Using the SecurityBundle in Symfony 6

About me...

Symfony Documentor

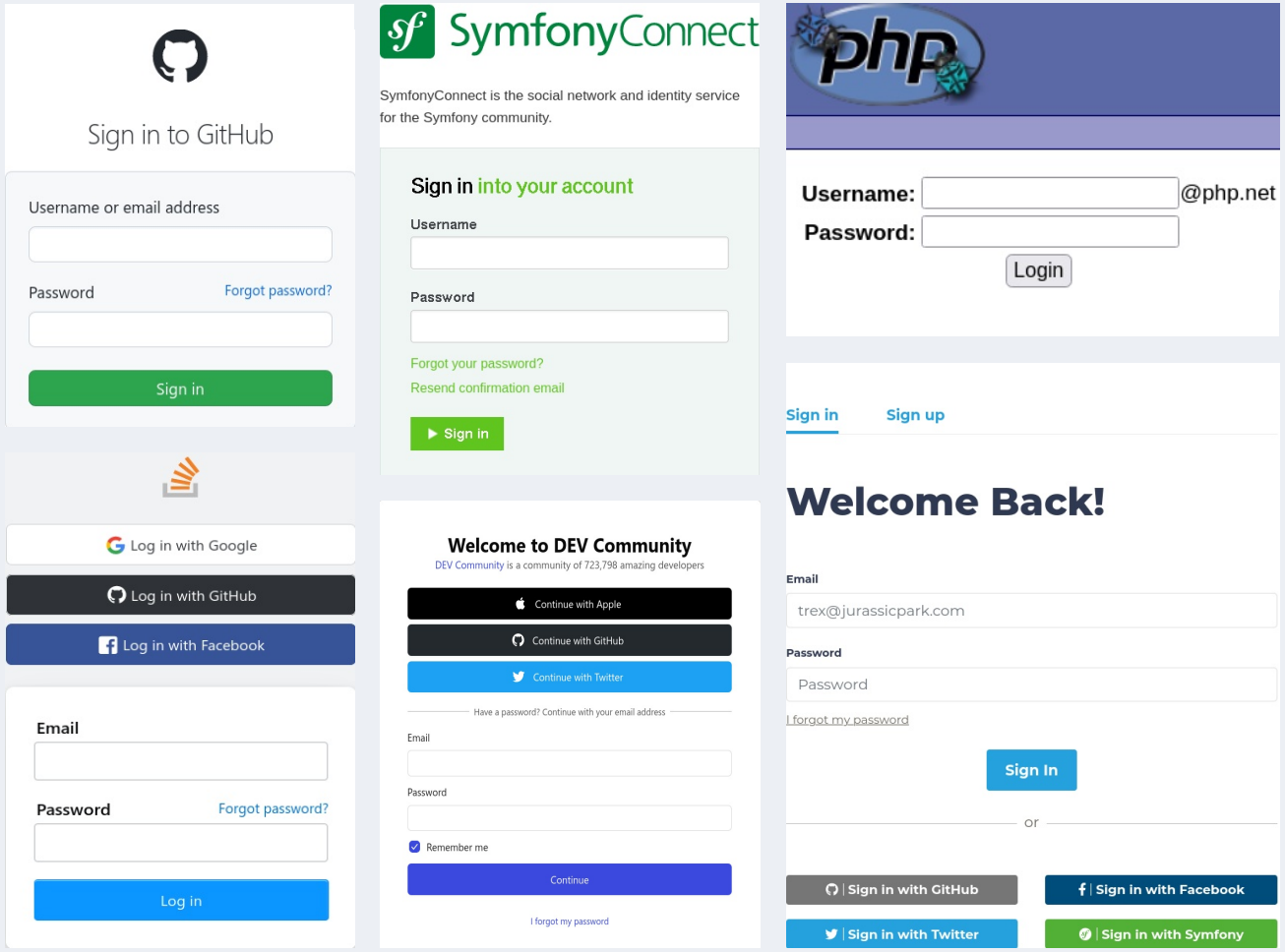
Symfony Core team member

Doing dev things at Amber

Modernizing Security

Security is boring

Everyone uses the same things



Security is boring,

reuse as much code as possible

Use the makers

```
$ composer require --dev symfony/maker-bundle
```


https://symfony.com/bundles/SymfonyMakerBundle/current/index.html

```
$ symfony console make:user
$ symfony console make:migration
$ symfony console doctrine:migrate:migrate
```

```
$ symfony console make:user
$ symfony console make:migration
$ symfony console doctrine:migrate:migrate
```

```
$ bin/console make:user
$ bin/console make:migration
$ bin/console doctrine:migrate:migrate
```


https://symfony.com/doc/current/setup/symfony_server.html

```
$ symfony console make:user
$ symfony console make:migration
$ symfony console doctrine:migrate:migrate
```


https://symfony.com/doc/current/setup/symfony_server.html

QUICK TIP

Symfony 6 comes with a *clean* UserInterface

```
interface UserInterface
{
    public function getUsername();

    public function getRoles(): array;

    public function eraseCredentials();

    public function getPassword();

    public function getSalt();
}
```

https://symfony.com/doc/current/security/#the-user

QUICK TIP

Symfony 6 comes with a *clean* UserInterface

```
$ symf
$ sym
$ sy
```

```
  interface UserInterface
  {
-     public function getUsername();
+     public function getUserIdentifier(): string;

      public function getRoles(): array;

      public function eraseCredentials();

      public function getPassword();

      public function getSalt();
  }
```

