سامانه متمرکز احراز هویت

روش احراز هویت به وسیله اعتبار کلاینت

تهیه‌کننده/تهیه‌کنندگان: هادی قهرمان زاده

تاریخ: پنجم تیر ماه سال ۱۴۰۲

شماره ویرایش: ۱/۰۱

تارخچه تغییرات مستند

ردیف	ورژن	تاریخ	توسط	شرح
۱	۱٫۰۰	۱۴۰۰/۱۱/۰۸	خدیجه امیدی نسب	نگارش نسخه نخست
۲	۱٫۰۱	۱۴۰۲/۰۴/۰۵	قهرمان زاده	بازنگری کلی

فهرست مطالب

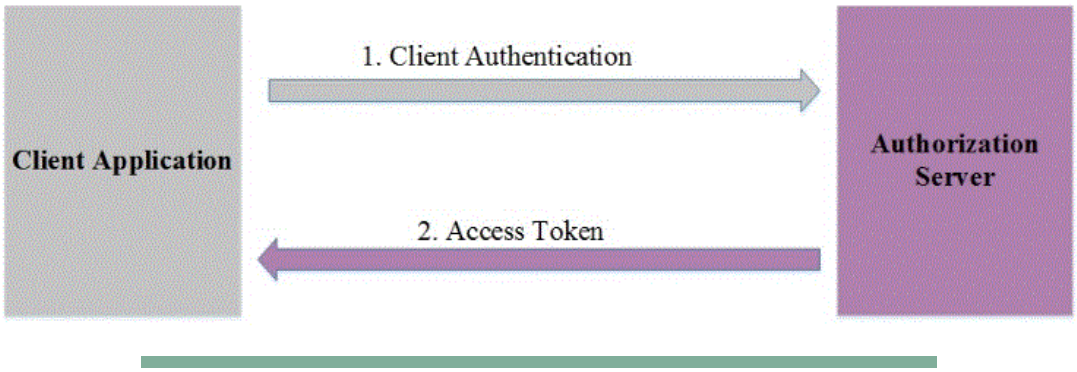
فهرست شکل‌ها

فهرست جداول

مقدمه

یکی از انواع Grant، Client Credentials می‌باشد و هنگامی استفاده می‌شود که برنامه‌ها برای دسترسی به منابع مورد نیاز خود، نه از طرف یک کاربر، Access Token را درخواست می‌کنند. در حقیقت این روند برای شرایطی می‌باشد که کاربری وجود ندارد مانند زمانی که سرور نیاز داشته باشد به یک API خاص متصل شود. بنابراین سرور قرار نیست با اجازه از سمت کسی به API دسترسی داشته باشد بلکه خود سرور به صورت مستقیم از API استفاده خواهد کرد. همان‌طور که گفته شد، معمولا این روش توسط کلاینت‌هایی استفاده می‌شود که می خواهند به منابع مربوط به خود دسترسی داشته باشند نه به منابع کاربر.

در این حالت Client یا همان برنامه کاربردی به جای این که با اجازه کاربر Access Token را دریافت کند، با استفاده از Client ID و Client Secretمختص به خود یک Access Token از Authorization Server درخواست می‌کند و سپس با استفاده از آن token به صورت مستقیم با Resource Server وارد تعامل خواهد شد. شکل زیر این تعامل را نشان می‌دهد.



شکل ۱ روش احراز هویت به وسیله اعتبار کلاینت

با استفاده از سامانه احراز هویت متمرکز، کسب و کار امکان فراخوانی سرویس های سامانه هایی که در آنها ثبت نام نموده است را با استفاده از دریافت توکن خواهد داشت. به این منظور لازم است کسب و کار پیش از فراخوانی سرویس مورد نظر ابتدا با ارسال client_Id و client_secret خود به سامانه احراز هویت اقدام به دریافت توکن نماید سپس با استفاده از توکن دریافتی امکان فراخوانی سرویس های مورد نظر خود در سایر سامانه هایی که با استفاده از سامانه احراز هویت در آنها ثبت نام نموده است را خواهد داشت. به این ترتیب سامانه‌ی ارائه دهنده سرویس با استفاده از توکن دریافتی امکان فراخوانی سرویس مورد نظر توسط کسب و کار را بررسی کرده و در صورتیکه کسب و کار دارای دسترسی های لازم باشد، سرویس مورد نظر را ارائه می‌دهد. این روش احراز هویت همان Client Credentials می باشد.

به هنگام دریافت توکن، کسب و کار می تواند در درخواست خود سطح دسترسی های مورد نظر را تحت عنوان scope ارسال نماید و در صورتیکه این دسترسی ها معتبر باشد، توکن برای کسب و کار ارسال می‌گردد.

توجه شود که نهایتا ارائه سرویس بر اساس scope های تایید شده توسط سامانه احراز هویت مبتنی بر تصمیم سامانه سرویس دهنده می باشد و سامانه سرویس دهنده می تواند بر اساس scope های دریافتی از سامانه احراز هویت در مورد ارائه سرویس مورد نظر تصمیم بگیرد.

