

Painless Authentication

with Access Tokens



Mathieu Santostefano

💻 Web developer at [joliCode](https://joliCode.com)

Symfony Core Team Member

Twitter: [@welcomattic](https://twitter.com/welcomattic)

GitHub: [@welcomattic](https://github.com/welcomattic)





Last year, on Nov. 18, I received an email from Fabien to join the Core Team

Welcome to the Symfony Core team



Externes



Boîte de réception

Symfony Core Team

Fabien Potencier

jeu. 18 nov. 2021 16:07



français Traduire le message

Hi,

First of all, congratulations for being part of the Symfony Core Team
:)

And thank you for accepting to be part of the Symfony Core Team.



Who has Access Token authentication in their apps?



Who is working with the new Symfony security?



Who is working with the new Symfony security?

Thank you for it, Wouter 🙏

The new Security System

- Removed everything but Guards
- Moved to an event-based system
- Passports and Badges
- Authenticator based

Your job is to use or implement your own Authenticators

The new Security System

Event-based system

- You can interact on different levels
 - CheckPassportEvent
 - AuthenticationTokenCreatedEvent
 - AuthenticationSuccessEvent
 - LoginSuccessEvent
 - LoginFailureEvent
 - LogoutEvent
 - TokenDeauthenticatedEvent
 - SwitchUserEvent

The new Security System

Handle success or failure

- As before, you can still handle what happen in case of authentication sucess or failure



As many things since few years in Symfony, it improves the DX



What is an Access Token?

- `i-am-an-4cc3ss-t0k3n` could be an Access Token
- `mF_9.B5f-4.1JqM` could be an Access Token
- `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikpvag4gRG9lIiwiawFOI` could be an Access token



What is an Access Token?

- `i-am-an-4cc3ss-t0k3n` could be an Access Token
- `mF_9.B5f-4.1JqM` could be an Access Token
- `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvG4gRG9lIiwiaWF0I could be an Access token



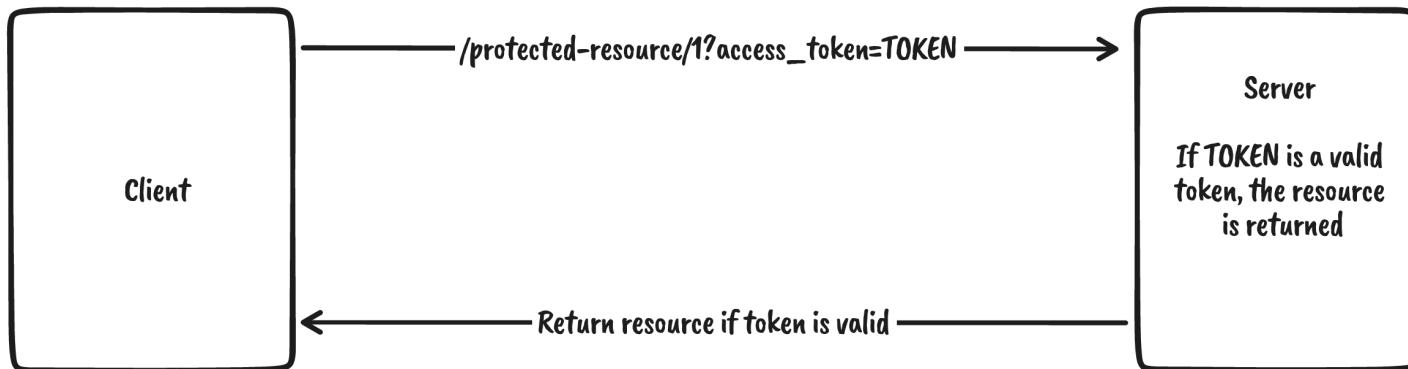
An Access Token is represented by a string.

Ok, but

- Must be sent in a HTTP request to fetch a protected resource
- The application expects to find a token, **verifies** it and decides to allow access or not

Ok, but

- Must be sent in a HTTP request to fetch a protected resource
- The application expects to find a token, **verifies** it and decides to allow access or not





Token issuer

Let's assume our tokens comes from an external authorization server



Verify the token?

The verification process is up to you

- Check if the string is present in a database
- Compute a hash and compare it to the expected one
- Decode the token (base64, ...) and make assertions on decoded values
- Call an external service to validate the token
- Verify the expiration date if needed
- Check if the token is revoked or not
- ...



Use cases

Authentication with Access Token is useful in some contexts, like

- Stateless user login
- API Gateway in front of private APIs
- Application that access to personal data provided by a third party
- Micro-services between them
- Client applications of a SaaS API
- ...



How to set up an Access Token auth with Symfony before 6.2?