QUICK TIP

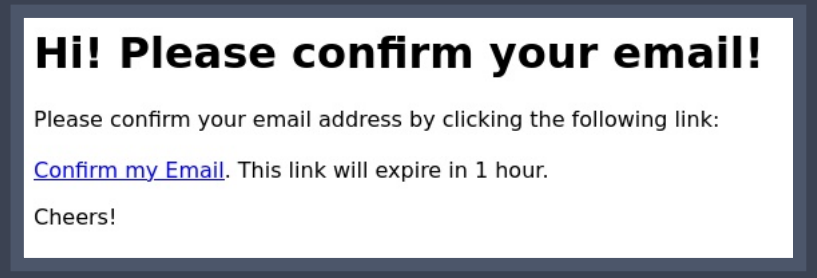
Symfony 6 comes with a *clean* UserInterface

```
  interface UserInterface
  {
-     public function getUsername();
+     public function getUserIdentifier(): string;

      public function getRoles(): array;

      public function eraseCredentials();

      // PasswordAuthenticatedUserInterface
-     public function getPassword();

      // LegacyPasswordAuthenticatedUserInterface
-     public function getSalt();
  }
```

```
$ symf
$ sym
$ sy
```

```
$ symfony console make:user
$ symfony console make:migration
$ symfony console doctrine:migrate:migrate

$ symfony console make:registration-form
```

```
$ symfony console make:user
$ symfony console make:migration
$ symfony console doctrine:migrate:migrate

$ symfony console make:registration-form
```



https://github.com/SymfonyCasts/verify-email-bundle

```
$ symfony console make:user
$ symfony console make:migration
$ symfony console doctrine:migrate:migrate

$ symfony console make:registration-form

$ symfony console make:reset-password
```


https://github.com/SymfonyCasts/reset-password-bundle

```
$ symfony console make:user
$ symfony console make:migration
$ symfony console doctrine:migrate:migrate

$ symfony console make:registration-form

$ symfony console make:reset-password
```

Security is boring,

use the built-in authenticators

Security is critical,

use the wisdom of the crowd

A login form is complicated...

CVE-2013-5958: Denial-of-service via large passwords

October 10, 2013 Fabien Potencier

CVE-2016-4423: Large username storage in session

May 9, 2016 Fabien Potencier

CVE-2017-16652: Open redirect vulnerability on security handlers

November 17, 2017 Fabien Potencier

CVE-2018-11385: Session Fixation Issue for Guard Authentication

May 25, 2018 Fabien Potencier

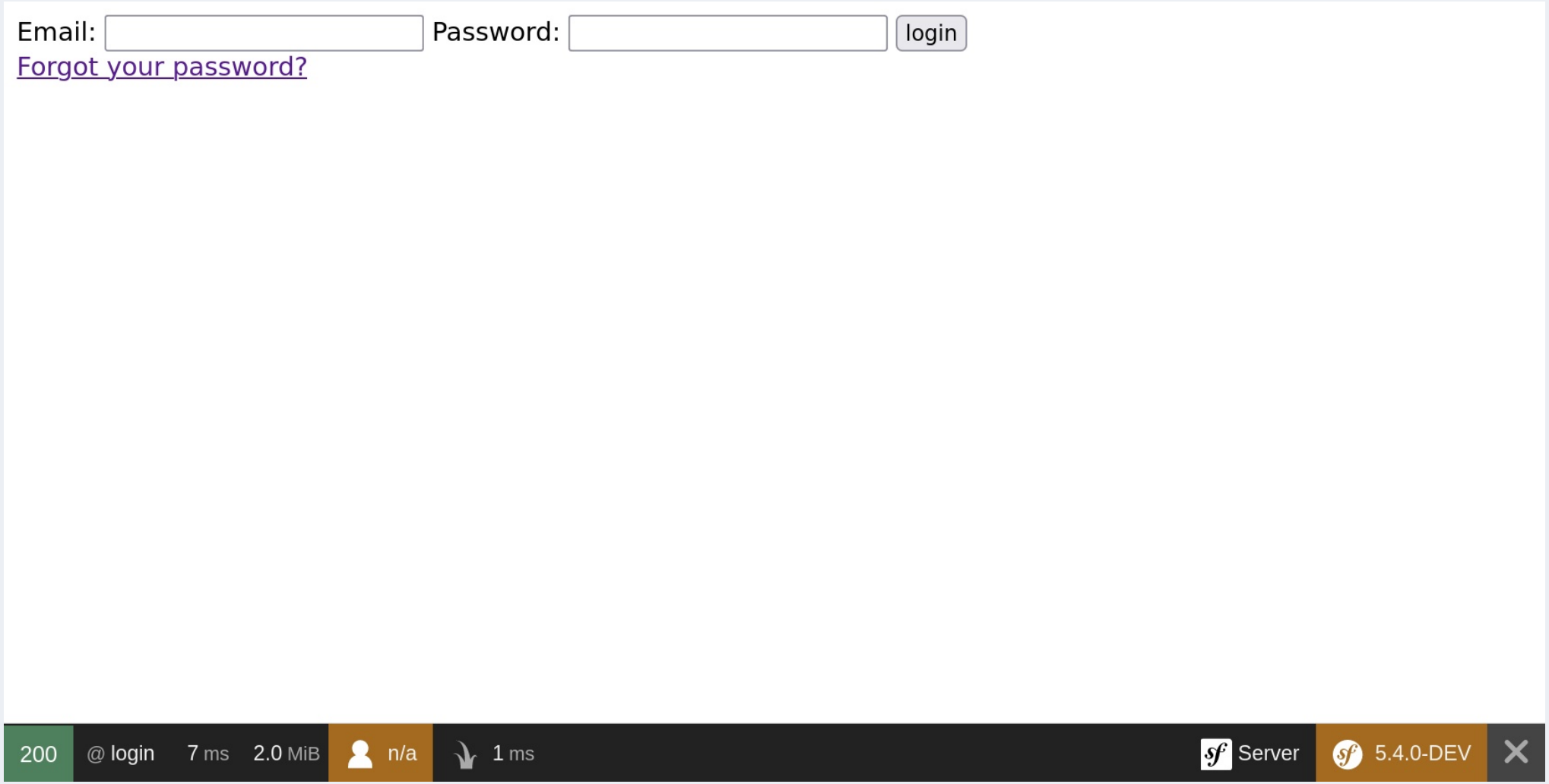
```yaml
# config/packages/security.yaml
security:
    # ...

    firewalls:
        main:
            form_login:
                login_path: 'login'
                check_path: 'login'
```

