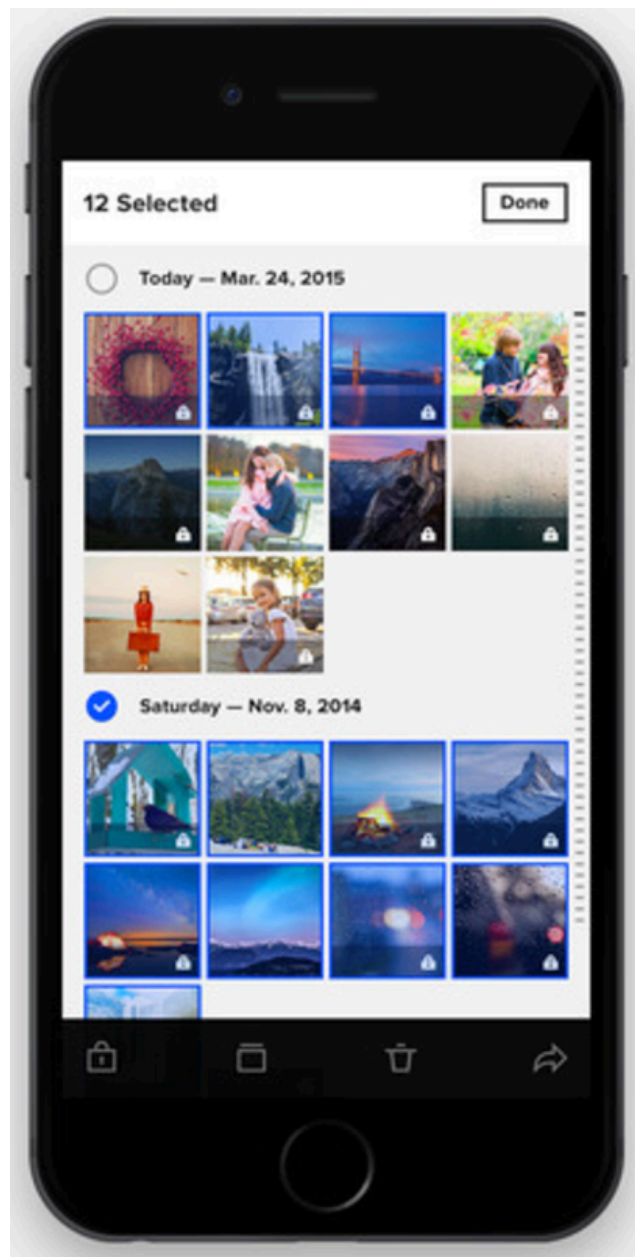


# UIImagePickerController



# UIImagePickerController

```
class ViewController: UIViewController, UIImagePickerControllerDelegate,
    UINavigationControllerDelegate {
    @IBOutlet weak var imageView: UIImageView! //UIImagePickerController 를 사용하려면
                                                UINavigationControllerDelegate
                                                Protocol을 적용해야한다.

    let picker = UIImagePickerController() //UIImagePickerController 생성

    override func viewDidLoad() {
        super.viewDidLoad()
        picker.delegate = self //delegate 지정
        picker.modalPresentationStyle = .popover //modalPresentationStyle 지정
    }

    •
    •
    •
    •
}
```

Application requires iPhone enviro...	Boolean	YES
Privacy - Photo Library Usage Des...	String	포토라이브러리 접근을 허용하시겠습니까?
Privacy - Camera Usage Description	String	카메라 사용을 허용하시겠습니까?
Launch screen interface file b...	String	LaunchScreen

info.plist

# UIImagePickerController

```
@IBAction func photofromLibrary(_ sender: Any) {  
    let alert = UIAlertController(title: "사진촬영", message: "사진의 소스를 선택하세요.",  
        preferredStyle: .actionSheet)  
    let action1 = UIAlertAction(title: "앨범에서 가져오기", style: .default,  
        handler:action1Handler)  
    alert.addAction(action1)  
    let action2 = UIAlertAction(title: "촬영하기", style: .default,  
        handler:action2Handler)  
    alert.addAction(action2)  
    let action3 = UIAlertAction(title: "취소", style: .cancel, handler:nil)  
    alert.addAction(action3)  
    present(alert, animated: true, completion: nil)  
}
```

