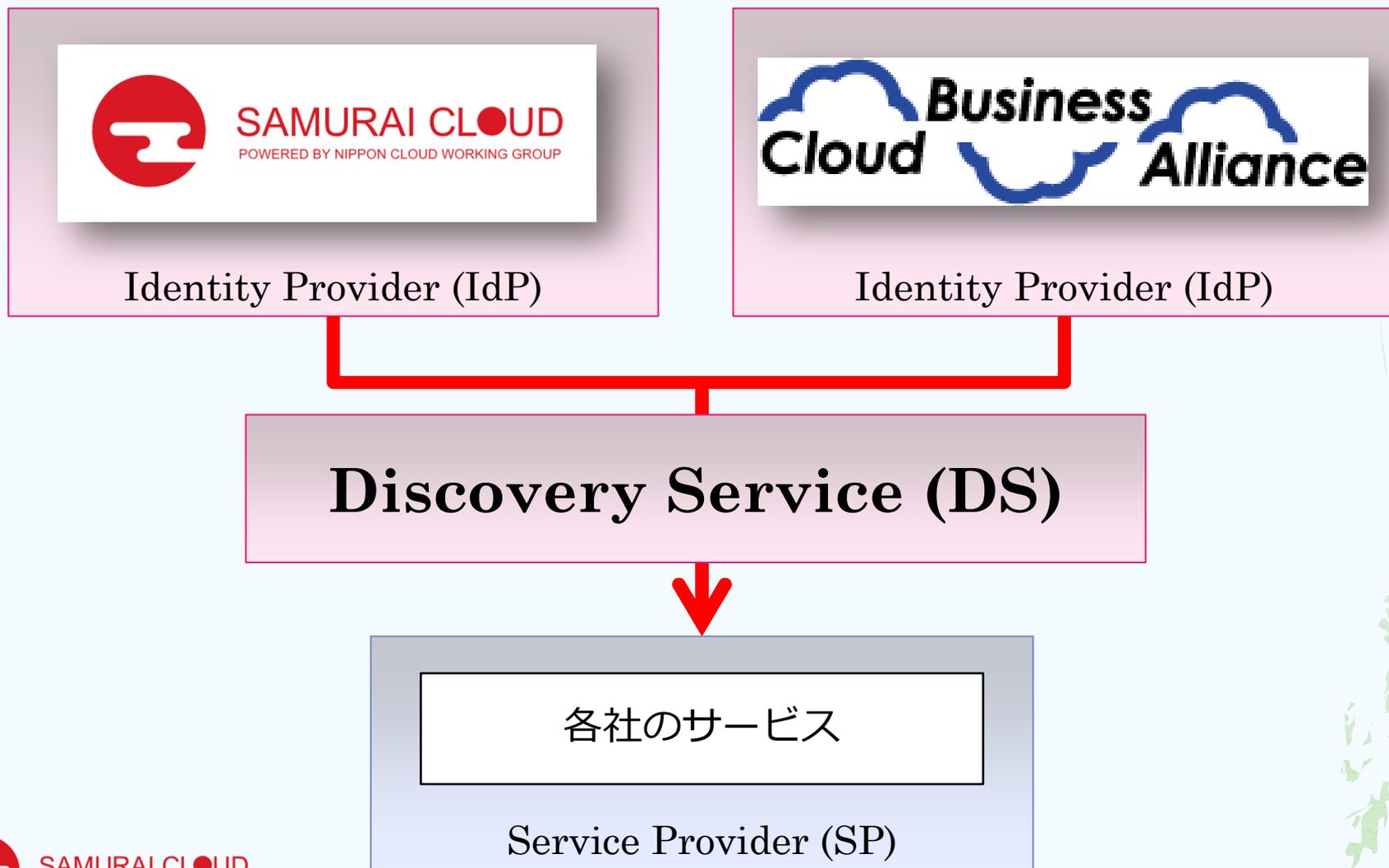


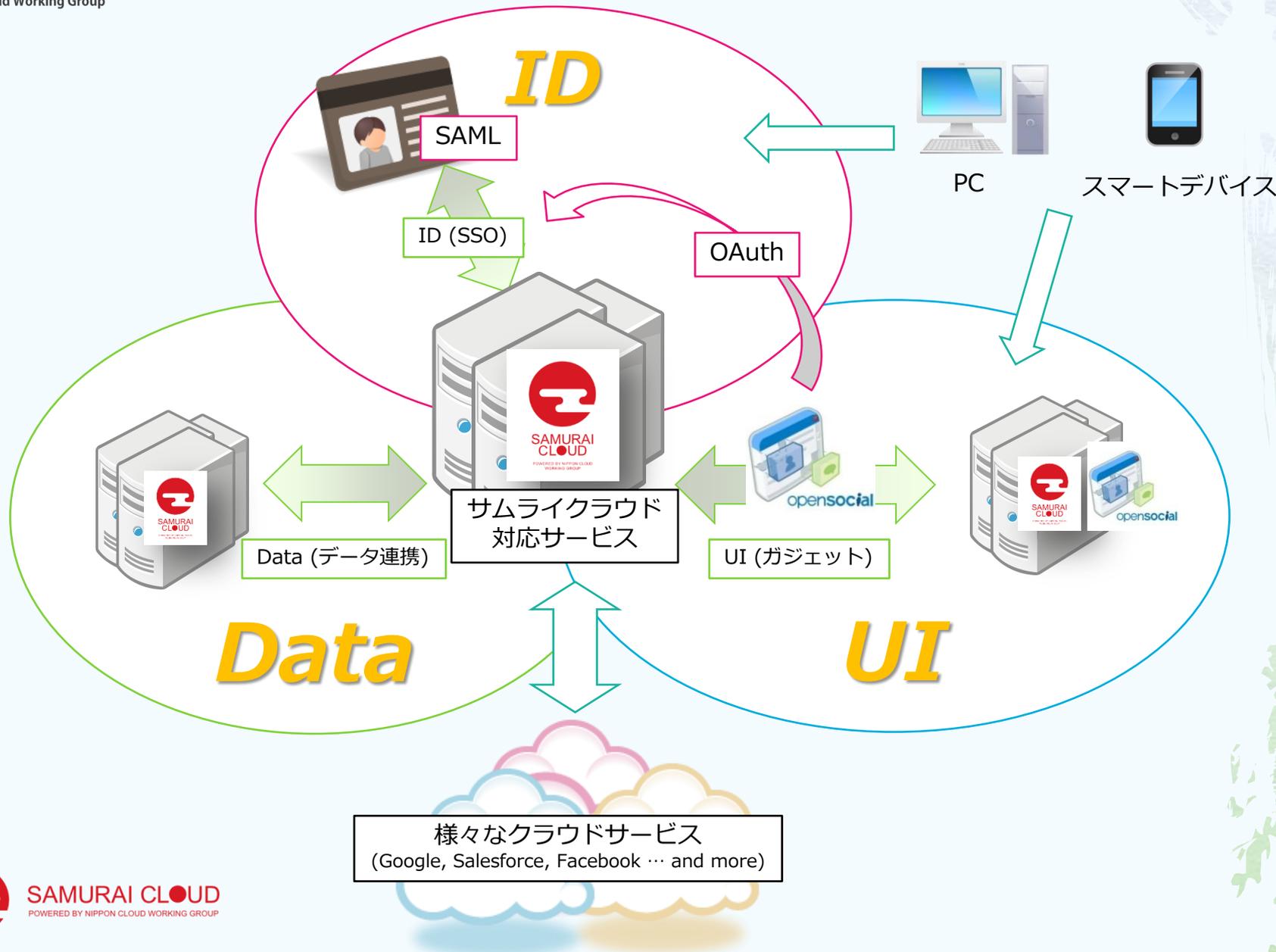
# サムライクラウド部会報告 2013年 2月 5日

---

# CBAとの相互接続性検証



# サムライクラウド連携





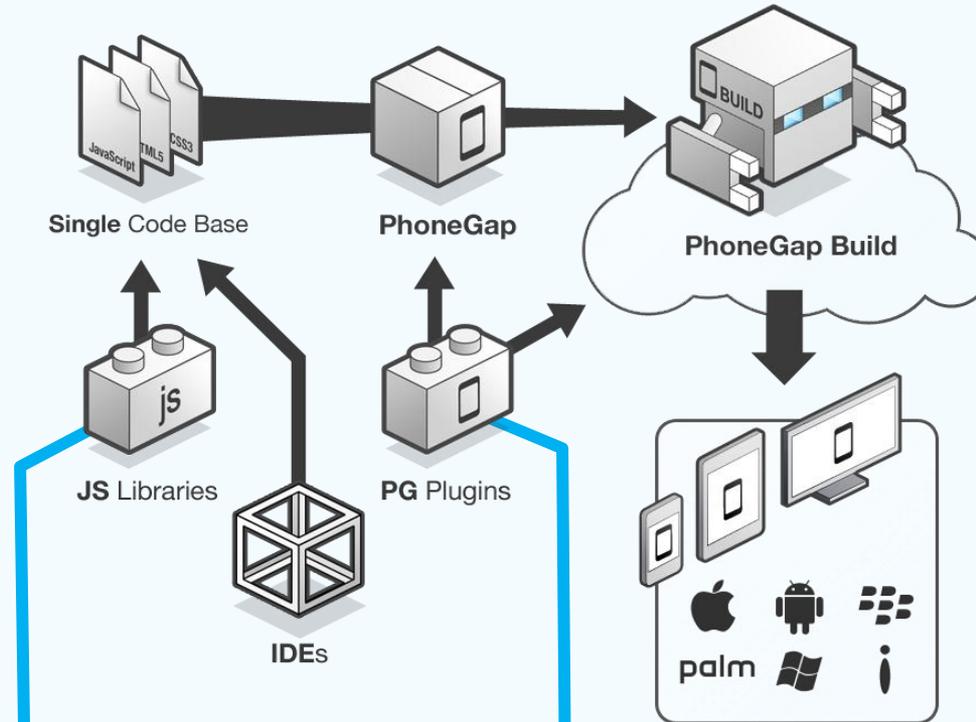
NIPPON Cloud Working Group

---

# PHONEGAPで OAUTH2.0対応アプリを作る

<http://ncwg.jp/>

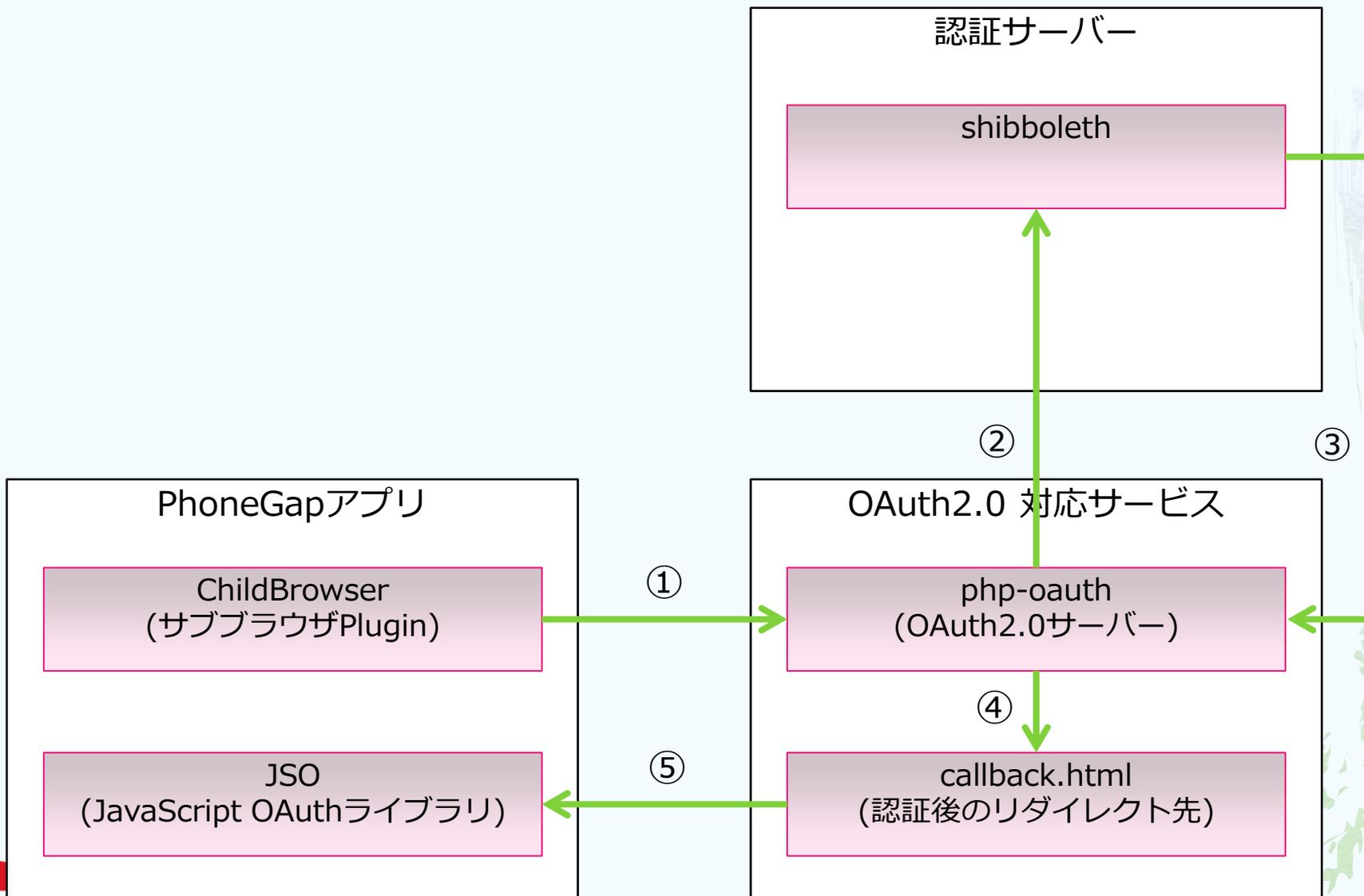
# アプリ構成概要



**JSO**(javascript OAuth 2.0 library)

**ChildBrowser**

# アクセストークン取得の流れ



# アプリ登録

## Application Registration

**Identifier**

**Name**

**Profile**  ▼

**Web Application**  
A client running on a web server, like 3-legged OAuth 1.0a, MUST have a secret.

**User-agent-based Application**  
The client runs in the web browser only.

**Native Application**  
The client runs on a mobile device or is a 'rich' desktop application.