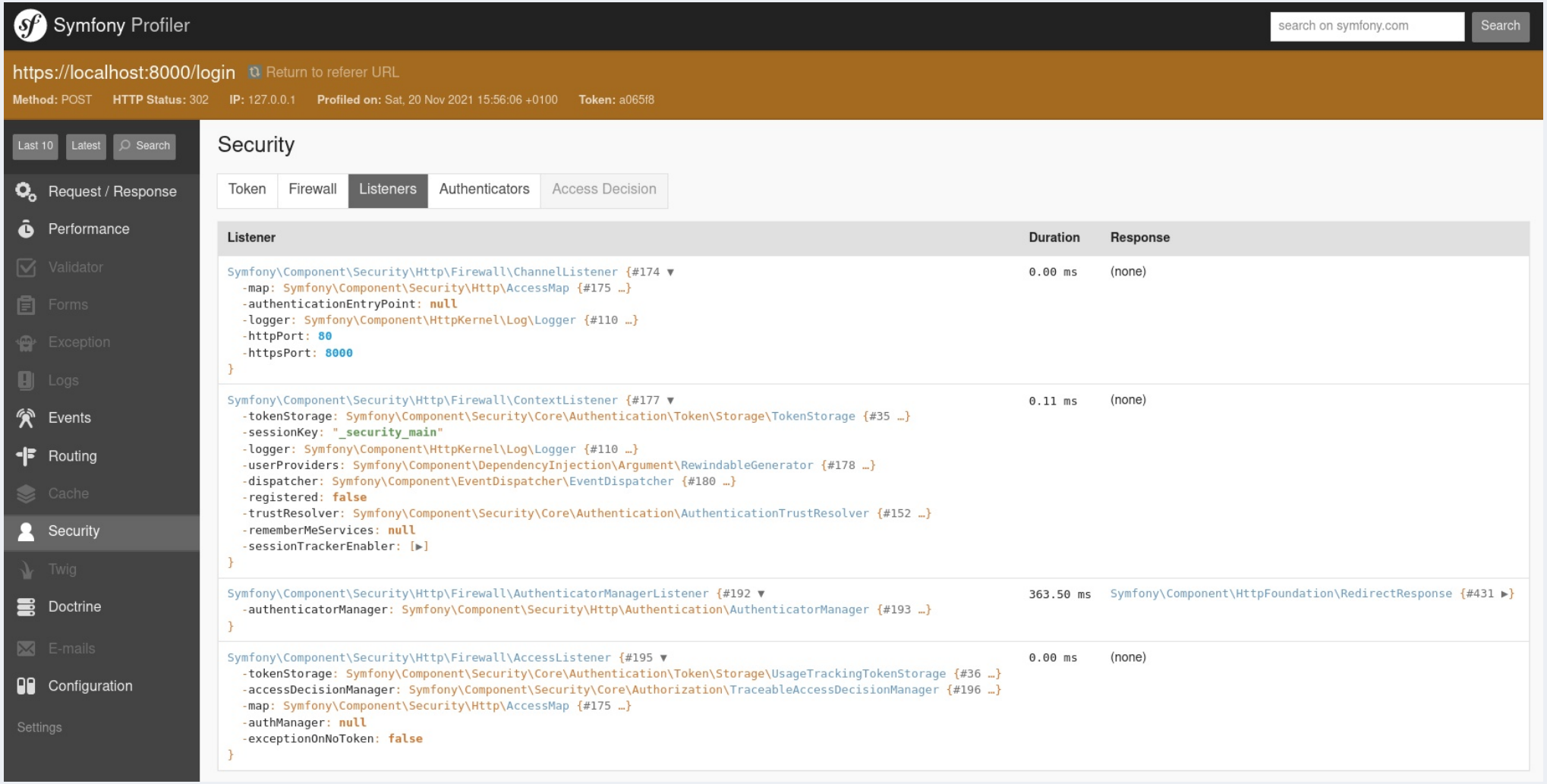

https://symfony.com/doc/current/security/#form-login

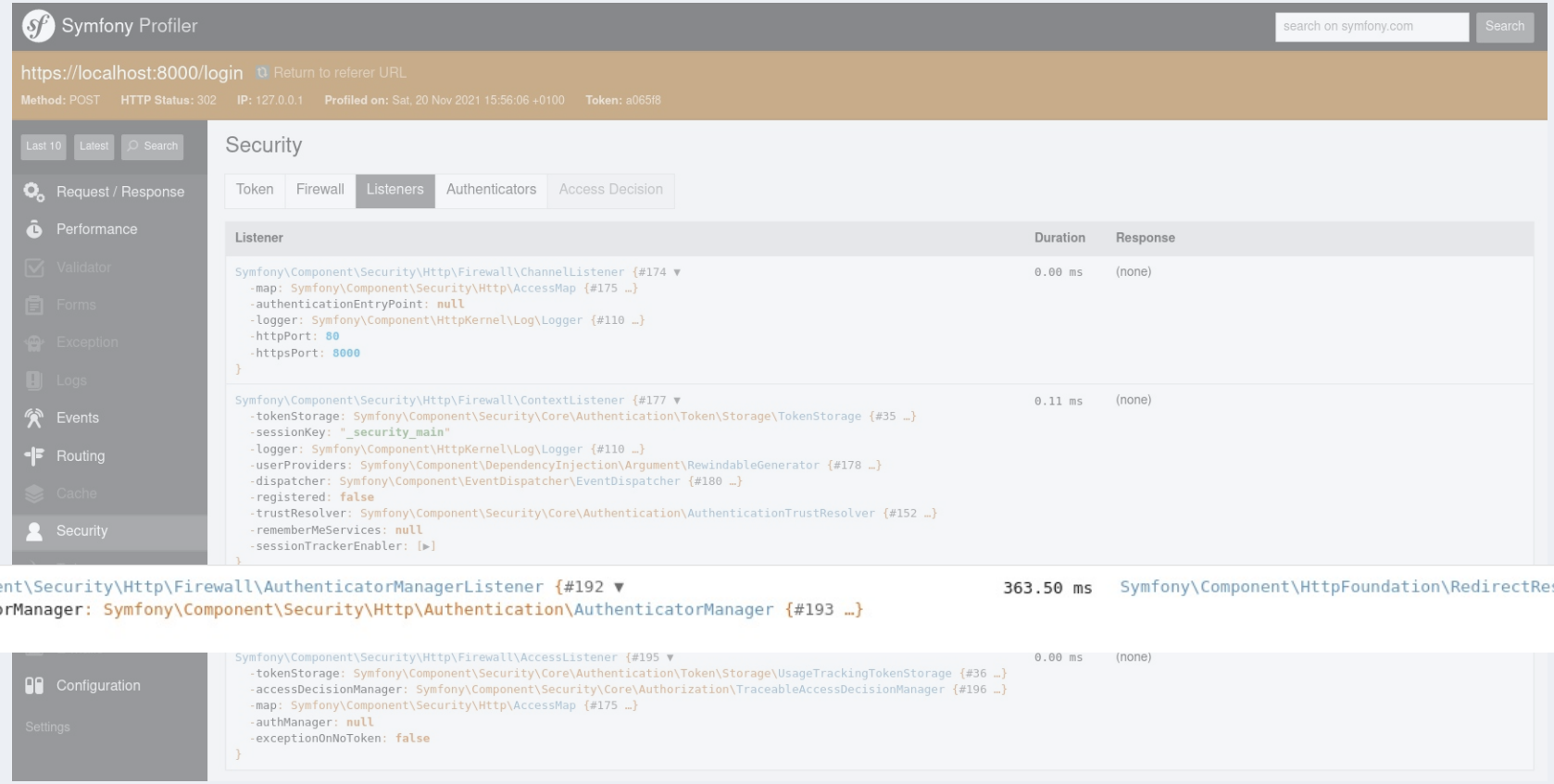
```twig
{# templates/login.html.twig #}
<form action="{{ path('login') }}" method="post">
    <label for="username">
        Email:
        <input type="text" name="_username" value="{{ last_username }}">
    </label>

    <label for="password">
        Password:
        <input type="password" id="password" name="_password">
    </label>

    <button type="submit">login</button>
</form>

<a href="{{ path("app_forgot_password_request") }}">Forgot your password?</a>
```

