

# «MySettings»

Fabrice.Kordon@lip6.fr



# Goal of the exercise

2

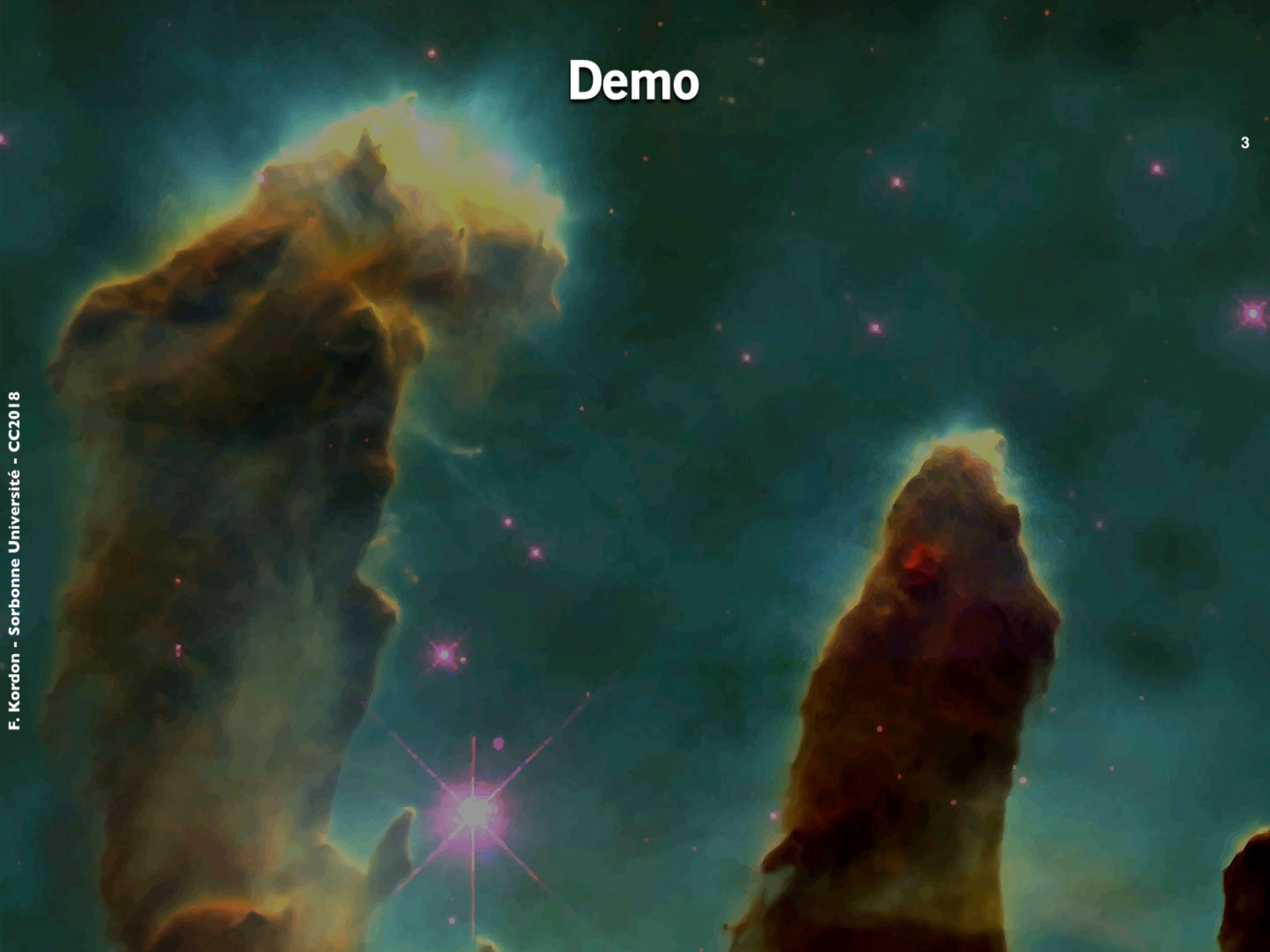


## Simple

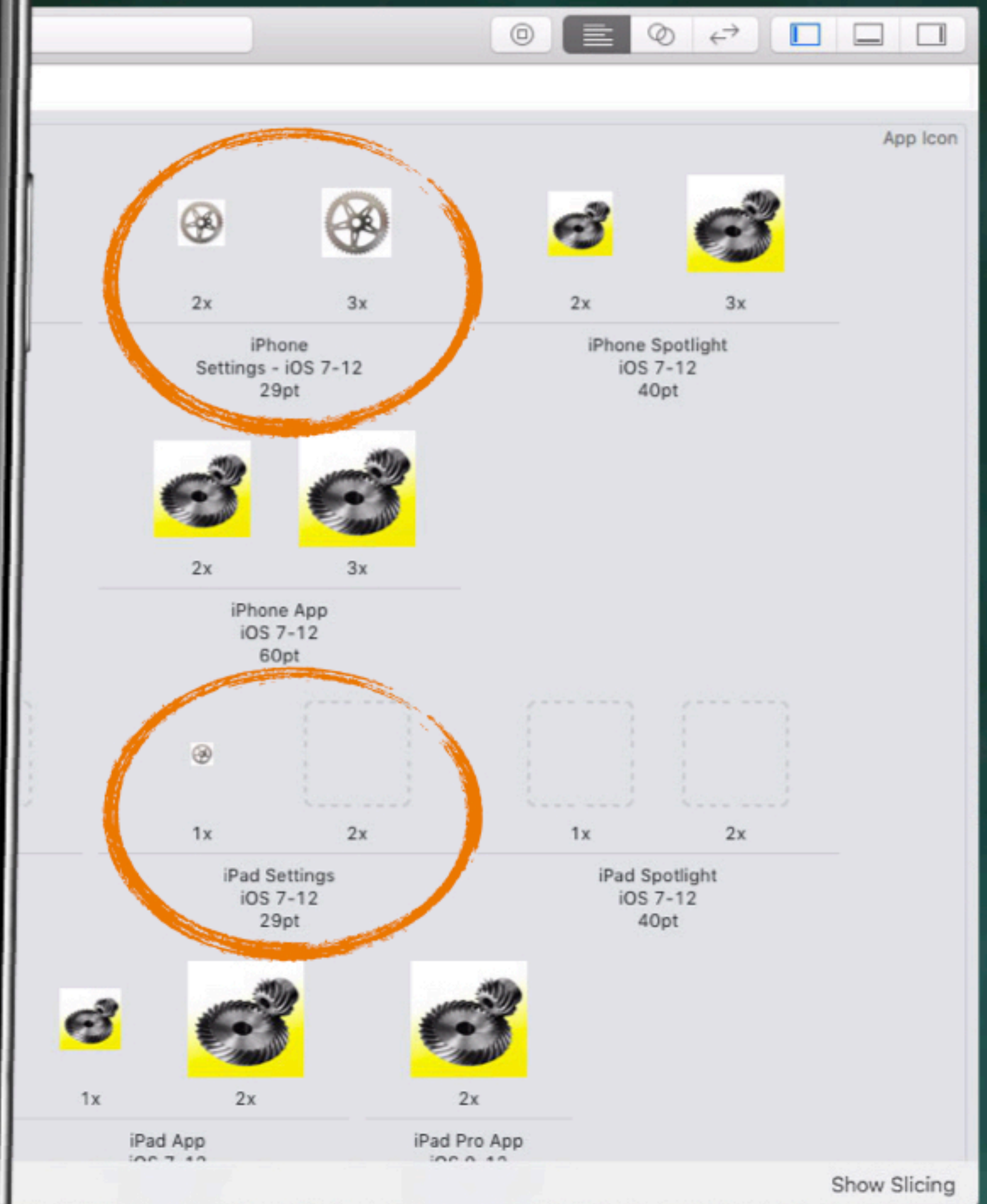