The Project : A private API for next Disney movies scenarios

- Form login authentication for users
- Users can create Scenarios
- Each user has an API key
- Users can request our private API with their API key
- We can revoke an API key at any time

With Symfony <= 6.1

- Create a custom Authenticator
 - Handle token extraction
 - Handle failure cases
 - Decode token if needed (JWT, SAML, ...)
 - Retrieve user identifier from the access token
 - Then load User object



Have you heard about RFC6750?

With Symfony <= 6.1

- Handle token extraction
 - In request header?
 - Handle correctly the `Authorization` header with `Bearer` scheme
 - In query?
 - In request body?
- Fill up WWW-Authenticate response header in case of failure
- All requests must be done using HTTPS protocol to secure the token transport

With Symfony <= 6.1

```
1  /** Simplified version */
2  private function extractToken(Request $req): ?string
3  {
4      return match (true) {
5          // Header
6          $req->headers->has('Authorization') => str_replace('Bearer ', '', $req->headers->get('Authorization')),
7
8          // Query string
9          Request::METHOD_GET === $req->getMethod()
10         && $req->query->has('access_token') => $req->query->get('access_token'),
11
12         // Request body
13         Request::METHOD_POST === $req->getMethod()
14         && $req->request->has('access_token') => $req->request->get('access_token'),
15
16         default => null,
17     };
18 }
```

With Symfony <= 6.1

```
1  public function authenticate(Request $request): Passport
2  {
3      if (null === $apiKey = $this->extractToken($request)) {
4          throw new AuthenticationException('No API Key provided');
5      }
6
7      /* Here's the token decoding if needed, then assertions on decoded values */
8
9      return new SelfValidatingPassport(
10         new UserBadge($apiKey, fn(string $apiKey) => $this->userRepository->findOneBy(['apiKey' => $apiKey]))
11     );
12 }
```



We consider the access token as user identifier. In case of JWT, we must decode it to get the user identifier



Boring code

- We have to repeat this code in all our applications, **boring**
- Our responsibility to respect RFC6750, **boring**
- No body likes boring code
- Poor Developer eXperience, Symfony tends to improve DX

This is definitely improvable



Discussions starts long time ago

- April 2019, at EU FOSSA Hackathon

Closed 24 of 35 tasks [RFC] Symfony Security rework tracking issue #30914
curry684 opened this issue on Apr 6, 2019 · 42 comments

Authentication

User impersonation (including multiple impersonation without having to log out) *not affected by Security changes*

Social login (login in the app using Google, Facebook, GitHub, Twitter, etc.) and generic OAuth support ([WIP][Security] OAuth2 component #31952 (comment))

Login throttling (limit the number of failed login attempts over a period of time)

Simultaneous session limiting (e.g. each user can login only from one device at the same time)

Two-factor (or multi-factor) authentication (Native support for 2FA #28868 (comment))



Discussions starts long time ago

- June 2019, first PR about OAuth2 Component. ✘ Aborted

The screenshot shows a GitHub pull request page. The title is "[WIP][Security] OAuth2 component #31952". The status is "Closed" with a red button. The base branch is "symfony:master" and the target branch is "Guikingone:feat/oauth". Below the title, there are metrics: Conversation (45), Commits (2), Checks (0), and Files changed (67). The author is "Guikingone (Guillaume Loulier) on Jun 8, 2019 • edited". A table below shows the questions and answers:

Q	A
Branch?	master
Bug fix?	no
New feature?	yes



Discussions starts long time ago

- September 2019, Wouter's 1st PR about redesign of Security

[Security] AuthenticatorManager to make "authenticators" first-class security #33558

Merged by fabpot symfony:master ← wouterj:security/deprecate-providers-listeners on Apr 21, 2020 v5.1.0-BETA1

Conversation 177 Commits 30 Checks 0 Files changed 77

wouterj (Wouter de Jong) on Sep 11, 2019 • edited Member

Q	A
Branch?	master
Bug fix?	no
New feature?	yes
Deprecations?	no
Tickets	-

Reviewers – review now

- javieregulluz
- rosier
- nicolas-grekas
- stof
- nonlagriconomie
- fabpot
- weaverryan
- chalar



Discussions starts long time ago

- April 2020, continuation of Wouter's work on new Security system

[Security] Integrated Guards with the Authenticator system #36570

Merged by fabpot symfony:master ← wouterj:security/new-system-guard on Apr 25, 2020 v6.1.0-BETA1

Conversation 4 Commits 1 Checks 0 Files changed 4

wouterj (Wouter de Jong) on Apr 24, 2020 • edited

Q	A
Branch?	master
Bug fix?	no
New feature?	yes
Deprecations?	no
Tickets	-

Reviewers – review now
fabpot
chalaar

Assignees – assign yours

Labels
Feature Status: Review

Projects

[Security] Removed anonymous in the new security system #36574

Merged by fabpot symfony:master ← wouterj:security/remove-anonymous on May 3, 2020 v6.1.0-BETA1

