

Productivity

stackoverflow

AboutProductsFor Teams

Search...

Log inSign up

Home

Questions

Tags

Users

Companies

LABS

Discussions

COLLECTIVES

TEAMS

Explore all Collectives

Explore Teams

Create a free Team

150

Warning: If you get in the habit of using `enumerated()` with `ForEach`, you may one day end up with `EXC_BAD_INSTRUCTION` or `Fatal error: Index out of bounds` exceptions. This is because not all collections have 0-based indexes.

A better default is to use `zip` instead:

```
ForEach(Array(zip(items.indices, items)), id: \.0) { index, item in
    // index and item are both safe to use here
}
```

The folks over at [Point-Free](#) mentioned that it's not safe to rely on `enumerated()` with `ForEach` in production since not all collections are zero-index based:

This is technically not the most correct way to do this. It would be more correct, and more verbose, to zip the `todos` array with its indices collection. In this case we are safe because we are dealing with a simple 0-based index array, but if we were doing this in production we should probably `zip`-based approach.

Apple's documentation for the enumerated function mentions this as well:

```
/// // Prints "4: 't'"
///
/// When you enumerate a collection, the integer part of each pair is a count
/// for the enumeration, but is not necessarily the index of the paired value.
/// These counters can be used as indices only in instances of zero-based,
/// integer-indexed collections, such as 'Array' and 'ContiguousArray'. For
/// other collections the counters may be out of range or of the wrong type
/// to use as an index. To iterate over the elements of a collection with indices,
/// use the 'zip(_:_:)' function.
///
/// This example iterates over the indices and elements of a set, building
/// list consisting of indices of names with five or fewer letters.
///
/// let names: Set = ["Sofia", "Camilla", "Martina", "Mateo", "Nicolás"]
/// var shorterIndices: [Set<String>.Index] = []
/// for (i, name) in zip(names.indices, names) {
```

view in SwiftUI

0

How we can force `ForEach` use `UUID` instead of `(id: \.self)`?

See more linked questions

Related

6

Issue with `ForEach` in SwiftUI

1

`ForEach` in SwiftUI

1

SwiftUI multiple foreach

1

How to use `Foreach` on view in SwiftUI?

12

How to iterate/foreach array in SwiftUI

2

How do you use `enumerated` in a list of structs in `ForEach` SwiftUI?

2

Using `enumerated` with `ForEach` in SwiftUI

7

`.enumerated()` method not working on array

0

`ForEach` not working as expected in SwiftUI

2

Can I iterate through an array with `forEach`, but not with `List`

Hot Network Questions

For community property that is to be divided equally during divorce, can I pay my spouse the equal amount instead of selling my house?

How can I find where a short has occurred inside of a PCB?

A question about a possible electrical wiring terminal/component

Could a VTOL aircraft use a helipad for landing/taking off?