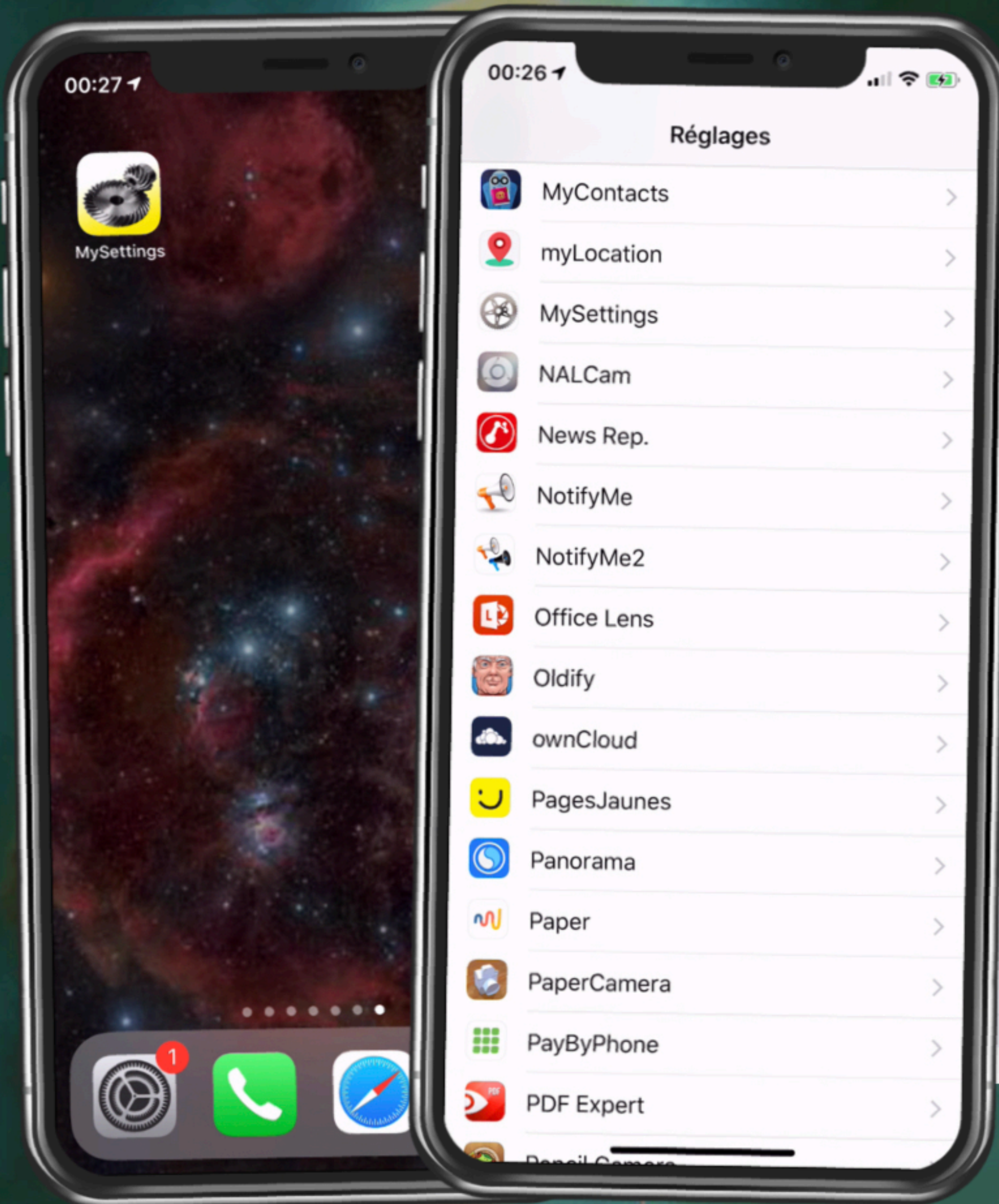
- Build a Bundle setting for preferences
- Read settings for display
- Set up default preferences
- Invoke the device's setting app for the Application