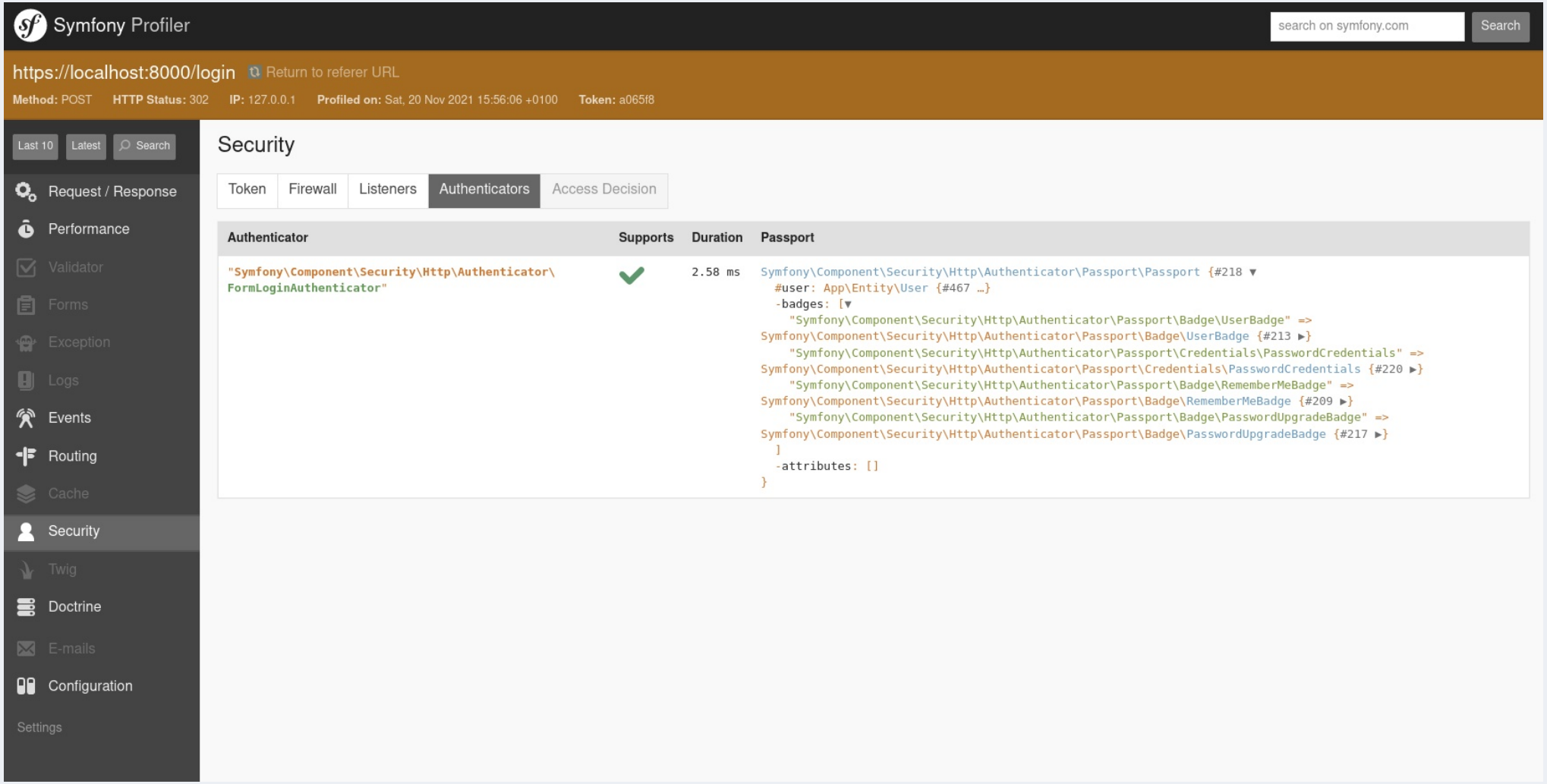
```twig
{# templates/login.html.twig #}
<form action="{{ path('login') }}" method="post">
    <label for="username">
        Email:
        <input type="text" name="_username" value="{{ last_username }}">
    </label>

    <label for="password">
        Password:
        <input type="password" id="password" name="_password">
    </label>

    <button type="submit">login</button>
</form>

<a href="{{ path("app_forgot_password_request") }}">Forgot your password?</a>
```