**Redirect URI**

**Secret**

**Description**

**Icon**

**Allowed Scope**

**Contact**

Close Save changes

# HTML

```

index.html  index.js
1  <!DOCTYPE html>
2  <html>
3  <head>
4  »   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
5  »   <meta name="format-detection" content="telephone=no" />
6  »   <meta name="viewport" content="width=device-width, initial-scale=1" />
7  »   <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">
8  »   <link href="bootstrap/css/bootstrap-responsive.min.css" rel="stylesheet">
9  »   <title>View Your Grades</title>
10 </head>
11 <body>
12 »   <script type="text/javascript" src="cordova-2.3.0.js"></script>
13 »   <script type="text/javascript" src="childbrowser.js"></script><!-- OAuth認証画面表示用、PhoneGapプラグイン -->
14 »   <script type="text/javascript" src="jquery-1.8.3.min.js"></script>
15 »   <script type="text/javascript" src="jquery.activity-indicator-1.0.0.min.js"></script>
16 »   <script type="text/javascript" src="jso.js"></script><!-- OAuth 2.0ライブラリ -->
17 »   <script type="text/javascript" src="jsrender.js"></script>
18 »   <script type="text/javascript" src="bootstrap/js/bootstrap.min.js"></script>
19 »   <script type="text/javascript" src="index.js"></script><!-- アプリのメインロジック -->
20
21 »   <div class="app">
22 »     <div class="navbar navbar-inverse navbar-fixed-top">
23 »       <div class="navbar-inner">
24 »         <div class="container">
25 »           <a class="brand" href="#">View Your Grades</a>
26 »         </div>
27 »       </div>
28 »     </div>

```

# JavaScript (認証)

```

index.html | index.js
37 >>    };
38
39 >>    // OAuthフロー中のリダイレクト要求イベントのハンドラを登録
40 >>    jso_registerRedirectHandler(
41 >>    >>    function(url) {
42 >>    >>    >>    window.plugins.childBrowser.showWebPage(url);
43 >>    >>    });
44
45 >>    // OAuth設定
46 >>    var jsoParams = {};
47 >>    jsoParams[oauth_providerID] = {
48 >>    >>    >>    client_id: "ここにクライアントID",
49 >>    >>    >>    redirect_uri: "https://sp-demo02.samuraicloud.jp/callback/phonegap.html",
50 >>    >>    >>    authorization: "https://sp-demo02.samuraicloud.jp/php-oauth/authorize.php"
51 >>    >>    };
52 // jsoParams["debug"] = true;
53 >>    jso_configure(jsoParams);
54 >>
55 >>    // OAuth認証
56 >>    var ensureOpt = {};
57 >>    ensureOpt[oauth_providerID] = oauth_scope;
58 >>    jso_ensureTokens(ensureOpt);
59 >>
60 >>    // アクションの初期化
61 >>    intActions();
62 >>
63 >>    // サービス実行
64 >>    executeService();
65 }
66