# Demo

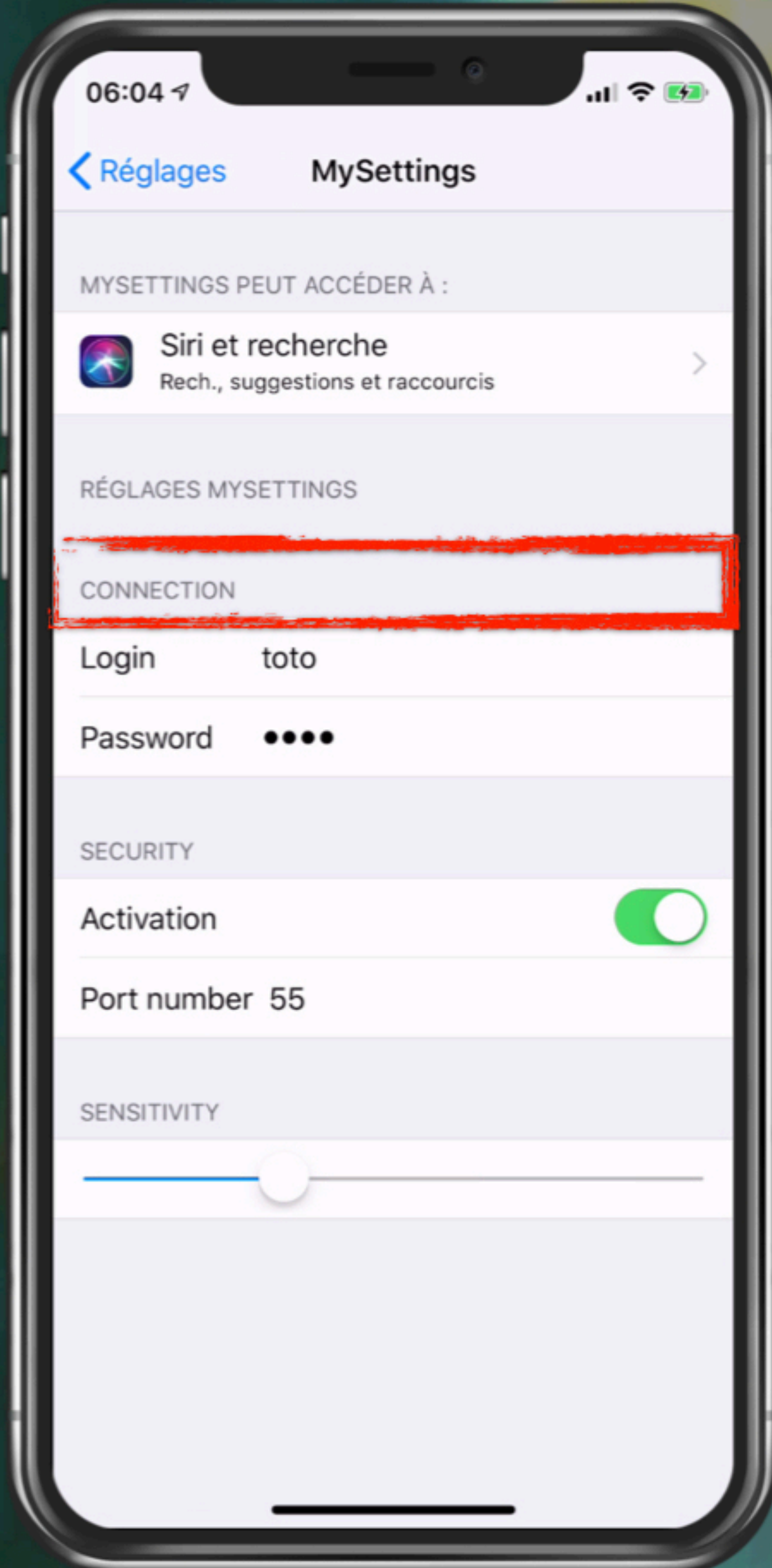


# About the setting icon



# Root.plist

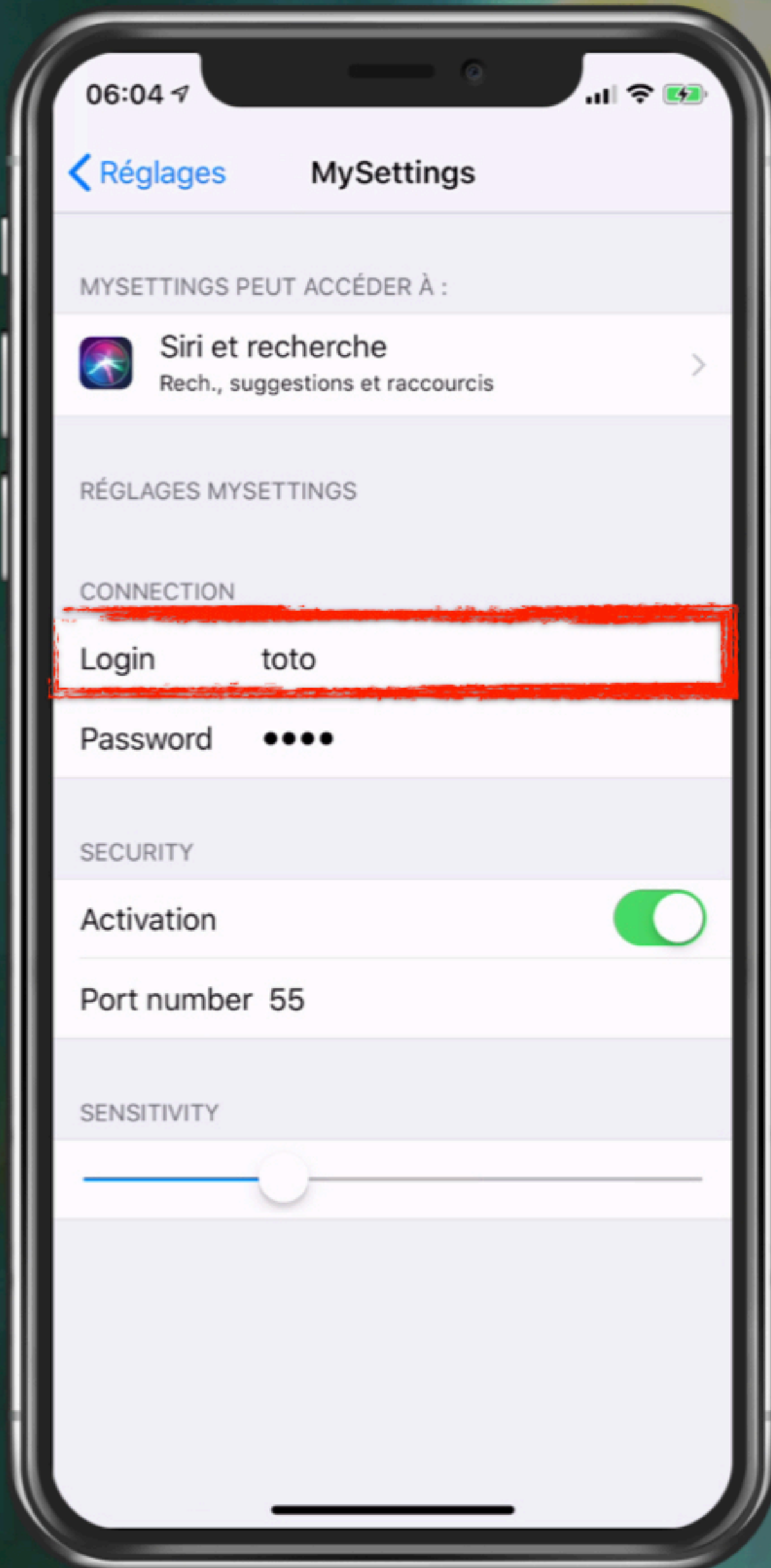
5



```
<key>PreferenceSpecifiers</key>
<array>
  <dict>
    <key>Title</key>
    <string>Connection</string>
    <key>Type</key>
    <string>PSGroupSpecifier</string>
  </dict>
```

