

# App Notes on Registration

03/22/2017

This document describes the different methods used to register an Ayla device with the Ayla service using methods provided in Ayla Mobile SDK for iOS. Code examples are provided in Swift 3.

There's one method to handle all different kinds of registration:

```
open func register(_ candidate: AylaRegistrationCandidate, success
successBlock: @escaping (AylaDevice) -> Swift.Void, failure
failureBlock: @escaping (Error) -> Swift.Void) -> AylaHTTPTask?
```

This method takes a candidate and a success and failure block which will receive the registered device upon success or an error in case of failure.

## Registration Methods

There are several registration methods, the registration method to be used is determined by the characteristics of the candidate. Let's review the appropriate procedure and the things to look for when registering a device of each registration type.

For all registration types you must use an `AylaRegistration` object from your session manager `registration` property or a new instance:

```
let registration = AylaRegistration(sessionManager: sessionManager)
```

## Same-LAN

When a Same-LAN enabled IoT device completes setup, the Ayla Cloud creates a registration token for it, and sends it to the device. This registration method works by fetching that registration token from the device over the LAN. Therefore, the IoT device and the phone must be connected to the same network and able to communicate over it. The registration token is used to prove local connectivity access to the IoT device. Retrieving the registration token from the device will happen without application intervention when the `registrationType` is set to `sameLan`.

## Details

1. If you have just performed W-iFi setup, you can create an `AylaRegistrationCandidate` from the `setupDevice` object, otherwise you'll need to fetch candidates:

- a. From `setupDevice` after confirming device connected to ADS:

```
let candidate = AylaRegistrationCandidate()
candidate.dsn = setupDevice.dsn
candidate.registrationType = setupDevice.registrationType
candidate.lanIp = setupDevice.lanIp
```

- b. By fetching candidate, optionally specifying dsn

```
registraton.fetchCandidate(withDSN: nil, registrationType:
    .sameLan, success: { (candidate) in
        // register candidate
    }, failure: { (error) in
        // Handle error
    })
```

2. Register the candidate

```
registration.register(candidate, success: { (AylaDevice) in
    // Successfully Registered Device.
}, failure: { (error) in
    // Registration Failed, handle error
})
```

## Potential pitfalls

- Make sure device and phone are connected to the same LAN and no filter or firewall is stopping traffic between them. Using a phone as a Wi-Fi hotspot may not work because all traffic is forwarded to the carrier.
- Some devices go to sleep after a few minutes and will stop replying to any request, including the one to fetch the registration token, to fix this you can power cycle the device.
- Device is already registered, if you just performed WiFi setup you should check if device has been registered to user in the past to avoid receiving a failure.

## AP Mode

When WiFi setup starts a token is generated by SDK and sent to IoT device. The setup token is also made available to you when you execute `confirmDeviceConnected`. The device will submit the token to Ayla Device Service (ADS) upon connection (setup completion). The setup token is used by the mobile SDK to confirm the ownership of the device and register it.

## Details

1. Create an instance of `AylaRegistrationCandidate` with the required fields

```
let candidate = AylaRegistrationCandidate()
candidate.registrationType = .apMode
candidate.setupToken = setupToken // from confirmDeviceConnected
candidate.dsn = dsn // from confirmDeviceConnected
```
2. Register the candidate

```
registration.register(candidate, success: { (AylaDevice) in
    // Successfully Registered Device.
}, failure: { (error) in
    // Registration Failed, handle error
})
```

## Potential Pitfalls

- If you are not going to be able to register the device right after performing Wi-Fi setup, make sure to store the token and DSN somewhere to retrieve them when you're ready to register. But not too long: the setup token has a limited life-time of a few hours.
- If the setup token is lost or expires before the device is registered, the process must begin again with Wi-Fi Setup so that a new setup token can be passed.
- AP Mode registration cannot be used with WPS or Ethernet based setup: the setup token is passed to the device during setup, which doesn't happen with these setup modes.

## Button Push

A 2 minutes registration window is started on Ayla Device Service when a registration button on the device is pushed. The device can only be registered during this registration window. Re-registration of the device to a different user is allowed while the window is open.

## Details

1. Create an instance of `AylaRegistrationCandidate` with the required fields. You can get the DSN from the `setupDevice` after WiFi setup or by fetching the candidates with Button-Push registration type:
  - a. After WiFi setup

```
let candidate = AylaRegistrationCandidate()
candidate.registrationType = .buttonPush
candidate.dsn = dsn // from confirmDeviceConnected
```
  - b. By fetching candidates

```
registration.fetchCandidate(withDSN: nil, registrationType:
.buttonPush, success: { (candidate) in
```

```
        //register candidate
        }, failure: { (error) in
            // Handle error
        })
    })
```

## 2. Register the candidate

```
registration.register(candidate, success: { (AylaDevice) in
    // Successfully Registered Device.
}, failure: { (error) in
    // Registration Failed, handle error
})
```

## Potential Pitfalls

- Registration window times out. Press the button again to reopen the window.
- Pressing the button opens the device up to new ownership. Only consider this method for IoT devices where physical access is limited, or ownership is unimportant.

## Display

Display mode registration requires the registration token to be sent to the cloud. The registration token is created on the cloud and sent to the device where it is displayed. The user then reads the registration token and enters it into the application. This registration method does not require the user and device to be in the same LAN. The device can be re-registered using its unique reg token.

### Details

1. Set the candidate registration type to display:

```
let candidate = AylaRegistrationCandidate()
candidate.registrationType = .display
```

2. The user should also enter the token from the device. Assign this token to the registrationToken:

```
candidate.registrationToken = regToken
```

3. Register the candidate

```
registration.register(candidate, success: { (AylaDevice) in
    // Successfully Registered Device.
}, failure: { (error) in
    // Registration Failed, handle error
})
```

### Potential Pitfalls

- Make sure to use registration token and not setup token.

## DSN

Device Serial Number (DSN) registration registers the device using the device's Ayla unique identifier.

### Details

1. Set the candidate dsn and registration type to display:

```
let candidate = AylaRegistrationCandidate()
candidate.registrationType = .dsn
candidate.registrationToken = dsn
```

2. Register the candidate

```
registration.register(candidate, success: { (AylaDevice) in
    // Successfully Registered Device.
}, failure: { (error) in
    // Registration Failed, handle error
})
```

```
})
```

## Potential Pitfalls

- DSN registration is the least secure registration method because the DSN could be guessed. That said, the registration window is limited after an IoT device connects.

## Node registration

Node type registration is used to register nodes discovered through an Ayla IoT gateway device.

### Details

1. To register nodes to a gateway, apps should first fetch the available registration candidates for that node using the method.

```
targetGateway.fetchCandidates(success: { (nodes) in
    // Display candidates
}, failure: { (error) in
    // No node candidates found
})
```

2. Register selected candidate:

```
func tableView(_ tableView: UITableView, didSelectRowAt
indexPath: IndexPath) {
    let candidate = candidateNodeList[indexPath.row]
    registration.register(candidate, success:{(AylaDevice) in
        }, failure: { (error) in
        })
}
```

## Potential Pitfalls

- Make sure the candidate is within the range of the gateway
- Only nodes visible to a gateway can be registered to it