برای اطلاعات بیشتر در این مورد به استاندارد RFC ٦٧٤٩ بخش ۴/۴ مراجعه شود (RFC ٦٧٤٩, section ٤/٤)

در نگاهی اجمالی یک OAuth Transaction به دو بخش اصلی تقسیم می شود:

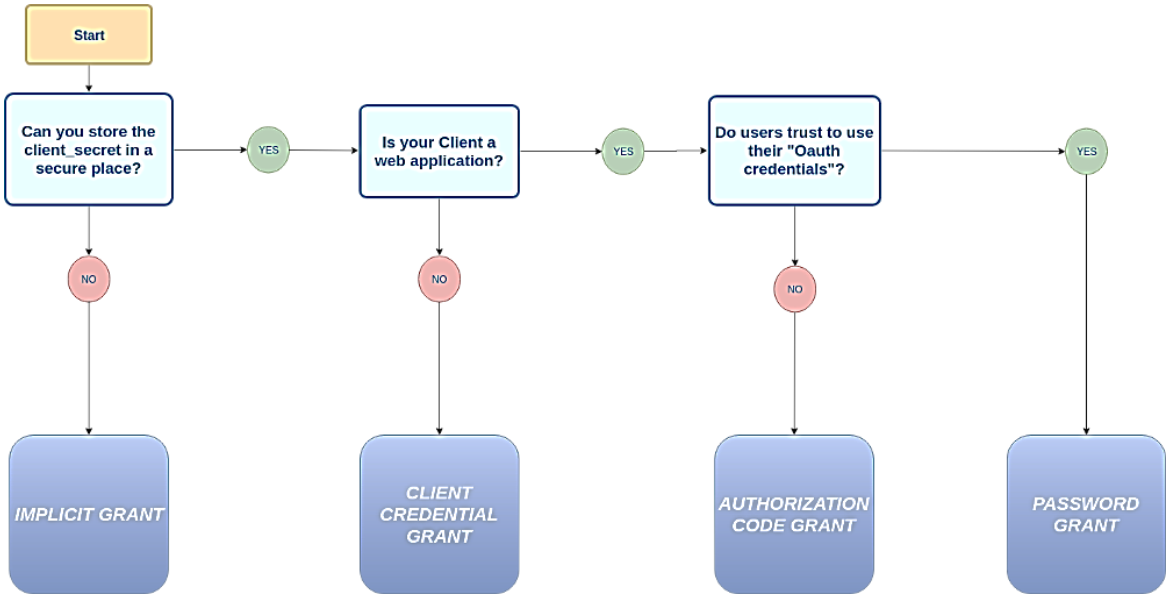
۱- دریافت Access Token توسط کلاینت از Authorization Server. در این مرحله مطابق استاندارد oAuth چند روش (flow) برای احراز هویت وجود دارد. انتخاب روش مناسب (Authorization Grant Type) برای احراز هویت بستگی به نوع کلاینت مورد استفاده، درجه امنیت مورد نیاز و تجربه کاربری مد نظر دارد.

انواع Authorization Grant Type عبارتند از :

- کد مجوز (Authorization Code Grant) رشته کدی که به منظور صدور access token استفاده می شود. این کد موقعی که کاربر در فرانت لاگین می کند منتشر و در مقابل access token در سمت سرور ایجاد می شود. در نتیجه کاربر با رمز عبور و کد حاصله احراز هویت می شود.
- ضمنی (Implicit Grant) وقتی که کاربر لاگین می کند بلافاصله access token مربوطه ایجاد می شود.
- اعتبار کلاینت(Client Credential Grant) access token در سمت سرور تنها برای احراز هویت کلاینت (نه کاربر) ایجاد می شود.
- پسورد (Password Grant OR Resource owner credentials grant) access token بلافاصله تنها با یک request شامل اطلاعات لاگین کاربر (نام کاربری و رمز عبور و یا client id و client secret) اجرای این روش راحت تر است و در عین حال نواقصی هم دارد.

۲- استفاده کلاینت از Access Token دریافتی برای دسترسی به Protected Resource.

انتخاب روش درست به عوامل متعددی بستگی داره. با توجه به فلوچارت زیر و پاسخ به سوالات آن، می‌توان یک روش مناسب را انتخاب نمود.



شکل ۲. روش های احراز هویت

۱ مراحل احراز هویت به وسیله اعتبار کلاینت

کلاینت هایی برای دسترسی به سرویس خاص نیاز به دریافت توکن دارند می‌توانند از این سرویس جهت دریافت توکن استفاده کنند.

درخواست دریافت توکن

جریان اصلی:

۱. کلاینت درخواست دریافت توکن را به سامانه احراز هویت ارسال می کند. (جدول ۱)

۲. سامانه احراز هویت اطلاعات ارسال شده را اعتبارسنجی می کند.