# Root.plist

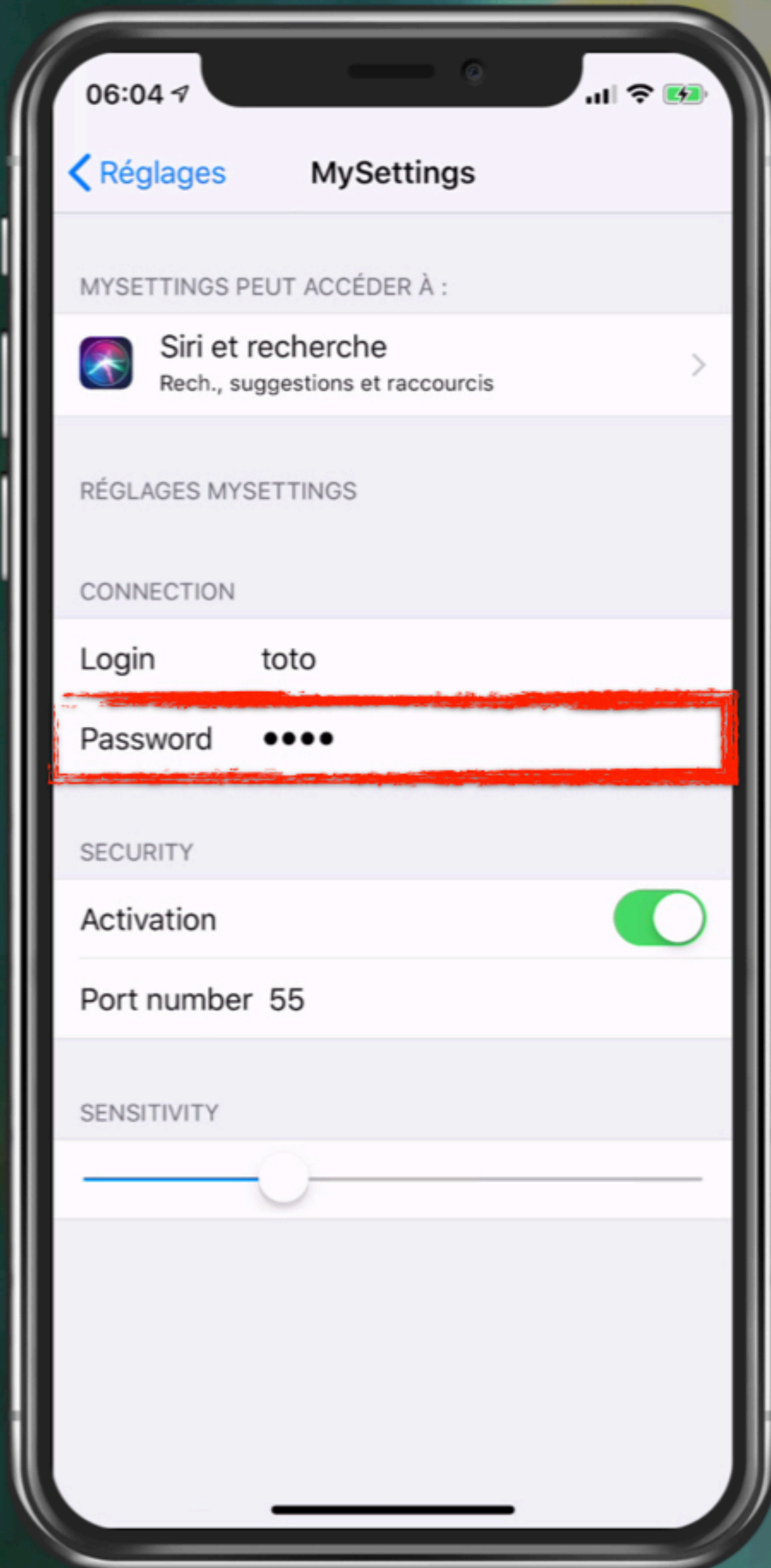
5



```
<dict>
  <key>AutocapitalizationType</key>
  <string>None</string>
  <key>AutocorrectionType</key>
  <string>No</string>
  <key>DefaultValue</key>
  <string></string>
  <key>IsSecure</key>
  <false/>
  <key>Key</key>
  <string>login_prefs</string>
  <key>KeyboardType</key>
  <string>Alphabet</string>
  <key>Title</key>
  <string>Login</string>
  <key>Type</key>
  <string>PSTextFieldSpecifier</string>
</dict>
```

