Article

Updating controls locally and remotely

Update and reload controls from your app or using push notifications.

Overview

A control reloads and updates when someone interacts with it, when the app asks to reload it using the shared `ControlCenter`, or when the system receives a specific remote push notification from Apple Push Notification service (APNs). When someone interacts with a control, it's reloaded once the assigned app intent finishes its `perform()` function. Plan for when someone interacts with your app or website and the state of a control needs to update to reflect the changes.

Request reloading controls from an app

Reload controls when their state changes after someone interacts with them, such as turning a light on or off from within your app. Use the `reloadControls(ofKind:)` method on the shared `ControlCenter` to ask the system to reload a control from your app that matches a specific `kind` unique identifier.

The following code shows reloading a garage door control toggle from within the app:

```
func toggleGarageDoor {
    //Open or close a garage door.

    ControlCenter.shared.reloadControls(ofKind: "com.example.myApp.garageDoorToggle"
}
```

Use the `reloadAllControls()` method to ask the system to reload all controls your app provides.

Use push notifications to reload controls

You may need the system to reload controls when someone interacts with a web server connected to your app that affects their state. Enable a control to update its state from a push notification event by registering a `ControlPushHandler` type on the `ControlWidgetConfiguration`. The system reloads any controls with a registered push handler in response to a push notification event. You don't need to reload the controls manually.

The `ControlPushHandler` requires a `pushTokensDidChange(controls:)` function that is called when push tokens change. The system calls this function when you add or remove controls, and may call it at other times if any push tokens change or if someone reconfigures a control.

The `controls` parameter describes all controls from your app registered with this handler type, which allows you to batch all push tokens together and send them to your server at the same time. Each `ControlInfo` object represents a control and includes the following:

- The `kind` you set when creating the control.
- The `configurationIntent(of:)` function. If your control is configurable, this function provides the custom intent containing the user-configured value.
- The `pushInfo` containing the unique push token that's used to deliver updates for this control, if a push handler is registered.

The following code registers a push handler on a control that opens and closes a garage door:

```
struct GarageDoorOpener: ControlWidget {
    var body: some ControlWidgetConfiguration {
        StaticControlConfiguration(
            kind: "com.apple.GarageDoorOpener"
        ) {
            ControlWidgetToggle(...)
        }
        .pushHandler(MyPushHandler.self)
    }
}

struct MyPushHandler: ControlPushHandler {
    func pushTokensDidChange(controls: [ControlInfo]) {
        // Send push tokens and subscription info to your server.
        // ...
    }
}
```

Query for the current set of your controls someone placed across their system in your app by calling `currentControls()` on the shared `ControlCenter`. Introspect the push tokens, if any, from the set of controls to sync state with your app and server.

The following code queries for the current set of controls and gets any push tokens associated with them:

```
let controls = await ControlCenter.shared.currentControls()
let pushTokens = controls.compactMap { $0.pushInfo?.token }

// Sync push tokens with your server.
```

Use APNs when your server has a state change to communicate to controls on someone's devices. The POST request indicates that state changed for the single control representing that given push token, and the system reloads the control.

The code snippet below shows a sample POST request created with an authentication token:

```
HEADERS
  - END_STREAM
  + END_HEADERS
  :method = POST
  :scheme = https
  :path = /3/device/00fc13adff785122b4ad28809a3420982341241421348097878e577c991de8f0
  host = api.sandbox.push.apple.com
  authorization = bearer eyAia2lkIjogIjhZTDNHM1JSWDciIH0.eyAiaXNzIjogIkM4Nk5WOUpYM0C
         jogIjE0NTkxNDM1ODA2NTAiIH0.MEYCIQDzqyahmH1rz1s-LFNkylXEa2lZ_aOCX4daxxTZkVEC
         Lxw7LZtEQcH6JENhJTMArwLf3sXwi
  apns-id = eabeae54-14a8-11e5-b60b-1697f925ec7b
  apns-push-type = controls
  apns-expiration = 0
  apns-priority = 10
  apns-topic = com.example.MyApp.push-type.controls
DATA
  + END_STREAM
  { "aps" : { "content-changed" : true } }
```

See Also

Updates

`protocol` **ControlPushHandler**

A type that can receive push information about user-configured controls.

`struct` **ControlPushInfo**

A structure that contains information about the push token of a user-configured control.