۳. توکن دسترسیدر پاسخ برگشت داده می‌شود.

٤. کلاینت بایستی توکن دریافت شده را به صورت امن ذخیره و جهت فراخوانی سرویس ها استفاده نماید.

۵. فرآیند پایان می‌یابد.

جریان فرعی:

۱. درگام دوم اگر اطلاعات کلاینت نامعتبر بود، سامانه احراز هویت در پاسخ به کلاینت خطای ٤۰۱ ارسال می کند.

۲. کلایت بایستی grant type با مقدار client_claims را داشته باشد، در غیر این صورت سامانه احراز هویت در پاسخ خطای ٤۰۱ ارسال می کند.

۳. در صورتی که کلاینت مجوز اضافه کردن claim را نداشته باشد فیلد client_claims پردازش نمی‌شود.

٤. در صورتی که اطلاعات ارسالی نامعتبر باشند، سامانه احراز هویت در پاسخ به کلاینت خطای ٤۰۰ با شرح خطای مرتبط بر می‌گرداند.

۲ پیوست

جدول ۳. درخواست دریافت توکن

آدرس	https://{{BaseURI}}/oauth/token		
متد	POST		
	نام فیلد (موارد ستاره‌دار ضروری می‌باشند)	نوع فیلد	شرح
پارامترهای هدر	Content-Type*	String	application/x-www-form-urlencoded
پارامترهای درخواست در بدنه درخواست با فرمت x-www-form-urlencoded	grant_type *	String	در این روش احراز هویت ثابت و برابر با client_credentials می‌باشد.
	client_id *	String	این شناسه از سوی سامانه احراز هویت به کلاینت اختصاص می‌یابد.
	client_secret *	String	رمز عبور کلاینت بوده و از سوی سامانه احراز هویت به کلاینت اختصاص می یابد.
	scope *	String	محدوده و سطح دسترسی که در سامانه احراز هویت برای کلاینت تعیین شده است. درصورتی که نیاز به ارسال چند مورد است، از جداکننده فاصله استفاده می شود.
	client_claims	String	در صورتی که کلاینت مجوز اضافه کردن claim را داشته باشد می تواند در درخواست خود claim را ارسال نموده تا این claim در payload توکن قرار بگیرد. مقادیر مد نظر بایستی به صورت ابجکت json ارسال شود.
پارامترهای پاسخ موفق	access_token	String	توکن دسترسی (ساختار توکن و نحوه بازگشایی آن طی مستند جداگانه‌ای حضورتان ارائه می گردد)
	token_type	String	Bearer
	expires_in	number	مدت زمانی که توکن پس از آن منقضی خواهد شد با واحد ثانیه.
	scope	String	محدوده و سطح دسترسی های ارسالی در درخواست قبل، در پاسخ بازگشت داده می شود
	iat	number	زمان تولد توکن با فرمت زمانی Unix time و با دقت ثانیه
پارامترهای پاسخ ناموفق	error	String	متن خطا
	error_description	String	شرح خطا

نمونه درخواست:

```
curl --location --request POST 'https://{{BaseURI}}/oauth/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=client_credentials'\
--data-urlencode 'client_id={{client_id}} \
--data-urlencode 'client_secret={{client_secret}} \
--data-urlencode 'scope={{scope}}' \
```

نمونه پاسخ موفق:

```
{
    "access_token": "access token value",
    "token_type": "Bearer",
    "expires_in": ۳۱۵۳۵۹۹۹,
    "scope": "read openid offline_access user write",
    "iat": ۱۶۶۹۱۱۹۶۶۷
}
```

نمونه پاسخ ناموفق:

```
{
    "error": "Bad Request”,
    "error_description": "Exception"
}
```

نکته: لطفا توجه شود که تمامی ارتباطات با سامانه احراز هویت بر روی بستر HTTPS و SSL می باشد.

جدول ۲. کد پاسخ

کد	شرح
٤٠٠	BAD_REQUEST
٤٠١	UNAUTHORIZED
٤٠٢	PAYMENT_REQUIRED
٤٠٣	FORBIDDEN
٤٠٤	NOT_FOUND
٤٠٥	METHOD_NOT_ALLOWED
٤٠٦	NOT_ACCEPTABLE
٤٠٨	REQUEST_TIMEOUT
٤١٥	UNSUPPORTED_MEDIA_TYPE
٤٢٩	TOO_MANY_REQUESTS
٥٠٠	INTERNAL_SERVER_ERROR
٥٠١	NOT_IMPLEMENTED
٥٠٢	BAD_GATEWAY
٥٠٣	SERVICE_UNAVAILABLE
٥٠٤	GATEWAY_TIMEOUT
٥٠٥	HTTP_VERSION_NOT_SUPPORTED
٥٠٩	BANDWIDTH_LIMIT_EXCEEDED
٥١١	NETWORK_AUTHENTICATION_REQUIRED