# Root.plist

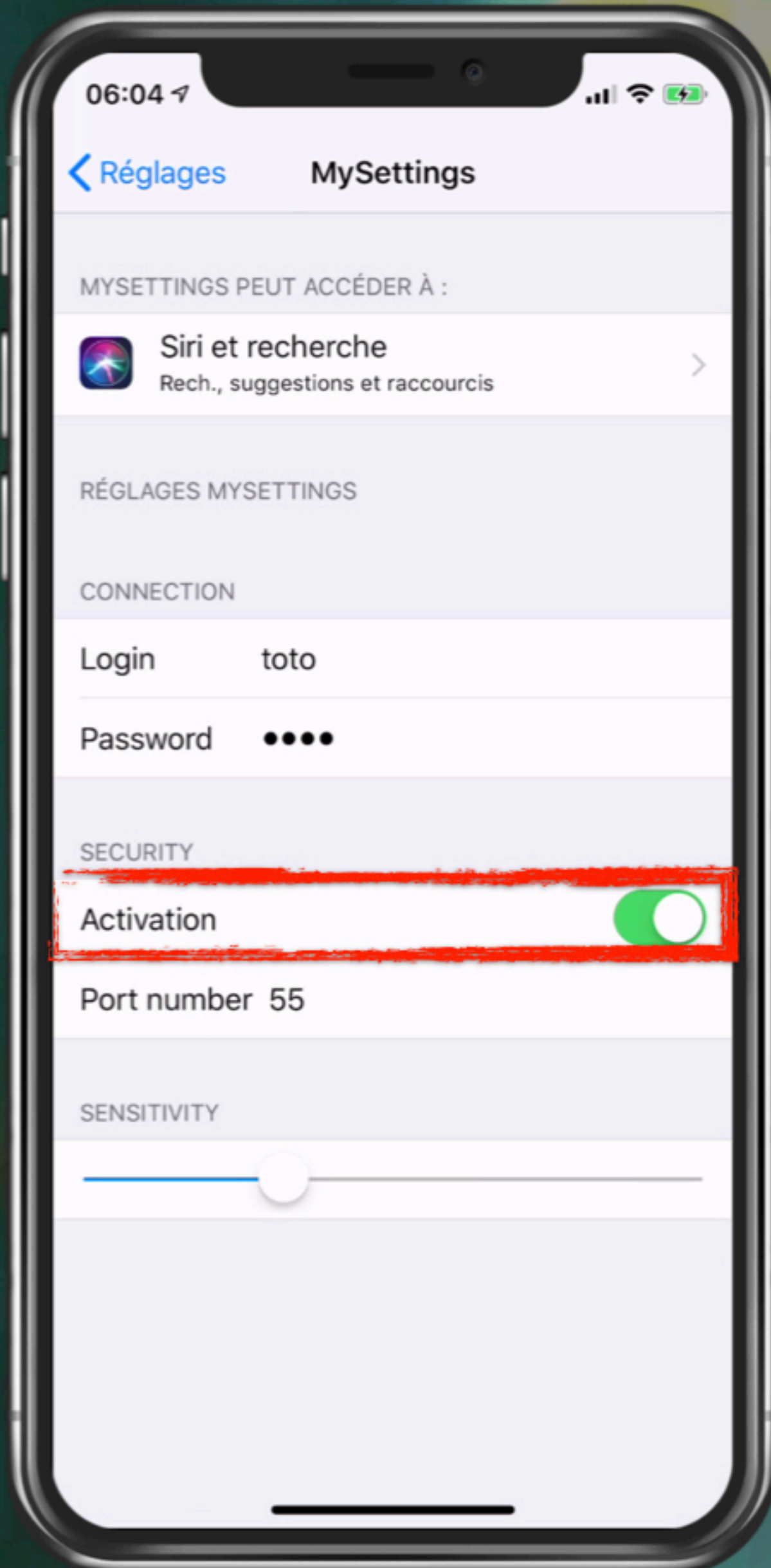
5



```
<dict>
  <key>AutocapitalizationType</key>
  <string>None</string>
  <key>AutocorrectionType</key>
  <string>No</string>
  <key>DefaultValue</key>
  <string></string>
  <key>IsSecure</key>
  <true/>
  <key>Key</key>
  <string>pass_prefs</string>
  <key>KeyboardType</key>
  <string>Alphabet</string>
  <key>Title</key>
  <string>Password</string>
  <key>Type</key>
  <string>PSTextFieldSpecifier</string>
</dict>
```