# UIImagePickerController

Image의 소스를 지정

```
func action1Handler(action:UIAlertAction) {  
    picker.sourceType = .photoLibrary    //이미지 소스 지정  
    present(picker, animated: true, completion: nil)  
}
```

```
func action2Handler(action:UIAlertAction) {  
    picker.sourceType = .camera           //이미지 소스 지정  
    present(picker, animated: true, completion: nil)  
}
```

# UIImagePickerController

선택한 Image를 imageView에 표시

```
func imagePickerController(_ picker: UIImagePickerController,
    didFinishPickingMediaWithInfo info: [String : Any]) {
    let chosenImage = info[.originalImage] as! UIImage //info dictionary 에서 이미지가져오기
    imageView.contentMode = .scaleAspectFit
    imageView.image = chosenImage
    dismiss(animated:true, completion: nil)
}

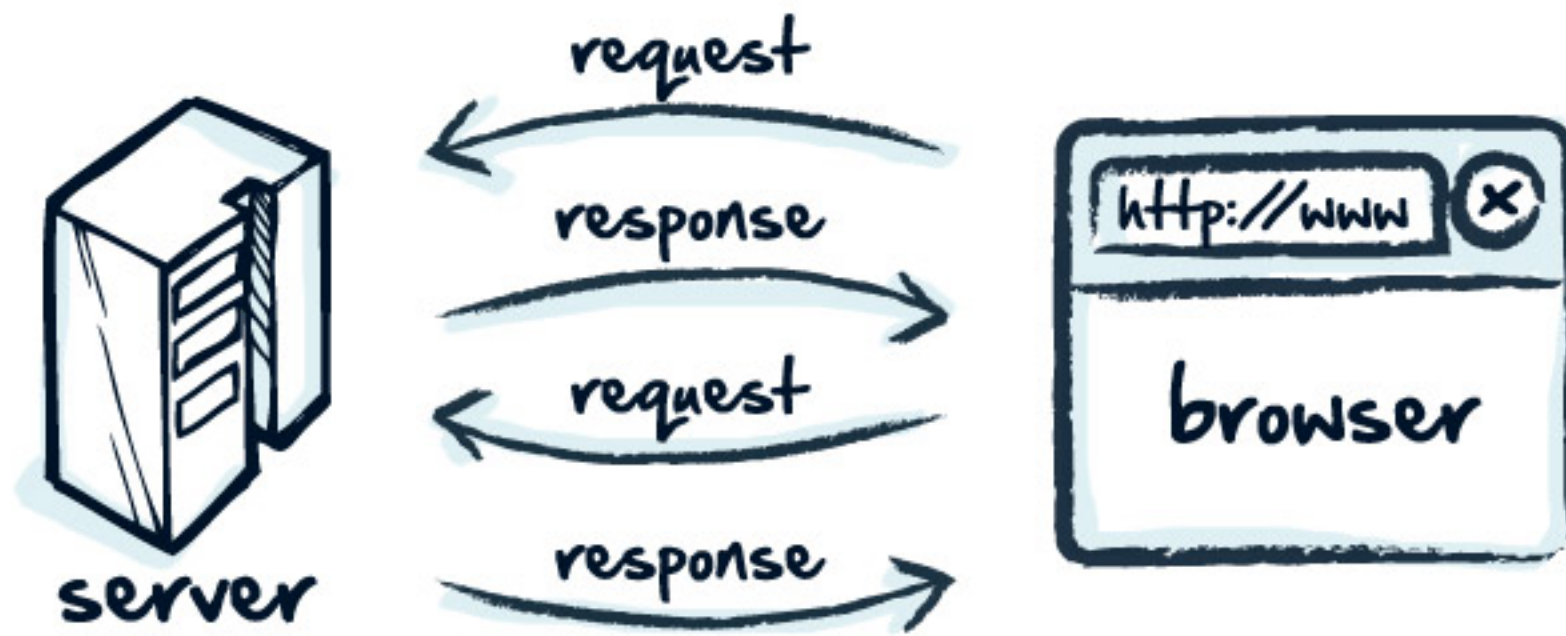
func imagePickerControllerDidCancel(_ picker: UIImagePickerController) {
    dismiss(animated:true, completion: nil)
}
```