```

アクセストークンの取得。  
既に取得済みのものがあれば認証はスキップされる。

# 参考 (JSON処理①)

```
index.html | index.js | jso.js ✖
434
435 >> log("Saving state [" + state + "]);
436 >> log(JSON.parse(JSON.stringify(request)));
437
438 >> api_storage.saveState(state, request);
439 >> api_redirect(authurl);
440
441 >> };
442
443 >> exp.jso_ensureTokens = function(ensure) {
444 >> >> var providerid, scopes, token;
445 >> >> for(providerid in ensure) {
446 >> >> >> scopes = undefined;
447 >> >> >> if(ensure[providerid]) scopes = ensure[providerid];
448 >> >> >> token = api_storage.getToken(providerid, scopes);
449 >>
450 >> >> >> log("Ensure token for provider [" + providerid + "]);
451 >> >> >> log(token);
452 >>
453 >> >> >> if(token === null) {
454 >> >> >> >> jso_authrequest(providerid, scopes);
455 >> >> >> }
456 >> >> }
457
458
459 >> >> return true;
460 >> }
461
```

ストレージから既存トークン取得。  
(デフォルトではローカルストレージ)

if(token === null) {  
 jso\_authrequest(providerid, scopes);  
}

トークンが無ければ認証リクエスト。

# 参考 (JSON処理②)

```

index.html | index.js | jso.js
383 >> /*
384 >>  * A config object contains:
385 >> */
386 >> var jso_authrequest = function(providerid, scopes, callback) {
387 >>
388 >>     var
389 >>     >>     state,
390 >>     >>     request,
391 >>     >>     authurl,
392 >>     >>     co;
393 >>
394 >>     if (!config[providerid]) throw "Could not find configuration for provider." + providerid;
395 >>     co = config[providerid];
396 >>

```

}

```

426 >>     if (window.location.hash) {
427 >>         request["restoreHash"] = window.location.hash;
428 >>     }
429 >>     request["providerID"] = providerid;
430 >>     if (scopes) {
431 >>         request["scopes"] = scopes;
432 >>     }
433 >>
434 >>
435 >>     log("Saving state [" + state + "]);
436 >>     log(JSON.parse(JSON.stringify(request)));
437 >>
438 >>     api_storage.saveState(state, request);
439 >>     api_redirect(authurl);
440 >>
441 >> };

```

Stateをストレージに保存して認証画面をChildBrowserで開く。

# JavaScript (認証終了①)

```

18  ⌘
19  var oauth_providerID = "samuraicloud"; //プロバイダ識別子(任意)⌘
20  var oauth_scope = ["grades"]; //OAuthのスコープ⌘
21  ⌘
22  //PhoneGap初期化完了イベント⌘
23  document.addEventListener("deviceready", onDeviceReady, false);⌘
24  function onDeviceReady(){⌘
25  »   window.plugins.childBrowser.onLocationChange = function(loc){⌘
26  »       // childBrowserで画面遷移イベント処理⌘
27  »       console.log("onLocationChange:" + loc);⌘
28  »       url = decodeURIComponent(loc);⌘
29  »       jso_checkfortoken(⌘
30  »           »   oauth_providerID,⌘
31  »           »   url,⌘
32  »           »   function(){⌘
33  »               »   console.log("Closing child browser, because a valid response was detected.");⌘
34  »               »   window.plugins.childBrowser.close();⌘
35  »               »   executeService();⌘
36  »           »   });⌘
37  »   };⌘
38  ⌘

```

ChildBrowserの画面遷移をハンドリング。

認証完了の画面かを判定し、アクセストークンを取得。

# 参考 (JSON処理③)

```

index.html | index.js | jso.js x
275 >> exp.json_checkfortoken = function(providerID, url, callback) {
276 >>     var
277 >>         atoken,
278 >>         h = window.location.hash,
279 >>         now = epoch(),
280 >>         state,
281 >>         co;
282
283 >>     log("json_checkfortoken(" + providerID + ")");
284
285 >>     // If a url is provided
286 >>     if (url) {
287 >>         // log('Hah, I got the url and it is ' + url);
288 >>         if (url.indexOf('#') === -1) return;
289 >>         h = url.substring(url.indexOf('#'));
290 >>         // log('Hah, I got the hash and it is ' + h);
291 >>     }
292
293 >>     /*
294 >>     .* Start with checking if there is a token in the hash
295 >>     */
296 >>     if (h.length < 2) return;
297 >>     if (h.indexOf("access_token") === -1) return;
298 >>     h = h.substring(1);
299 >>     atoken = parseQueryString(h);
300