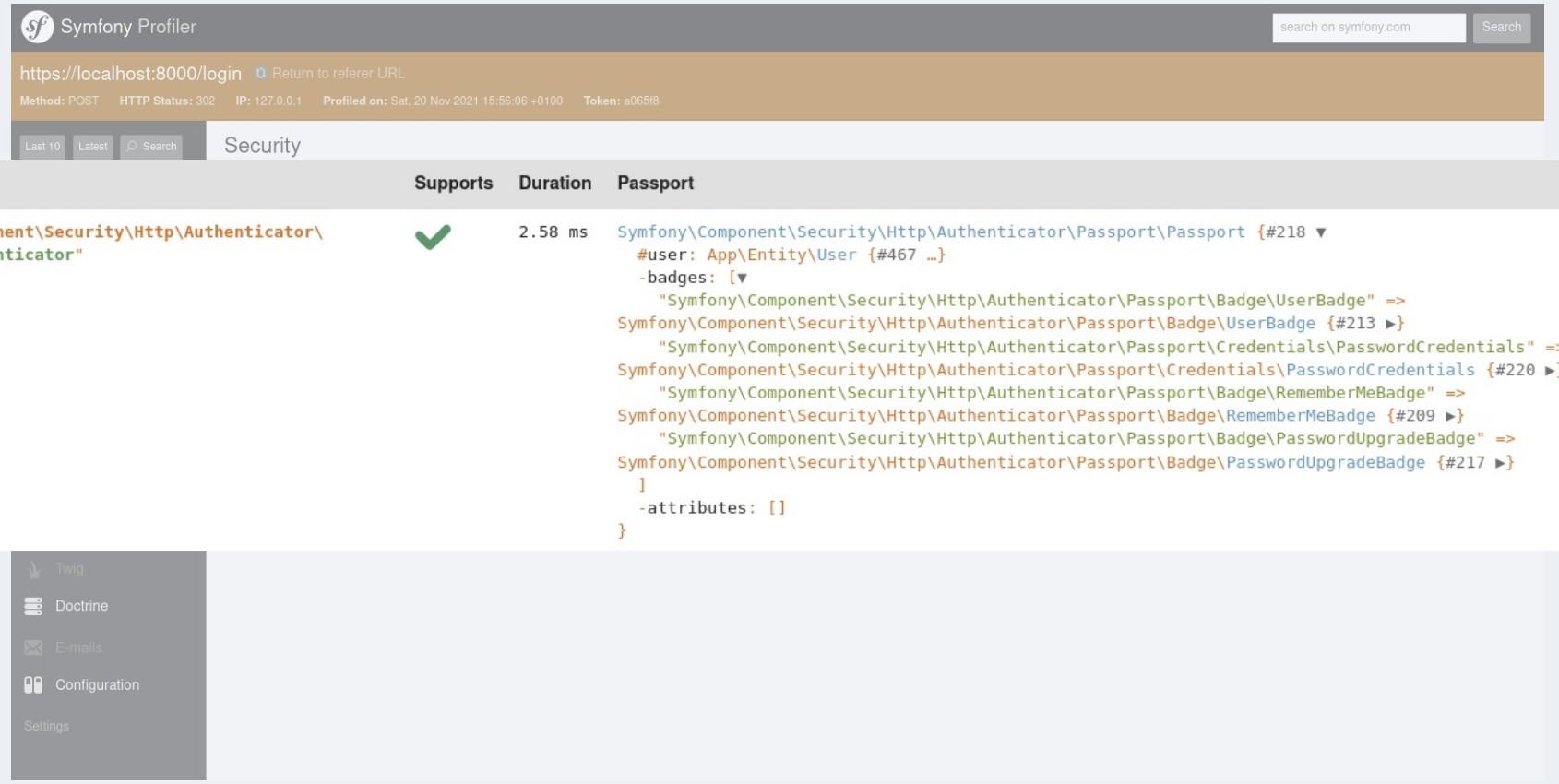
```twig
{# templates/login.html.twig #}
<form action="{{ path('login') }}" method="post">
    <label for="username">
        Email:
        <input type="text" name="_username" value="{{ last_username }}">
    </label>

    <label for="password">
        Password:
        <input type="password" id="password" name="_password">
    </label>

    <button type="submit">login</button>
</form>

<a href="{{ path("app_forgot_password_request") }}">Forgot your password?</a>
```


Email:
Password:
login
Forgot your password?
200
@ login
7 ms
2.0 MiB
n/a
1 ms
Server
5.4.0-DEV


Symfony Profiler
search on symfony.com
Search
https://localhost:8000/login
Return to referer URL
Method: POST
HTTP Status: 302
IP: 127.0.0.1
Profiled on: Sat, 20 Nov 2021 15:56:06 +0100
Token: a065f8
Last 10
Latest
Search
Request / Response
Performance
Validator
Forms
Exception
Logs
Events
Routing
Cache
Security
Twig
Doctrine
E-mails
Configuration
Settings
Security
Token
Firewall
Listeners
Authenticators
Access Decision
Listener
Duration
Response
Symfony\Component\Security\Http\Firewall\ChannelListener {#174 ▼
-map: Symfony\Component\Security\Http\AccessMap {#175 …}
-authenticationEntryPoint: null
-logger: Symfony\Component\HttpKernel\Log\Logger {#110 …}
-httpPort: 80
-httpsPort: 8000
}
0.00 ms
(none)
Symfony\Component\Security\Http\Firewall\ContextListener {#177 ▼
-tokenStorage: Symfony\Component\Security\Core\Authentication\Token\Storage\TokenStorage {#35 …}
-sessionKey: "_security_main"
-logger: Symfony\Component\HttpKernel\Log\Logger {#110 …}
-userProviders: Symfony\Component\DependencyInjection\Argument\RewindableGenerator {#178 …}
-dispatcher: Symfony\Component\EventDispatcher\EventDispatcher {#180 …}
-registered: false
-trustResolver: Symfony\Component\Security\Core\Authentication\AuthenticationTrustResolver {#152 …}
-rememberMeServices: null
-sessionTrackerEnabler: [▶]
}
0.11 ms
(none)
Symfony\Component\Security\Http\Firewall\AuthenticatorManagerListener {#192 ▼
-authenticatorManager: Symfony\Component\Security\Http\Authentication\AuthenticatorManager {#193 …}
}
363.50 ms
Symfony\Component\HttpFoundation\RedirectResponse {#431 ▶}
Symfony\Component\Security\Http\Firewall\AccessListener {#195 ▼
-tokenStorage: Symfony\Component\Security\Core\Authentication\Token\Storage\UsageTrackingTokenStorage {#36 …}
-accessDecisionManager: Symfony\Component\Security\Core\Authorization\TraceableAccessDecisionManager {#196 …}
-map: Symfony\Component\Security\Http\AccessMap {#175 …}
-authManager: null
-exceptionOnNoToken: false
}
0.00 ms
(none)


