

Содержание

- [1 Разработка модуля с OAuth авторизацией](#)
 - [1.1 1. Определения.](#)
 - [1.2 2. Важно!!!](#)
 - [1.3 3. Функции и события](#)
 - [1.3.1 3.1. Функции](#)
 - [1.3.2 3.2. Событие открытия страницы](#)
 - [1.4 4. GMAIL OAUTH](#)

Разработка модуля с OAuth авторизацией

1. Определения.

Модуль - это компонент, который позволяет вам контролировать один тип оборудования по его уникальному ID. Модуль расположен в облачном хранилище.

OAuth — открытый протокол авторизации, который позволяет предоставить третьей стороне ограниченный доступ к защищенным ресурсам пользователя без необходимости передавать ей (третьей стороне) логин и пароль.

2. Важно!!!

OAuth авторизация нужна чтобы дать приложениям доступ к вашей учетной записи . OAuth авторизация работает только на устройствах с операционными системами android и iOS, это значит что проверить работоспособность вашего модуля на Windows невозможно. Разработка модуля для android и iOS немного отличается.

3. Функции и события

3.1. Функции

Для открытия формы авторизации нужно открыть браузер для этого используем эту функцию:
`IR.OpenBrowser(url);`

Входным параметром является URL необходимой формы

Чтобы закрыть форму и браузер нужно воспользоваться функцией `IR.CloseBrowser();`

3.2. Событие открытия страницы

При открытии страницы срабатывает событие `IR.EVENT_BROWSER_PAGE_STARTED`

Это событие срабатывает и когда вы открываете страницу и когда переходите по ссылкам

Напишем слушатель который для примера будет закрывать страницу когда она открылась

```

IR.AddListener(IR.EVENT_BROWSER_PAGE_STARTED, 0, function(url) {
IR.CloseBrowser();
});
ВАЖНО!!! для iOS нужно продублировать слушатель с событием
EVENT_BROWSER_PAGE_FINISHED

```

целиком это будет выглядеть так:

```

IR.AddListener(IR.EVENT_BROWSER_PAGE_STARTED, 0, function(url) {
IR.CloseBrowser();
});
IR.AddListener(IR.EVENT_BROWSER_PAGE_FINISHED, 0, function(url) {
IR.CloseBrowser();
});

```

4. GMAIL OAUTH

Для примера разберём авторизацию через OAuth на Gmail. Нашем пример будет для iOS Для начала создадим объект и запишем входные параметры

```
GMAIL = {};
```

```

GMAIL.OAUTH_TOKEN_URL = "/auth/oauth2/token";
GMAIL.CLIENT_ID = "*****"; //Ваши персональные данные
GMAIL.CLIENT_SECRET = "*****"; //Ваши персональные данные
GMAIL.SCOPE =
"https://mail.google.com+https://www.googleapis.com/auth/gmail.modify+https://www.googleapis.com/auth/gmail.readonly";
GMAIL.REDIRECT_URI = "http://localhost";

```

```

GMAIL.googleApi = IR.CreateDevice(IR.DEVICE_CUSTOM_HTTP_TCP, "GoogleApi",
"www.googleapis.com", 443, "", "", true);
GMAIL.moduleID = ModuleID;
GMAIL.token = ;
GMAIL.email = ;

```

Так же создадим объект отвечающий за авторизацию

```
GMAIL.oAuth = {};
```

Далее напишем функцию получения доступа. Для этого откроем браузер с нужным URL и подпишемся на его открытие:

```

GMAIL.oAuth.getToken = function(in_callback) {
IR.Log("OPEN BROWSER");
IR.OpenBrowser(
"https://accounts.google.com/o/oauth2/auth?" +
"&redirect_uri=" + GMAIL.REDIRECT_URI +
"&response_type=code" +
"&client_id=" + GMAIL.CLIENT_ID +
"&scope=" + GMAIL.SCOPE
);