```

2

URLからアクセストークンを抽出。

# 参考 (JSO処理④)

```

index.html | index.js | jso.js ✕
348  »  »  */
349  »  »  if (atoken["scope"]) {
350  »  »  »  atoken["scopes"] = atoken["scope"].split(".");
351  »  »  } else if (state["scopes"]) {
352  »  »  »  atoken["scopes"] = state["scopes"];
353  »  »  }
354  »
355  »
356  »
357  »  »  api_storage.saveToken(state.providerID, atoken);
358  »
359  »  »  if (state.restoreHash) {
360  »  »  »  window.location.hash = state.restoreHash;
361  »  »  } else {
362  »  »  »  window.location.hash = '';
363  »  »  }
364  »
365  »
366  »  »  log(atoken);
367  »
368  »  »  if (internalStates[atoken.state] && typeof internalStates[atoken.state] === 'function') {
369  »  »  »  // log("InternalState is set, calling it now!");
370  »  »  »  internalStates[atoken.state]();
371  »  »  »  delete internalStates[atoken.state];
372  »  »  }
373  »
374  »
375  »  »  if (typeof callback === 'function') {
376  »  »  »  callback();
377  »  »  }
378  »
379  »  »  // log(atoken);
380  »
381  »  }

```

ストレージにトークンを保存。

api\_storage.saveToken(state.providerID, atoken);

指定されたコールバック関数を呼び出して終了。

if (typeof callback === 'function') {  
 callback();  
}

# JavaScript (認証終了②)

```

index.html | index.js | jso.js
18
19 var oauth_providerID = "samuraicloud"; //プロバイダ識別子(任意)
20 var oauth_scope = ["grades"]; //OAuthのスコープ
21
22 //PhoneGap初期化完了イベント
23 document.addEventListener("deviceready", onDeviceReady, false);
24 function onDeviceReady() {
25     window.plugins.childBrowser.onLocationChange = function(loc) {
26         // childBrowserで画面遷移イベント処理
27         console.log("onLocationChange: " + loc);
28         url = decodeURIComponent(loc);
29         jso_checkfortoken(
30             oauth_providerID,
31             url,
32             function() {
33                 console.log("Closing child browser, because a valid response was detected.");
34                 window.plugins.childBrowser.close();
35                 executeService();
36             });
37     };
28

