

Connection to an existing session

Since iOS SDK build [2.6.76](#), it is possible to connect to an existing websocket session on the server to accept an incoming call when push notification is received:

1. Set `keepAlive` option when connection is established for the first time

```
- (FPWCSApi2Session *)connectWithOptions:(FPWCSApi2SessionOptions
*)options {
    options.urlServer = _connectUrl.text;
    options.keepAlive = true;
    options.sipRegisterRequired = _sipRegRequired.control.isOn;
    options.sipLogin = _sipLogin.input.text;
    options.sipAuthenticationName = _sipAuthName.input.text;
    options.sipPassword = _sipPassword.input.text;
    options.sipDomain = _sipDomain.input.text;
    options.sipOutboundProxy = _sipOutboundProxy.input.text;
    options.sipPort = [NSNumber numberWithInt: [_sipPort.input.text
integerValue]];
    options.appKey = @"defaultApp";
    ...
    session = [FPWCSApi2 createSession:options error:&error];
    ...
    [session connect];
    return session;
}
```

2. Keep a session token after successful connection

```
[session on:kFPWCSSessionStatusEstablished callback:^(FPWCSApi2Session
*rSession){
    _authToken.input.text = [rSession getAuthToken];
    [self changeConnectionStatus:[rSession getStatus]];
    [self onConnected:rSession];
    if (!_sipRegRequired.control.isOn) {
        [self changeViewState:_callButton enabled:YES];
    }
}
}}];
```

Then, the session can be disconnected on mobile device when application goes to background, but the session will be kept on server during 1 hour by default.

3. When push notification is received, connect to the existing session by token

```
- (FPWCSApi2Session *)connectWithToken {
    FPWCSApi2SessionOptions *options = [[FPWCSApi2SessionOptions alloc]
init];
    options.authToken = _authToken.input.text;
```

```
        return [self connectWithOptions:options];
    }
}
```

4. Receive incoming call event and create answer/hangup alert dialog

```
[session onIncomingCallCallback:^(FPWCSCall *rCall) {
    call = rCall;
    ...
    alert = [UIAlertController
        alertControllerWithTitle:[NSString
            stringWithFormat:@"Incoming call from '%@'", [rCall getCallee]]
        message:error.localizedDescription
        preferredStyle:UIAlertControllerStyleAlert];

    UIAlertAction* answerButton = [UIAlertAction
        initWithTitle:@"Answer"
        style:UIAlertActionStyleDefault
        handler:^(UIAlertAction * action) {
            [call answer];
        }];

    [alert addAction:answerButton];
    UIAlertAction* hangupButton = [UIAlertAction
        initWithTitle:@"Hangup"
        style:UIAlertActionStyleDefault
        handler:^(UIAlertAction * action) {
            [call hangup];
        }];

    [alert addAction:hangupButton];
    [self presentViewController:alert animated:YES completion:nil];
}];
```

5. Accept the incoming call

```
[call on:kFPWCSCallStatusEstablished callback:^(FPWCSCall *call){
    [self changeCallStatus:call];
    [self toHangupState];
    [self changeViewState:_holdButton enabled:YES];
}];
```