# UIImagePickerController

Image를 도큐먼트 폴더에 저장

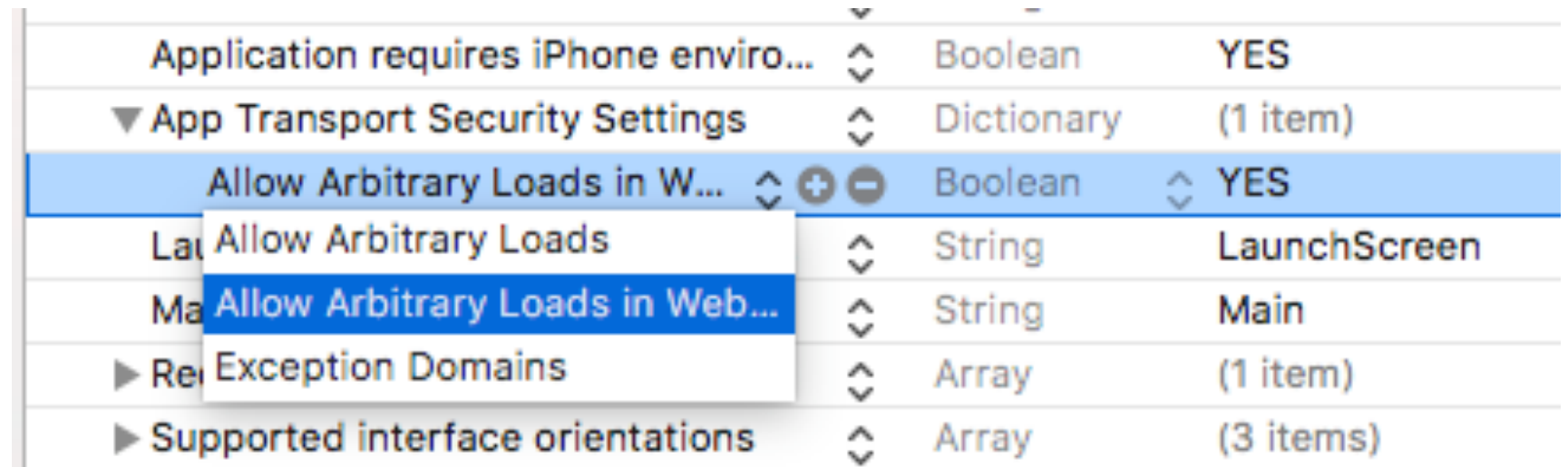
```
func getDocumentsDirectory() -> URL {  
    let paths = FileManager.default.urls(for:.documentDirectory, in:.userDomainMask)  
    let documentsDirectory = paths[0] //Document 폴더경로  
    return documentsDirectory  
}  
  
@IBAction func actSaveImage(_ sender: Any) {  
    if let data = UIImageJPEGRepresentation(imageView.image!, 0.8) {  
        let filename = getDocumentsDirectory().appendingPathComponent("sample.png")  
        try? data.write(to: filename)  
    }  
}
```

# WKWebView



# WKWebView

```
override func viewDidLoad() {  
    super.viewDidLoad()  
    if let url = URL(string: "http://www.daum.net"){  
        let request = URLRequest(url: url)  
        webView.load(request)  
    }  
}
```



