IOS SmartWiFi Connect library

For the SDK to work, you need to add Capability - Access WiFi Information and Hotspot Configuration to the application target.

The SDK uses a certificate to verify the server; in the application you need to enable/add Keychain Sharing capability in Xcode - com.apple.networkextensionsharing to put the certificate there.

Adding an SDK to a project:

```
CocoaPods

pod 'smartwifi_ios_sdk', :git =>
'https://github.com/VitaliyPedan/smartwifi_ios_sdk.git'
```

Connection к Wi-Fi:

1. Initializing the WiFi session object, you need to assign a delegate to inform the connection status:

```
let wifiSession: SWFWiFiSession

// Create new session instance

     wifiSession = SWFWiFiSession(teamId:<TeamIdentifier>, delegate:
<delegate>)
```

2. Create a Wi-Fi session instance with SWFSessionObject: where:

- user_id - a unique user identifier by which you can recognize him

- apiKey - Access key к API SmartWi-FI (Issued separately)

- channelId - Channel identifier in the SmartWi-FI system (Issued separately)

- projectId - Project identifier in the SmartWi-FI system (Issued separately)

- apiDomain - Domain name of the API server (https://...) (Issued separately)

When creating a session, configurations are automatically requested from the server for connection. If the response is successful, the configuration data is cached by the SDK, therefore, it is recommended to create a session when the user has Internet access, for example, when logging into the application.

```
// Create session object

let sessionObject = SWFSessionObject(

     apiKey: apiKey,

     userId: userId,

     channelId: channelId,

     projectId: projectId,

     apiDomain: apiDomain

)
```

```
// Configuration of session
      wifiSession.createSession(sessionObject: sessionObject) { [weak
self] (result) in
if case .success = result {
self?.wifiSession.startSession()
}
}
```

3. Start a Wi-Fi session (connection к Wi-Fi):

When connecting, the configuration is taken from the cache obtained and saved during the creation of the Wi-Fi instance createSession(sessionObject: step 2 of this instruction, otherwise we get a missing configuration error. The delegate will be informed about the connection result in the corresponding method.

```
// Start session if session instance present
      wifiSession.startSession()
```

4. The cancelSession method removes configurations and disconnects from the network:

```
// Disconnect and removing of network
wifiSession.cancelSession()
```

5. Connection steps:

```
public protocol SWFWiFiSessionDelegate {
      func willCreate(session: SWFWiFiSession)
      func didCreate(session: SWFWiFiSession, error: SWFError?)
      func willConnectToWiFi(session: SWFWiFiSession)
      func didConnectToWiFi(via configType: SWFConfigType?, session:
SWFWiFiSession, error: SWFError?)
      func didStopWiFi(session: SWFWiFiSession)
}
```

6. To open the full Internet through the application, you need to check whether the device is connected to a Wi-Fi network (which one is not important) and if connected, then send a get request to the address (This link will open the Internet for the user for 20 minutes):

```
https://smartwf.ru:40443/open/full/?time=1200&comment=from_mobile_app&pass=96wImExVQPmJvd46&down_bw=1gbit&up_bw=1gbit
```

IOS SmartWiFi Connect library errors

Errors that need to be displayed to the user are highlighted with a frame.

case mappingModelFailure

- This error means an error in creating a data model with data received from the server.

- Occurs when trying to decode a model with Data.

- "Error decoding model with data"

case savingDataFailure

- This error means that an error occurred while caching configurations.

- Occurs when trying to encode a model (configuration) into data for further saving by key.

- "Error encoding model in data"

case objectDoNotExist

- This error means that the session object is corrupted or unloaded from memory.

- May occur when requesting settings (configurations), as well as when adding/applying a configuration.

- "The session object is corrupted"

case wifiModuleSwitchOff

- This error means that you must turn on the Wi-Fi module before connecting.

- It occurs in cases where the user's Wi-Fi module is disabled when connecting. This error must be displayed to the user.

- "Your Wi-Fi module is turned off, open settings and turn it on"

case sessionIsNotConfigured

- This error means that the session is not configured. You must set up a session before using it.

- Occurs when trying to start/stop a session without prior configuration.

- "Session not configured. Session must be configured before use"

case configsNotSaved

- This error means that there are no cached configurations for the connection.

- It occurs in cases where the connect/disconnect method was launched, but the configurations were not successfully saved.

- "No saved configurations"

case emptyConfigs

- This error means that the list of configurations for connection is empty.

- It occurs in cases where empty configurations are received from the server. In this case, you must report the problem.

- "The resulting list of configurations is empty"

case configHasNoPriority

- This error means that the received configurations do not have priority or the priority is incorrect

- It occurs in cases where connection parameters are incorrectly received from the server. Occurs when attempting to apply a configuration.

- "Configuration priority is missing or incorrect"

case saveIdentifierFailure

- This error means that the Wi-Fi network cannot be found when connecting using the wpa2 method.

- It occurs in cases where the user is out of range of Wi-Fi or there is a problem with the router. In this case, there will be a system error and the SDK will begin connecting using the next priority connection method.

- "The device has not connected to this Wi-Fi network"

case unableToJoinNetwork

- This error means that the Wi-Fi network cannot be found.

- Occurs when the configuration is successfully applied, but it is impossible to connect to the network. In this case, there will be a system error and the SDK will begin connecting using the next priority connection method.

- It occurs in cases where the user is outside the Wi-Fi coverage area.

- "Wi-Fi network not detected nearby"

case sessionIsNotCreated

- This error means that the session was not created. You must create a session before using it.

- Occurs when trying to request/receive settings.

case saveIdentifierRequestFailure

- This error means an error from the server when requesting saveIdentifier

case fullWifiAccessRequestFailure

- This error means an error from the server when requesting fullWifiAccess

case getWiFiSettingsRequestFailure

- This error means an error from the server when requesting WiFiSettings

case notConnectedPreviously

- This error means that the Wi-Fi network they are trying to disable was not previously added.

- "Has not been connected before"

case serverError(serverError: SWFServerError)

- Server error

case unknownError

- Unknown error

IOS SmartWiFi Connect library errors when applying configuration

These errors are returned by NEHotspotConfigurationManager when calling the method:

- apply(_ configuration: NEHotspotConfiguration).

case invalid- "Wi-Fi configuration is invalid"

case invalidSSID- "The given SSID string is invalid"

case invalidWPAPassphrase- "Personal WPA/WPA2 password is invalid"

case invalidWEPPassphrase- "The given WEP password is invalid"

case invalidEAPSettings- "Incorrect EAP settings"

case invalidHS20Settings- "Incorrect Hotspot 2.0 settings."

case invalidHS20DomainName- "This Hotspot 2.0 domain name is invalid"

case userDenied- "Unable to obtain user approval to add a new configuration"

case `internal`- "An internal error has occurred"

case pending- "The calling application's previous request is pending"

case systemConfiguration- "The calling application cannot change the system configuration (MDM / carrier)"

case joinOnceNotSupported- "The JoinOnce parameter is not supported by the EAP configuration"

case alreadyAssociated- “Wi-Fi is already connected”, this error occurs when the phone is already connected to the desired SSID and tries to do it again

case applicationIsNotInForeground- "The application is not active"

case invalidSSIDPrefix- "The given SSID prefix string is invalid"