```

```
IR.AddListener(IR.EVENT_BROWSER_PAGE_STARTED, 0, function(url) {
```

```
    //вырезаем код полученного url, у других сервисов длинна может отличаться
```

```
var idx = url.indexOf("localhost/?code=");
if (idx >= 0) {
var code = url.substr(idx + 16);
IR.CloseBrowser();
```

```
GMAIL.oAuth.SendTokenRequest(code, in_callback);
}
});
IR.AddListener(IR.EVENT_BROWSER_PAGE_FINISHED, 0, function(url) {
var idx = url.indexOf("localhost/?code=");
if (idx >= 0) {
var code = url.substr(idx + 16);
IR.CloseBrowser();
```

```
GMAIL.oAuth.SendTokenRequest(code, in_callback);
}
});
};
```

Наша функция вызывает `GMAIL.oAuth.SendTokenRequest(in_code, in_callback)` - это функция запрашивает токен и обрабатывает его, у неё два входных параметра:

`in_code {string}` - код полученный из браузера

`in_callback {function}` - колбек-функция

Вот так она выглядит:

```
GMAIL.oAuth.sendTokenRequest = function(in_code, in_callback) {
GMAIL.googleApi.SendEx({
Type: "GET",
Url: GMAIL.OAUTH_TOKEN_URL + "?client_id=" + GMAIL.CLIENT_ID +
"&client_secret=" + GMAIL.CLIENT_SECRET +
"&grant_type=authorization_code&code=" + in_code,
Headers: {
"Content-length": "0"
},
cbReceiveText: function(text, code, headers) {
var resp = JSON.Parse(text);
GMAIL.token = "Bearer " + resp.access_token;
IR.SetVariable("Global.gmail_token" + GMAIL.moduleID) = resp.access_token;
IR.SetVariable("Global.gmail_refresh_token" + GMAIL.moduleID) =
resp.refresh_token;
var date = new Date();
IR.SetVariable("Global.gmail_time" + GMAIL.moduleID) = date.getTime() +
resp.expires_in - 10;
```

```
in_callback();
}
});
};
```

Полученный токен может стать не действительным, значит нам нужно проверять его и обновлять если он не действителен, напомним функцию:

```
GMAIL.oAuth.checkAndRefreshToken = function(in_callback) {
var date = new Date();
if (IR.GetVariable("Global.gmail_time" + GMAIL.moduleID) > date.getTime()) {
```

```

if (in_callback) {
in_callback();
};
} else if (IR.GetVariable("Global.gmail_refresh_token" + GMAIL.moduleID) !=
0) {
GMAIL.googleApi.SendEx({
Type: "POST",
Url: GMAIL.OAUTH_TOKEN_URL,
Headers: {
"Content-Type": "application/x-www-form-urlencoded"
},
cbReceiveText: function(text, code, headers) {
var resp = JSON.Parse(text);
GMAIL.token = "Bearer " + resp.access_token;
IR.SetVariable("Global.gmail_token" + GMAIL.moduleID, resp.access_token);
IR.SetVariable("Global.gmail_refresh_token" + GMAIL.moduleID,
resp.refresh_token);
var date = new Date();
IR.SetVariable("Global.gmail_time" + GMAIL.moduleID, date.getTime() +
resp.expires_in - 10);

if (in_callback) {
in_callback();
};
},
Data: [ "&client_id=" + GMAIL.CLIENT_ID +
"&client_secret=" + GMAIL.CLIENT_SECRET +
"&grant_type=authorization_code" +
"&refresh_token=" + IR.GetVariable("Global.gmail_refresh_token")
]
});
} else {
GMAIL.oAuth.GetToken(in_callback);
};
};
IR.SetInterval(15000, function() {
GMAIL.oAuth.checkAndRefreshToken(in_callback);
});

```

Мы разобрали авторизацию через OAuth на Gmail на iOS. Можно дополнить несколькими функциями такими как отправка GET запроса, отправка POST запроса, получение количества непочитанных писем.