```

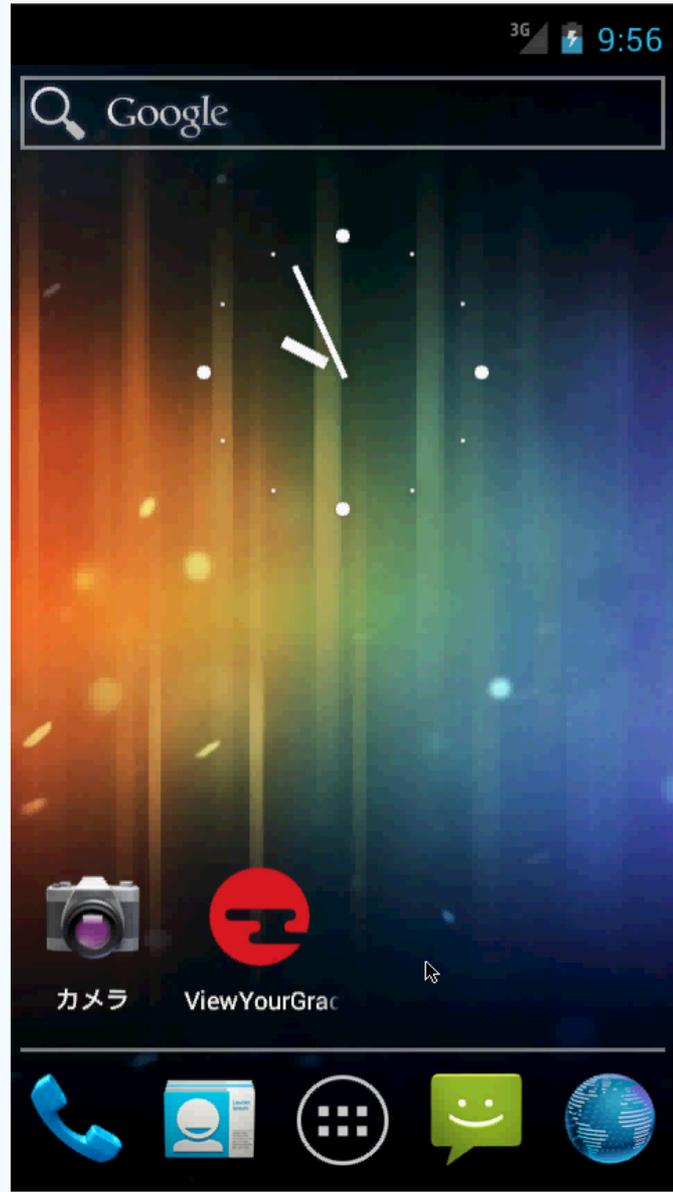
呼び出されるコールバック処理。

# JavaScript (OAuthリクエスト)

```
index.html | index.js
109
110 // 生徒一覧の取得・表示
111 function renderStudentBox(){
112     $.ajax({
113         url: "https://sp-demo02.samuraicloud.jp/php-oauth-grades-rs/api.php/grades/",
114         jso_provider: "oauth_providerID",
115         jso_scopes: "oauth_scope",
116         jso_allowia: true,
117         dataType: "json",
118         async: true,
119         success: function(data){
120             $("#studentList").html($("#studentListTemplate").render(data));
121             $("#studentListForm").show();
122             renderGradeList(data[0].id);
123         }
124         $("#a#studentShow").click(function(){
125             var f = parseForm($("#form#studentListForm"));
126             renderGradeList(f.studentList);
127         });
128     });
129 }
130
```

APIの実行。  
(jQueryのajax拡張)

# デモ



# facebookアプリも作れます

## facebook Sample



### ニッポンクラウドワーキンググループ

サムライクラウド部会！！@マインドで開催中です！！



### ニッポンクラウドワーキンググループ

第12回会合のメンバー発表内容も決まりましたので、再  
2013年2月5日(火) 17:00~19:00(16

**【場所】**

ニフティ株式会社様 18階セミナールーム  
<http://www.nifty.co.jp/company/outline/>

- アシアル株式会社 田中正裕

「モバイルアプリにおけるクロスプラットフォーム開発の  
近年、iOS、Android、Windows 8などの多種多様な  
本発表では、クロスプラットフォーム向けライブラリー

- 株式会社データ変換研究所 畑中豊司

「Android機でUSB機器を制御するDerimo(デリモ)のご  
〜機器制御センサをネットワークを通じてスマホで活用  
Derimoを活用することで、汎用的な手法でUSB機器制御  
現状、携帯電話はスマートフォンが普及してきており、  
会合ならびに懇親会への参加申し込みはfacebookイベント

# URL

## ◆ 利用ライブラリ

- ◆ JSO (A javascript OAuth 2.0 library)
  - ◆ <https://github.com/andreassolberg/jso>
- ◆ php-oauth (PHP OAuth 2.0 Authorization Server)
  - ◆ <https://github.com/fkooman/php-oauth>
- ◆ SimpleSAMLphp
  - ◆ <http://simplesamlphp.org/>

## ◆ 参考情報

- ◆ OAuth 2.0 オフィシャルサイト
  - ◆ <http://oauth.net/2/>
- ◆ OAuth 2.0 RFC日本語訳
  - ◆ <http://openid-foundation-japan.github.com/rfc6749.ja.html>

# お知らせ

- ◆ CBAクラウドバレンタイン2013 で共同発表します
  - ◆ 日時
    - ◆ 2013年2月13日(水) 15:00 ~
    - ◆ ホテル日航東京(お台場)
  - ◆ 内容
    - ◆ 「フロント連携WG活動紹介」の中で、共同で行っている活動の内容を発表します。
  
- ◆ サムライクラウド部会
  - ◆ ミニ部会を開催しました（1月31日実施）
    - ◆ 今後、部会の中ではテーマを絞り継続して検証・開発を進める
      - ◆ 現在有力なテーマはスマホ
    - ◆ 並行して、具体的な連携の可能性に関するブレストも定期実施する
  - ◆ 次回部会
    - ◆ 2月下旬予定
      - ◆ 決まり次第アナウンス