Symfony Profiler
search on symfony.com
Search
https://localhost:8000/login
Return to referer URL
Method: POST HTTP Status: 302 IP: 127.0.0.1 Profiled on: Sat, 20 Nov 2021 15:56:06 +0100 Token: a065f8
Last 10
Latest
Search
Request / Response
Performance
Validator
Forms
Exception
Logs
Events
Routing
Cache
Security
Configuration
Settings
Security
Token
Firewall
Listeners
Authenticators
Access Decision
Listener
Duration
Response
Symfony\Component\Security\Http\Firewall\ChannelListener {#174 ▼
-map: Symfony\Component\Security\Http\AccessMap {#175 …}
-authenticationEntryPoint: null
-logger: Symfony\Component\HttpKernel\Log\Logger {#110 …}
-httpPort: 80
-httpsPort: 8000
}
0.00 ms
(none)
Symfony\Component\Security\Http\Firewall\ContextListener {#177 ▼
-tokenStorage: Symfony\Component\Security\Core\Authentication\Token\Storage\TokenStorage {#35 …}
-sessionKey: "_security_main"
-logger: Symfony\Component\HttpKernel\Log\Logger {#110 …}
-userProviders: Symfony\Component\DependencyInjection\Argument\RewindableGenerator {#178 …}
-dispatcher: Symfony\Component\EventDispatcher\EventDispatcher {#180 …}
-registered: false
-trustResolver: Symfony\Component\Security\Core\Authentication\AuthenticationTrustResolver {#152 …}
-rememberMeServices: null
-sessionTrackerEnabler: [▶]
}
0.11 ms
(none)
Symfony\Component\Security\Http\Firewall\AuthenticatorManagerListener {#192 ▼
-authenticatorManager: Symfony\Component\Security\Http\Authentication\AuthenticatorManager {#193 …}
}
363.50 ms
Symfony\Component\HttpFoundation\RedirectResponse {#431 ▶}
Symfony\Component\Security\Http\Firewall\AccessListener {#195 ▼
-tokenStorage: Symfony\Component\Security\Core\Authentication\Token\Storage\UsageTrackingTokenStorage {#36 …}
-accessDecisionManager: Symfony\Component\Security\Core\Authorization\TraceableAccessDecisionManager {#196 …}
-map: Symfony\Component\Security\Http\AccessMap {#175 …}
-authManager: null
-exceptionOnNoToken: false
}
0.00 ms
(none)


Symfony Profiler
search on symfony.com
Search
https://localhost:8000/login
Return to referer URL
Method: POST
HTTP Status: 302
IP: 127.0.0.1
Profiled on: Sat, 20 Nov 2021 15:56:06 +0100
Token: a065f8
Last 10
Latest
Search
Request / Response
Performance
Validator
Forms
Exception
Logs
Events
Routing
Cache
Security
Twig
Doctrine
E-mails
Configuration
Settings
Security
Token
Firewall
Listeners
Authenticators
Access Decision
Authenticator
Supports
Duration
Passport
"Symfony\Component\Security\Http\Authenticator\FormLoginAuthenticator"
2.58 ms
Symfony\Component\Security\Http\Authenticator\Passport\Passport {#218 ▼
#user: App\Entity\User {#467 …}
-badges: [▼
"Symfony\Component\Security\Http\Authenticator\Passport\Badge\UserBadge" => Symfony\Component\Security\Http\Authenticator\Passport\Badge\UserBadge {#213 ▶}
"Symfony\Component\Security\Http\Authenticator\Passport\Credentials\PasswordCredentials" => Symfony\Component\Security\Http\Authenticator\Passport\Credentials\PasswordCredentials {#220 ▶}
"Symfony\Component\Security\Http\Authenticator\Passport\Badge\RememberMeBadge" => Symfony\Component\Security\Http\Authenticator\Passport\Badge\RememberMeBadge {#209 ▶}
"Symfony\Component\Security\Http\Authenticator\Passport\Badge\PasswordUpgradeBadge" => Symfony\Component\Security\Http\Authenticator\Passport\Badge\PasswordUpgradeBadge {#217 ▶}
]
-attributes: []
}


Symfony Profiler
search on symfony.com
Search
https://localhost:8000/login
Return to referer URL
Method: POST
HTTP Status: 302
IP: 127.0.0.1
Profiled on: Sat, 20 Nov 2021 15:56:06 +0100
Token: a065f8
Last 10
Latest
Search
Security
Authenticator
Supports
Duration
Passport
"Symfony\Component\Security\Http\Authenticator\FormLoginAuthenticator"
2.58 ms
Symfony\Component\Security\Http\Authenticator\Passport\Passport {#218 ▼
#user: App\Entity\User {#467 …}
-badges: [▼
"Symfony\Component\Security\Http\Authenticator\Passport\Badge\UserBadge" => Symfony\Component\Security\Http\Authenticator\Passport\Badge\UserBadge {#213 ▶}
"Symfony\Component\Security\Http\Authenticator\Passport\Credentials\PasswordCredentials" => Symfony\Component\Security\Http\Authenticator\Passport\Credentials\PasswordCredentials {#220 ▶}
"Symfony\Component\Security\Http\Authenticator\Passport\Badge\RememberMeBadge" => Symfony\Component\Security\Http\Authenticator\Passport\Badge\RememberMeBadge {#209 ▶}
"Symfony\Component\Security\Http\Authenticator\Passport\Badge\PasswordUpgradeBadge" => Symfony\Component\Security\Http\Authenticator\Passport\Badge\PasswordUpgradeBadge {#217 ▶}
]
-attributes: []
}
Twig
Doctrine
E-mails
Configuration
Settings

Authenticators in Symfony 6

Form login

JSON login

HTTP Basic

Login Links (aka Magic Links)

