

Before, it had a strong reference which could cause performance issues. This was resolved by adding the “weak” keyword in the enclosure.

THE END, please feel free to reach out and ask any questions or correct anything i’ve been mistaken at, have a nice day! 😊😊😊