Conversation 47 Commits 1 Checks 0 Files changed 14

wouterj (Wouter de Jong) on Apr 24, 2020 • edited

Q	A
Branch?	master
Bug fix?	no
New feature?	yes
Deprecations?	no
Tickets	-

Reviewers – review now
javiergulluz
weaverryan
simonberger
fabpot

Assignees – assign yours

Labels
Feature Security



Discussions starts long time ago

- March 2022, Vincent Chalamon opens an issue about "Bearer Authenticator"

[Security] Add Bearer Authenticator #45844

(Closed) 14 comments · Fixed by #46428

vinchalamon (Vincent) on Mar 25 • edited

Description

I'm implementing an OIDC architecture in an API first project, and wanted to secure my API with an authenticator. I first took a look at bundles like [hwiv/oauth-bundle](#) or [knpuniversity/oauth2-client-bundle](#), but they're not adapted to an API as they provide bunch of features not needed in an API, and provides their own JWT generation and storage (e.g.: cookie).

I talked about it with [@dunglas](#) who shares the idea to implement a Bearer authenticator directly in the Symfony Security bundle. The idea behind this authenticator would be to retrieve the token from the `Authorization` header (prefixed with `Bearer`), and validate and decode it in a `BearerToken` using [lcobucci/jwt](#).

Example

```
# config/packages/security.yaml
security:
    firewall:
        main:
            bearer:
                signature: '/path/to/signature/key' # Could be a file path, plaintext, or even an url
                algorithm: 'hmac.sha256'
                key: 'customUserId' # Which key from the JWT should be sent to the UserProvider (default: '')
```

Assignees – assign yourself

Labels

Feature x Security x

Projects

None yet

Milestone

No milestone

Development

Successfully merging a pull request may close this issue.

↳ [Security] Access Token Authenticator
Spomky/symfony

↳ [Security] Implement HttpBearerAuthenticator
vincalachalamon/symfony



Discussions starts long time ago

- Finally, on **21 May 2022**, Vincent Chalamon & Florent Morselli each open a PR to implement the Authenticator

[Security] Implement HttpBearerAuthenticator #46429

Closed [symfony/symfony#46429](#)

Conversation 2 · Commits 1 · Checks 7 · Files changed 14

vinchalamon (Vincent) on May 21

Q	A
Branch?	6.2
Bug fix?	no
New feature?	yes
Deprecations?	no
Tickets	Fix #45844
License	MIT

Reviewers — review now
wouterj
chalars

Assignees — assign yourself
None yet

Labels
Feature Status

Projects

[Security] Access Token Authenticator #46428

Merged [symfony/symfony#46428](#)

Conversation 195 · Commits 1 · Checks 7 · Files changed 39

Spomky (Florent Morselli) on May 21 · edited

Q	A
Branch?	6.2
Bug fix?	yes
New feature?	yes
Deprecations?	no
Tickets	Fix #45844
License	MIT
Doc PH	symfony/symfony-docs#16819

Reviewers — review now
vincentchalamon
yrtxx
stof
34noni
derrabus
dunglas
wouterj
chalars

Assignees — assign yourself
None yet

Labels
Bug Status Ready Security

Status: Reviewed

Projects

Hi,
This PR aims at fixing [#45844](#).
It adds a new authenticator that is able to fetch a token in the request header and retrieve the associated user identifier.
The authenticator delegates the token loading to a handler. This handler could manage opaque tokens (random strings stored in a database) or self-contained tokens such as JWT, Paseto, SAML...



Thanks to Florent Morselli @Spomky



In Symfony 6.2 we won't have to worry about this code anymore!

[Security] Access Token Authenticator #46428

Merged by chalasr symfony:6.2 ← Spomky:features/access-token-authenticator on Aug 10

Conversation 195 Commits 1 Checks 7 Files changed 38 +1,994 -1

Spomky (Florent Morselli) on May 21 · edited

Q	A
Branch?	6.2
Bug fix?	yes
New feature?	yes
Deprecations?	no
Tickets	Fix #45844
License	MIT
Doc PR	symfony/symfony-docs#16819

Hi,

This PR aims at fixing [#45844](#).
It adds a new authenticator that is able to fetch a token in the request header and retrieve the associated user identifier.

The authenticator delegates the token loading to a handler. This handler could manage opaque tokens (random strings stored in a database) or self-contained tokens such as JWT, Paseto, SAML...