# Root.plist

5

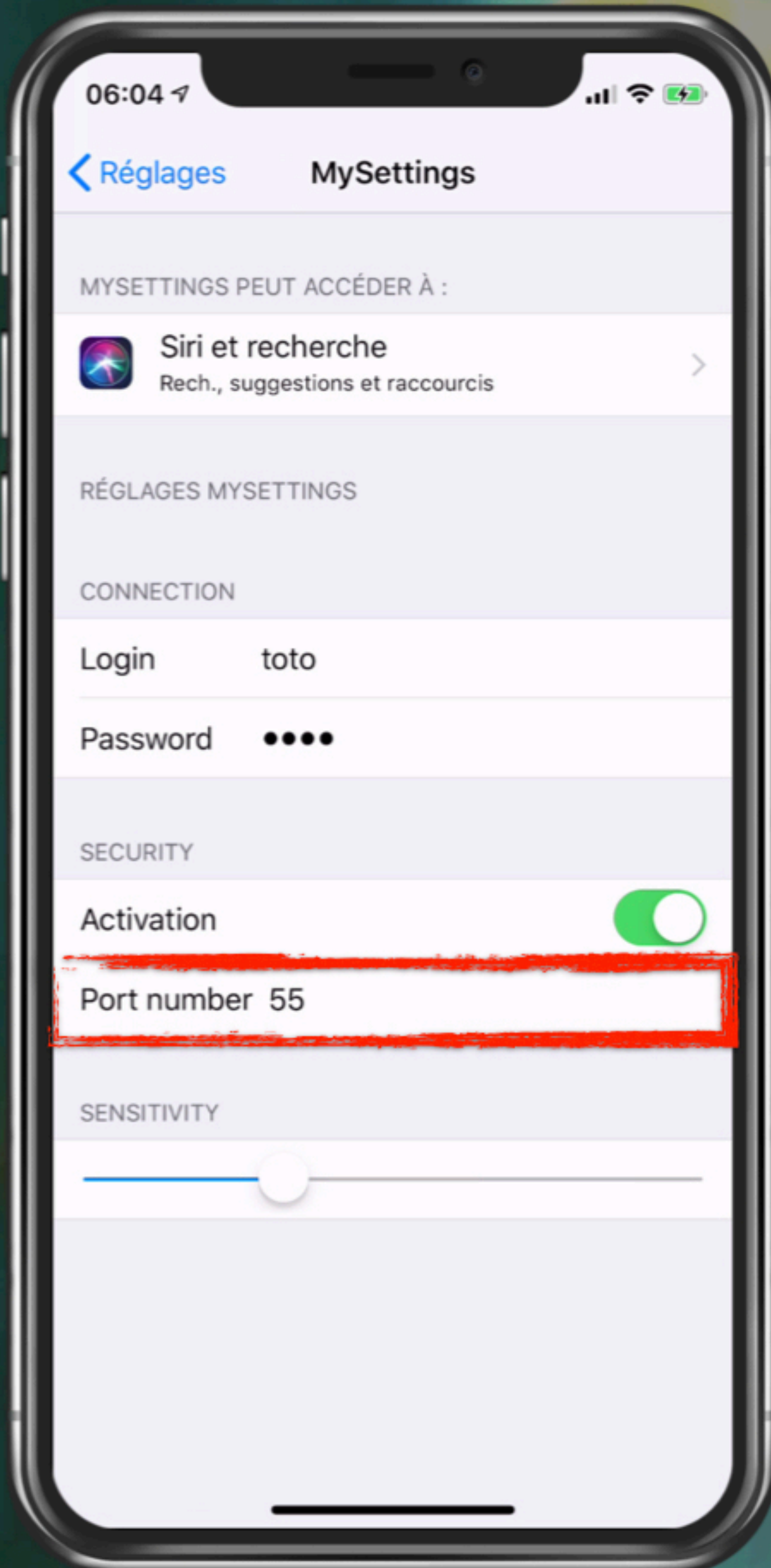


```
<dict>
  <key>DefaultValue</key>
  <false/>
  <key>Key</key>
  <string>security_prefs</string>
  <key>Title</key>
  <string>Activation</string>
  <key>Type</key>
  <string>PSToggleSwitchSpecifier</string>
</dict>
```