X509 Client Certificates

Remote users (e.g. Kerebos)

https://symfony.com/doc/current/security.html#authenticating-users

...and in community bundles

LexikJWTAuthenticationBundle

OneloginSamlBundle

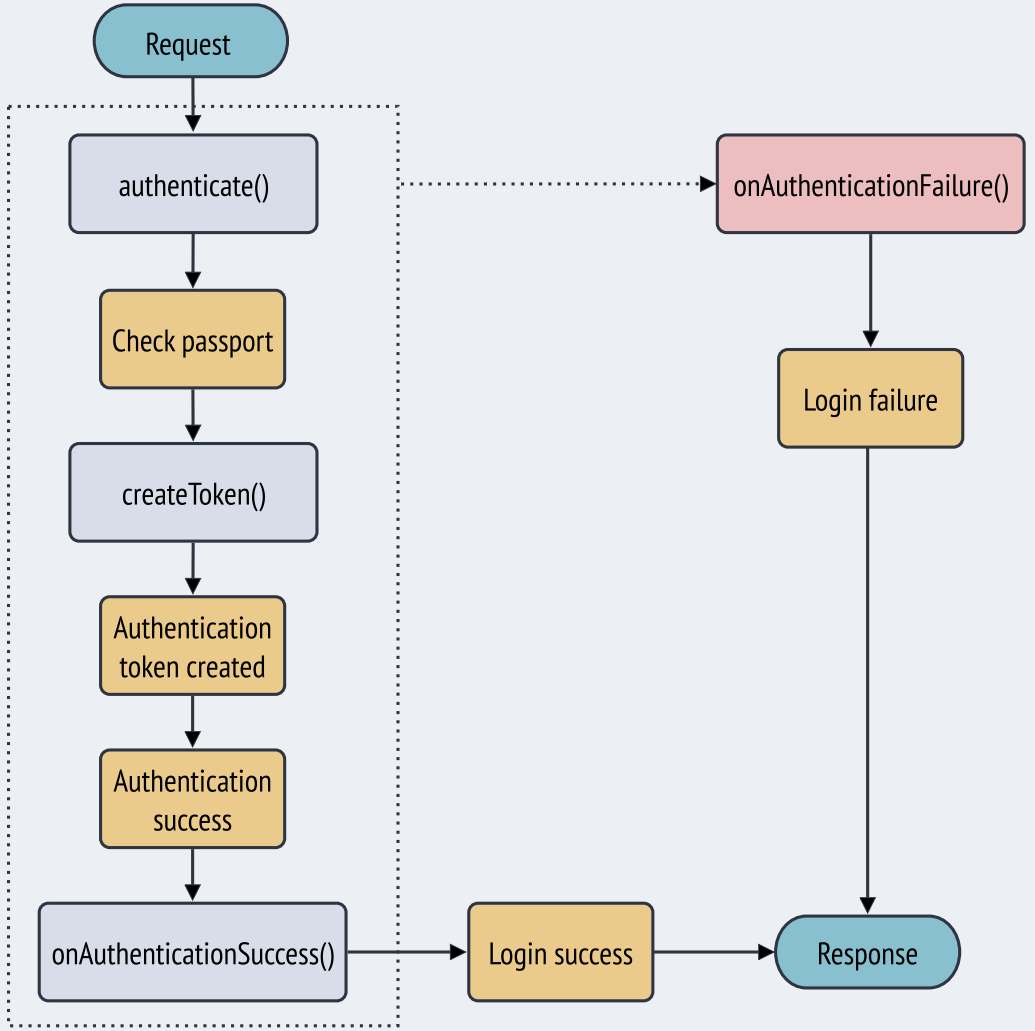
WebauthnSymfonyBundle

KnpUOAuth2ClientBundle

SchebTwoFactorBundle

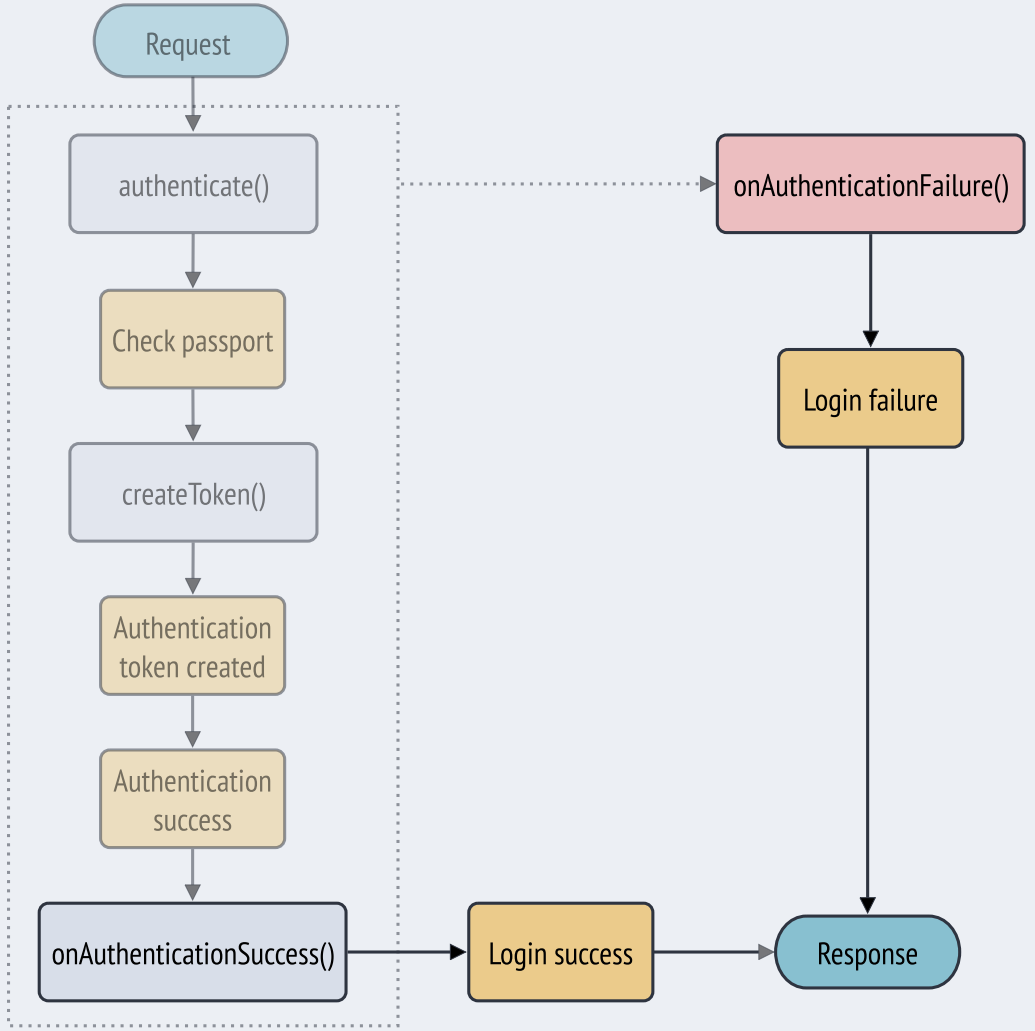
Customizing authentication

Security events cycle



https://symfony.com/doc/current/security/#security-events

Security events cycle



https://symfony.com/doc/current/security/#security-events

```php
use Symfony\Component\EventDispatcher\Attribute\AsEventListener;
use Symfony\Component\Security\Http\Event\LoginFailureEvent;
use Symfony\Component\Security\Http\Event\LoginSuccessEvent;

class LoginAttemptListener
{
    #[AsEventListener(event: LoginSuccessEvent::class)]
    #[AsEventListener(event: LoginFailureEvent::class)]
    public function onLoginAttempt(LoginSuccessEvent|LoginFailureEvent $event)
    {
        // ...
    }
}
```

```php
use Symfony\Component\EventDispatcher\Attribute\AsEventListener;
use Symfony\Component\Security\Http\Event\LoginFailureEvent;
use Symfony\Component\Security\Http\Event\LoginSuccessEvent;

class LoginAttemptListener
{
    #[AsEventListener(event: LoginSuccessEvent::class)]
    #[AsEventListener(event: LoginFailureEvent::class)]
    public function onLoginAttempt(LoginSuccessEvent|LoginFailureEvent $event)
    {
        // ...
    }
}
```

```php
// ...
public function onLoginAttempt(LoginSuccessEvent|LoginFailureEvent $event)
{
    $userIdentifier = $event->getPassport()
        ->getBadge(UserBadge::class)->getUserIdentifier();









}
```

```php
// ...
public function onLoginAttempt(LoginSuccessEvent|LoginFailureEvent $event)
{
    $userIdentifier = $event->getPassport()
        ->getBadge(UserBadge::class)->getUserIdentifier();
    $clientIp = $event->getRequest()->getClientIp();









}
```

```php
// ...
public function onLoginAttempt(LoginSuccessEvent|LoginFailureEvent $event)
{
    $userIdentifier = $event->getPassport()
        ->getBadge(UserBadge::class)->getUserIdentifier();
    $clientIp = $event->getRequest()->getClientIp();
    $success = $event instanceof LoginSuccessEvent;









}
```

```php
// ...
public function onLoginAttempt(LoginSuccessEvent|LoginFailureEvent $event)