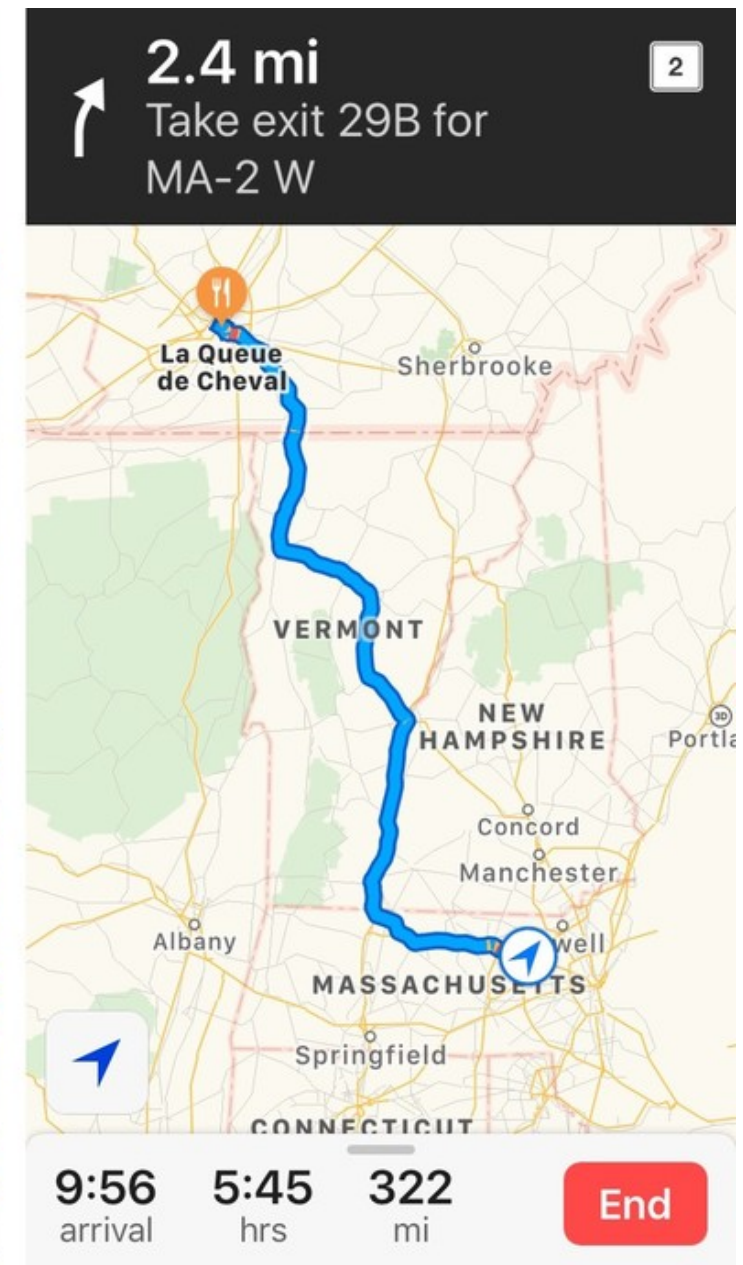
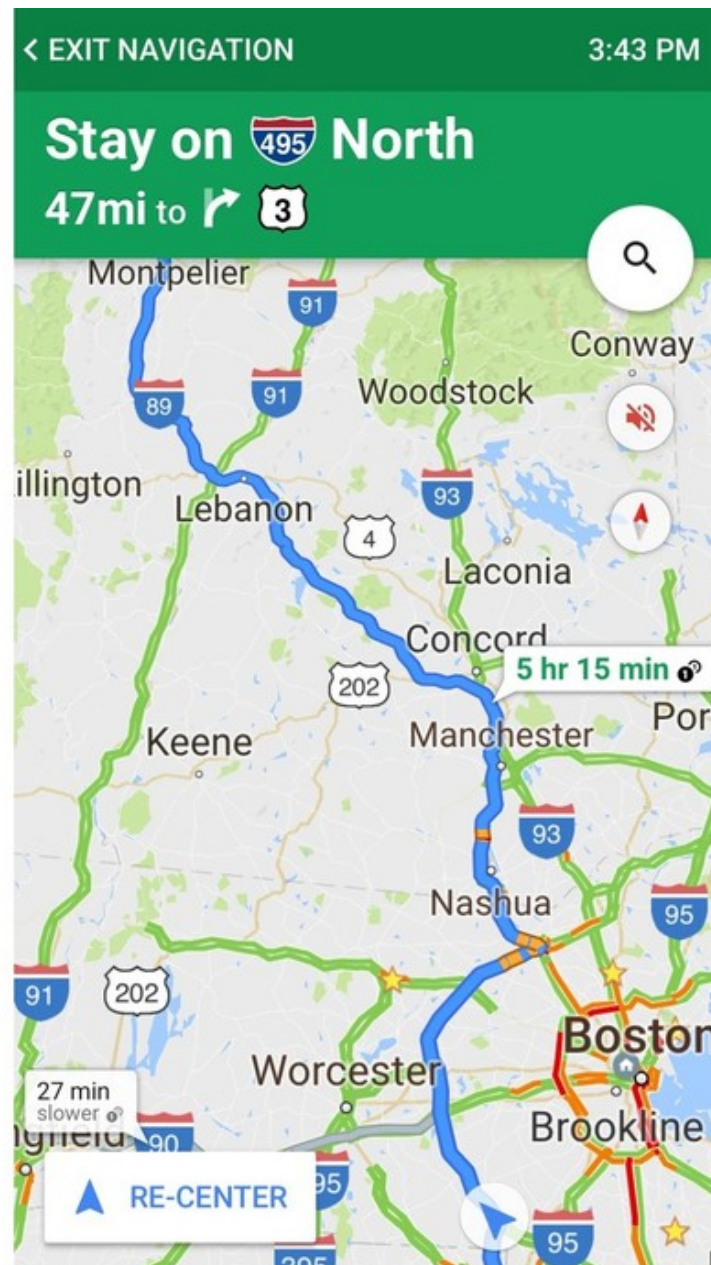
The screenshot shows the 'App Transport Security Settings' section in Xcode. A dropdown menu is open for the 'Allow Arbitrary Loads in WebViews' setting, showing options: 'Allow Arbitrary Loads' (selected), 'Allow Arbitrary Loads in Web...', and 'Exception Domains'. The main table lists the following settings:

