

```
//  
// ContentView.swift  
// MyTest  
//  
// Created by jazz335 on 2021/02/19.  
//
```

```
import SwiftUI
```

```
struct Item: Identifiable, Hashable {  
    var id = UUID()  
    var title: String  
    var deadline: String  
  
    func hash(into hasher: inout Hasher) {  
        id.hash(into: &hasher)  
    }  
}
```

```
struct ContentView: View {
```

```
    @State var items: [Item] = [  
        Item(title: "task1", deadline: "2021/2/20"),  
        Item(title: "task2", deadline: "2021/2/21"),  
        Item(title: "task3", deadline: "2021/2/22")  
    ]
```

ここで指定した日付を次ページのDatePickerに渡して
変更した日付がListにセットされるように

```
    var body: some View {  
        NavigationView {  
  
            List {  
                ForEach(items.indices, id: \.self) { index in  
                    NavigationLink(destination: ItemDetailView(item: $items[index])) {  
                        ItemRow(item: items[index])  
                    }  
                }  
            }  
            .navigationTitle("ToDoリスト")  
            .navigationBarTitleDisplayMode(.inline)  
            .toolbar {  
                ToolbarItem(placement: .bottomBar) {  
                    Button(action: {}) {  
                        Label("", systemImage: "plus.circle")  
                    }  
                }  
            }  
        }  
    }  
}
```

```

struct ItemRow: View {
    var item: Item

    var body: some View {
        VStack(alignment: .leading, spacing: 4) {
            Text(item.title)
            Text(item.deadline)
        }
    }
}

```

```

struct ItemDetailView: View {
    @Binding var item: Item
    var body: some View {
        VStack {
            TextField("タスクを入力してください", text: $item.title)
                .textFieldStyle(RoundedBorderTextFieldStyle()) //
                .keyboardType(.default)
                .padding()

```

```

                DatePicker(selection: .constant(Date()), displayedComponents: .date, label:
{ Text("期限") })
                    .environment(\.locale, Locale(identifier: "ja_jp"))
                    .padding()

```

```

            }
        }
    }
}

```

```

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}

```