# Root.plist

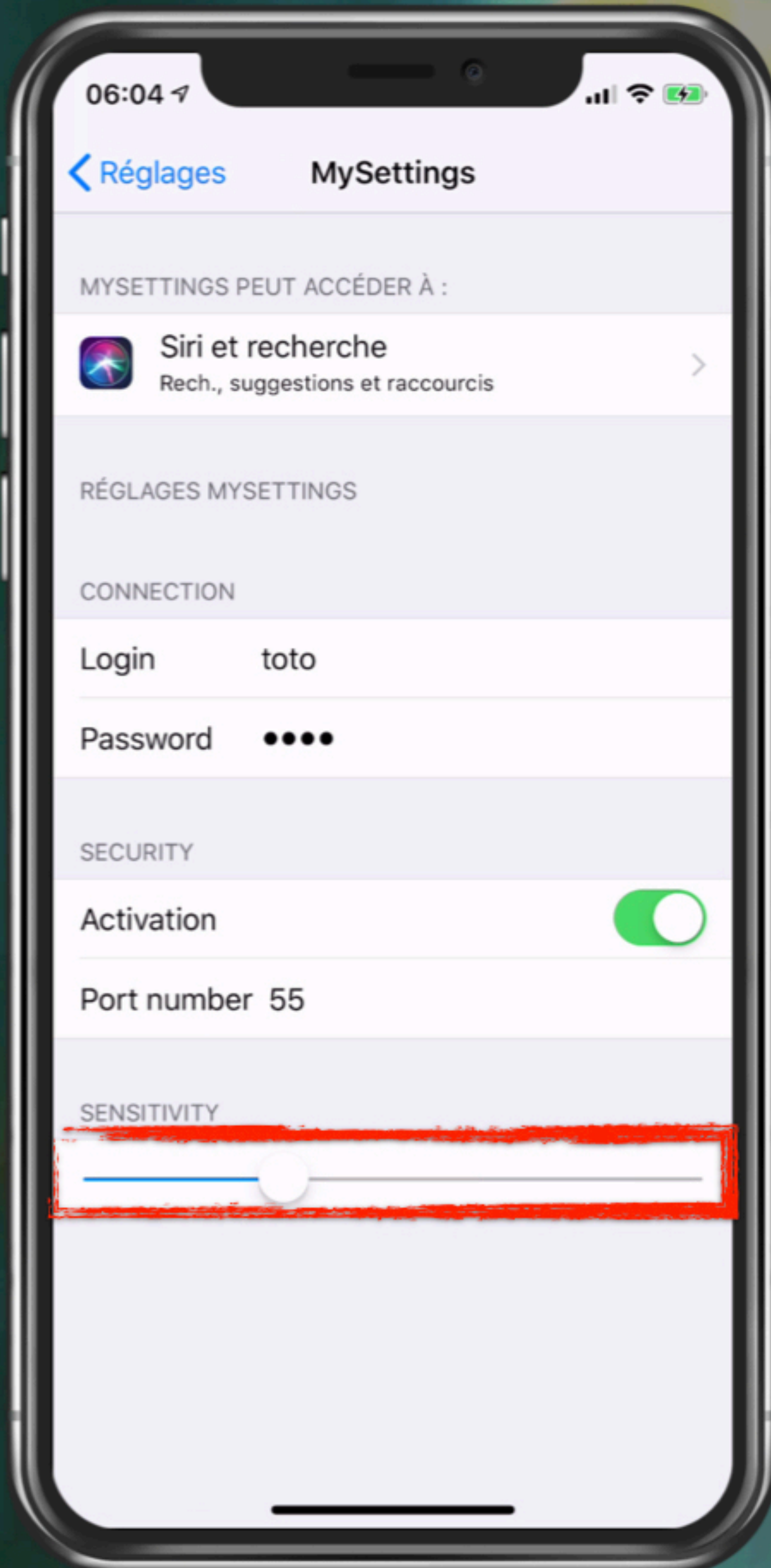
5



```
<dict>
  <key>AutocapitalizationType</key>
  <string>None</string>
  <key>AutocorrectionType</key>
  <string>No</string>
  <key>DefaultValue</key>
  <string></string>
  <key>IsSecure</key>
  <false/>
  <key>Key</key>
  <string>port_prefs</string>
  <key>KeyboardType</key>
  <string>NumberPad</string>
  <key>Title</key>
  <string>Port number</string>
  <key>Type</key>
  <string>PSTextFieldSpecifier</string>
</dict>
```