Reviewers – review now

- vincentchalamon
- lyrixx
- stof
- 94noni
- derrabus
- dunglas
- wouterj
- chalasr

Assignees – assign yourself

Labels

- Bug
- Feature
- Ready
- Security

Status: Reviewed

Projects

None yet



Let's meet AccessTokenAuthenticator in Symfony 6.2

AccessTokenAuthenticator

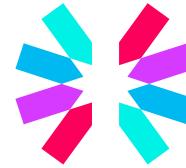
- Takes care of token extraction
 - Header
 - Query string
 - Request body
 - Or Custom extractors
- Calls **your** Token Handler to check the token (revocation, expiration, signature, ...)
- Custom success / failure handlers if needed



All this via configuration!

What kind of tokens are supported?

- JWT
- SAML2
- Biscuit
- Macaroons
- Homemade tokens
- ...
- Any kind of token, as it's up to you to handle them



How much easier is it than in 6.1?

6.1 security.yaml

```
1  security:
2    providers:
3      client_app_provider:
4        entity:
5          class: App\Entity\App
6          property: apiKey
7    firewalls:
8      api:
9        pattern: ^/api
10        lazy: true
11        provider: client_app_provider
12        custom_authenticator: App\Security\ApiKeyAuther
```

6.2 security.yaml

```
1  security:
2    providers:
3      client_app_provider:
4        entity:
5          class: App\Entity\App
6          property: apiKey
7    firewalls:
8      api:
9        pattern: ^/api
10        lazy: true
11        provider: client_app_provider
12        access_token:
13          token_extractors:
14            - 'security.access_token_extractor.query_st
15            - 'security.access_token_extractor.request_
16            - 'security.access_token_extractor.header'
17        token_handler: App\Security\AccessTokenHandle
```

```
1  /** 6.1 **/
2  class ApiKeyAuthenticator extends AbstractAuthenticator
3  {
4      public function __construct(
5          private readonly AppRepository $repository,
6      ) {}
7
8      public function supports(Request $request): ?bool
9      {
10         return $request->headers->has('Authorization')
11             || $request->query->has('access_token')
12             || $request->request->has('access_token');
13     }
14
15     public function authenticate(Request $request): Passport
16     {
17         $apiKey = $this->extractToken($request);
18
19         if (null === $apiKey) {
20             throw new AuthenticationException('No API Key provided');
21         }
22
23         // In case of JWT, decode it here
24
25         return new SelfValidatingPassport(
26             new UserBadge(
27                 $apiKey,
28                 function (string $apiKey) {
29                     return $this->repository->findOneBy(['apiKey' => $apiKey, 'revoked' => false]);
30                 }
31         );
32     }
33 }
```

```
1  /** 6.1 */
2  class ApiKeyAuthenticator extends AbstractAuthenticator
3  {
4      public function __construct(
5          private readonly AppRepository $repository,
6      ) {}
7
8      public function supports(Request $request): ?bool
9      {
10         return $request->headers->has('Authorization')
11             || $request->query->has('access_token')
12             || $request->request->has('access_token');
13     }
14
15     public function authenticate(Request $request): Passport
16     {
17         $apiKey = $this->extractToken($request);
18
19         if (null === $apiKey) {
20             throw new AuthenticationException('No API Key provided');
21         }
22
23         // In case of JWT, decode it here
24
25         return new SelfValidatingPassport(
26             new UserBadge(
27                 $apiKey,
28                 function (string $apiKey) {
29                     return $this->repository->findOneBy(['apiKey' => $apiKey, 'revoked' => false]);
30                 }
31         );
32     }
33 }
```



```
1  /** 6.1 */
2  class ApiKeyAuthenticator extends AbstractAuthenticator
3  {
4      public function __construct(
5          private readonly AppRepository $repository,
6      ) {}
7
8      public function supports(Request $request): ?bool
9      {
10         return $request->headers->has('Authorization')
11             || $request->query->has('access_token')
12             || $request->request->has('access_token');
13     }
14
15     public function authenticate(Request $request): Passport
16     {
17         $apiKey = $this->extractToken($request);
18
19         if (null === $apiKey) {
20             throw new AuthenticationException('No API Key provided');
21         }
22
23         // In case of JWT, decode it here
24
25         return new SelfValidatingPassport(
26             new UserBadge(
27                 $apiKey,
28                 function (string $apiKey) {
29                     return $this->repository->findOneBy(['apiKey' => $apiKey, 'revoked' => false]);
30                 }
31         );
32     }
33 }
```

AccessTokenHandler 6.2

```
1  class AccessTokenHandler implements AccessTokenHandlerInterface
2  {
3      public function __construct(
4          private readonly AppRepository $repository,
5      ) {}
6
7      public function getUserIdentifierFrom(string $token): string
8      {
9          // In case of JWT, decode it here
10
11         $app = $this->repository->findOneBy(['apiKey' => $token]);
12
13         if ($app === null || $app->isRevoked()) {
14             throw new BadCredentialsException('Invalid credentials.');
15         }
16
17         return $token;
18     }
19 }
```

TODO

Thank you



Any questions?

qrcode, slide link etc