Application requires iPhone environment	Boolean	YES
App Transport Security Settings	Dictionary	(1 item)
Allow Arbitrary Loads in WebViews	Boolean	YES
Launch Screen	String	LaunchScreen
Main Screen	String	Main
Exception Domains	Array	(1 item)
Supported interface orientations	Array	(3 items)

Application Transport Security 설정



# MapView



# MapView

```
import MapKit
```

```
class ViewController: UIViewController {
```

```
    @IBOutlet weak var mapView: MKMapView!
```

```
    override func viewDidLoad() {
```

```
        super.viewDidLoad()
```

```
        mapView.mapType = .hybrid
```

```
        let location = CLLocationCoordinate2D(latitude: 37.46329969457437,  
longitude:126.90589148241693)
```

```
        let span = MKCoordinateSpan(latitudeDelta: 0.005, longitudeDelta: 0.005)
```

```
        let region = MKCoordinateRegion(center: location, span: span)
```

```
        mapView.setRegion(region, animated: true)
```

```
    }
```

```
}
```

## ▼ Linked Frameworks and Libraries

Name	Status
 MapKit.framework	Required ⬆
+ -	

MapKit framework 추가

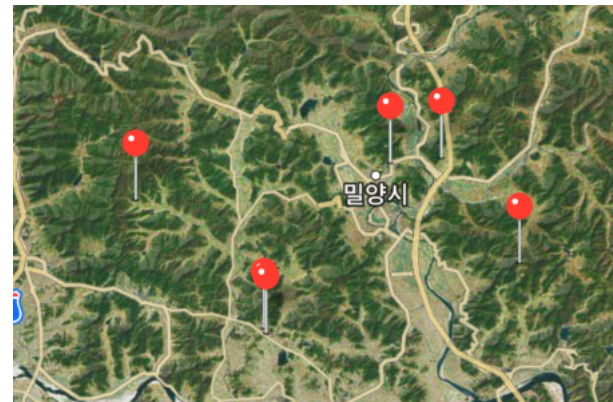
# Point Annotation

```
let myLocation = CLLocationCoordinate2DMake(35.242888, 128.697237)
```

```
let mark = MKPointAnnotation()
```

```
mark.coordinate = myLocation
```

```
mapView.addAnnotation(mark)
```



# Custom Annotation

```
import MapKit

class MyAnnotation: NSObject, MKAnnotation {
    let title: String?
    let subtitle: String?
    let mapURL: String
    let coordinate: CLLocationCoordinate2D
    let thumbNail:String
    init(title: String, subtitle: String, mapURL: String,
         coordinate: CLLocationCoordinate2D, thumbNail:String) {
        self.title = title
        self.subtitle = subtitle
        self.mapURL = mapURL
        self.coordinate = coordinate
        self.thumbNail = thumbNail
        super.init()
    }
}
```

# Custom Annotation

```
let mark = MyAnnotation(title: "남부여성발전센터", subtitle: "금천구청",  
                        mapURL: "https://nambu.seoulwomanup.or.kr",  
                        coordinate: myLocation,  
                        thumbNail: "logo")
```



# Custom Annotation

```
func mapView(_ mapView:MKMapView, viewFor annotation:MKAnnotation)->MKAnnotationView?{
    let reuseId = "pin"
    let aAnnotation = annotation as! MyAnnotation
    var pinView = mapView.dequeueReusableAnnotationView(withIdentifier: reuseId)
    if pinView == nil {
        pinView = MKPinAnnotationView(annotation: annotation, reuseIdentifier: reuseId)
        pinView!.canShowCallout = true
        let button = UIButton(type:.detailDisclosure)
        pinView!.rightCalloutAccessoryView = button
        let image = UIImage(named: aAnnotation.thumbNail)
        let imageView = UIImageView(image: image)
        imageView.frame.size = CGSize(width: 60, height: 40)
        pinView!.leftCalloutAccessoryView = imageView
    } else {
        pinView?.annotation = annotation
    }
    return pinView
}
```