# Root.plist

5



```
<dict>
  <key>DefaultValue</key>
  <real>0.5</real>
  <key>Key</key>
  <string>sensitivity_prefs</string>
  <key>MaximumValue</key>
  <integer>1</integer>
  <key>MaximumValueImage</key>
  <string></string>
  <key>MinimumValue</key>
  <integer>0</integer>
  <key>MinimumValueImage</key>
  <string></string>
  <key>Type</key>
  <string>PSSliderSpecifier</string>
</dict>
```

# ViewController



**Sake of simplicity...**

Code located in a  
ViewController

# ViewController

```
import UIKit

class ViewController: UIViewController {
    private let pLogin = UILabel()
    private let pPasswd = UILabel()
    private let pAct = UILabel()
    private let pPort = UILabel()
    private let pSlider = UILabel()

    let b1 = UIButton(type: .system)
    let b2 = UIButton(type: .system)
    let b3 = UIButton(type: .system)

    override var preferredStatusBarStyle: UIStatusBarStyle {
        return .lightContent
    }
}
```

# ViewController

```
override func viewDidLoad() {
    super.viewDidLoad()
    let s = UIScreen.main.bounds
    let v = UIView()
    let bg = UIImageView(image: UIImage(named: "background"))
    bg.center = CGPoint(x: s.height / 2, y: s.height / 2)
    self.view = v
    v.addSubview(bg)
    pLogin.frame = CGRect(x: 20, y: 50, width: s.width - 40, height: 20)
    pLogin.textColor = UIColor.white
    pLogin.font = UIFont.boldSystemFont(ofSize: 16.0)
    v.addSubview(pLogin)
    pPasswd.frame = CGRect(x: 20, y: 80, width: s.width - 40, height: 20)
    pPasswd.textColor = UIColor.white
    pPasswd.font = UIFont.boldSystemFont(ofSize: 16.0)
    v.addSubview(pPasswd)
    pAct.frame = CGRect(x: 20, y: 110, width: s.width - 40, height: 20)
    pAct.textColor = UIColor.white
    pAct.font = UIFont.boldSystemFont(ofSize: 16.0)
    v.addSubview(pAct)
    pPort.frame = CGRect(x: 20, y: 140, width: s.width - 40, height: 20)
    pPort.textColor = UIColor.white
    pPort.font = UIFont.boldSystemFont(ofSize: 16.0)
    v.addSubview(pPort)
    pSlider.frame = CGRect(x: 20, y: 170, width: s.width - 40, height: 20)
    pSlider.textColor = UIColor.white
    pSlider.font = UIFont.boldSystemFont(ofSize: 16.0)
    v.addSubview(pSlider)
    b1.frame = CGRect(x: 20, y: 200, width: s.width - 40, height: 20)
    b1.setTitle("Fetch device settings", for: .normal)
    b1.tintColor = UIColor.yellow
    b1.addTarget(self, action: #selector(readSettings), for: .touchDown)
    v.addSubview(b1)
    b2.frame = CGRect(x: 20, y: 230, width: s.width - 40, height: 20)
    b2.setTitle("Set default device settings", for: .normal)
    b2.tintColor = UIColor.yellow
    b2.addTarget(self, action: #selector(setDefaults), for: .touchDown)
    v.addSubview(b2)
    b3.frame = CGRect(x: 20, y: 260, width: s.width - 40, height: 20)
    b3.setTitle("Invoke device settings", for: .normal)
    b3.tintColor = UIColor.yellow
    b3.addTarget(self, action: #selector(invokerSettings), for: .touchDown)
    v.addSubview(b3)
    self.display("", pass: "", security: "", port: "", sensitivity: "")
}
```

# ViewController

```
func display(_ login : String?, pass : String?,
            security : String?, port : String?,
            sensitivity : String?) {
    var s = ""
    if (login != nil) {
        s = login!
    }
    pLogin.text = "Login\t\t\t: \(s)"
    if (pass != nil) {
        s = pass!
    } else {
        s = ""
    }
    pPasswd.text = "Password\t\t: \(s)"
    if (security != nil) {
        s = security!
    } else {
        s = ""
    }
    pAct.text = "Security\t\t: \(s)"
    if (port != nil) {
        s = port!
    } else {
        s = ""
    }
    pPort.text = "Port\t\t\t: \(s)"
    if (sensitivity != nil) {
        s = sensitivity!
    } else {
        s = ""
    }
    pSlider.text = "Sensitivity\t\t: \(s)"
}
```

# ViewController

```
@objc func readSettings () {  
    let prefs = UserDefaults.standard  
    self.display(prefs.string(forKey: "login_prefs"),  
                pass: prefs.string(forKey: "pass_prefs"),  
                security: prefs.string(forKey: "security_prefs"),  
                port: prefs.string(forKey: "port_prefs"),  
                sensitivity: prefs.string(forKey: "sensitivity_prefs"))  
}
```

```
@objc func setDefaults () {  
    let prefs = UserDefaults.standard  
    let myDefaults = ["login_prefs" : "MyLogin",  
                     "pass_prefs" : "MyPass",  
                     "security_prefs" : "true",  
                     "port_prefs" : "22",  
                     "sensitivity_prefs" : "0.75"]  
    prefs.register(defaults: myDefaults)  
    prefs.synchronize()  
}
```

```
@objc func invokeSettings () {  
    let appl = UIApplication.shared  
    let url = URL(string: UIApplication.openSettingsURLString)  
    appl.open(url!, options: [:], completionHandler: nil)  
}
```

```
}
```

# As a conclusion...



**Super easy!!!**

• No need to ignore...