{
    $userIdentifier = $event->getPassport()
        ->getBadge(UserBadge::class)->getUserIdentifier();
    $clientIp = $event->getRequest()->getClientIp();
    $success = $event instanceof LoginSuccessEvent;

    $this->logger->info('Login attempt for "{user}"', [
        'user' => $userIdentifier,
        'ip' => $clientIp,
        'success' => $success,
        'error' => $success ? $event->getException()->getMessage() : null,
    ]);
}
```

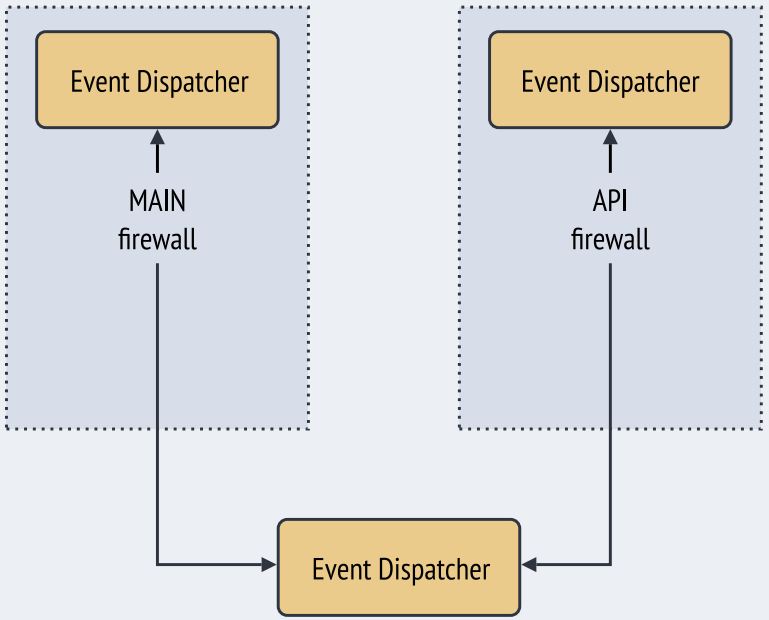
```php
// ...
public function onLoginAttempt(LoginSuccessEvent|LoginFailureEvent $event)
{
    $userIdentifier = $event->getPassport()
        ->getBadge(UserBadge::class)->getUserIdentifier();
    $clientIp = $event->getRequest()->getClientIp();
    $success = $event instanceof LoginSuccessEvent;

    $this->logger->info('Login attempt for "{user}"', [
        'user' => $userIdentifier,
        'ip' => $clientIp,
        'success' => $success,
        'error' => $success ? $event->getException()->getMessage() : null,
    ]);
}
```

```yaml
# config/packages/security.yaml
security:
    # ...

    firewalls:
        api:
            pattern: '^/api'
            jwt: null
            # ...

        main:
            form_login:
                # ...
```


https://github.com/lexik/LexikJWTAuthenticationBundle

Firewall-specific event dispatchers

Firewall event dispatchers



```
security.event_dispatcher.<FIREWALL NAME>
```

```
security.event_dispatcher.main
```

```php
class LoginAttemptListener
{
    #[AsEventListener(
        event: LoginSuccessEvent::class,
        dispatcher: 'security.event_dispatcher.main'
    )]
    #[AsEventListener(
        event: LoginFailureEvent::class,
        dispatcher: 'security.event_dispatcher.main'
    )]
    public function onLoginAttempt(LoginSuccessEvent|LoginFailureEvent $event)
    {
        // ...
    }
}
```

```php
class LoginAttemptListener
{
    #[AsEventListener(
        event: LoginSuccessEvent::class,
        dispatcher: 'security.event_dispatcher.main'
    )]
    #[AsEventListener(
        event: LoginFailureEvent::class,
        dispatcher: 'security.event_dispatcher.main'
    )]
    public function onLoginAttempt(LoginSuccessEvent|LoginFailureEvent $event)
    {
        // ...
    }
}
```

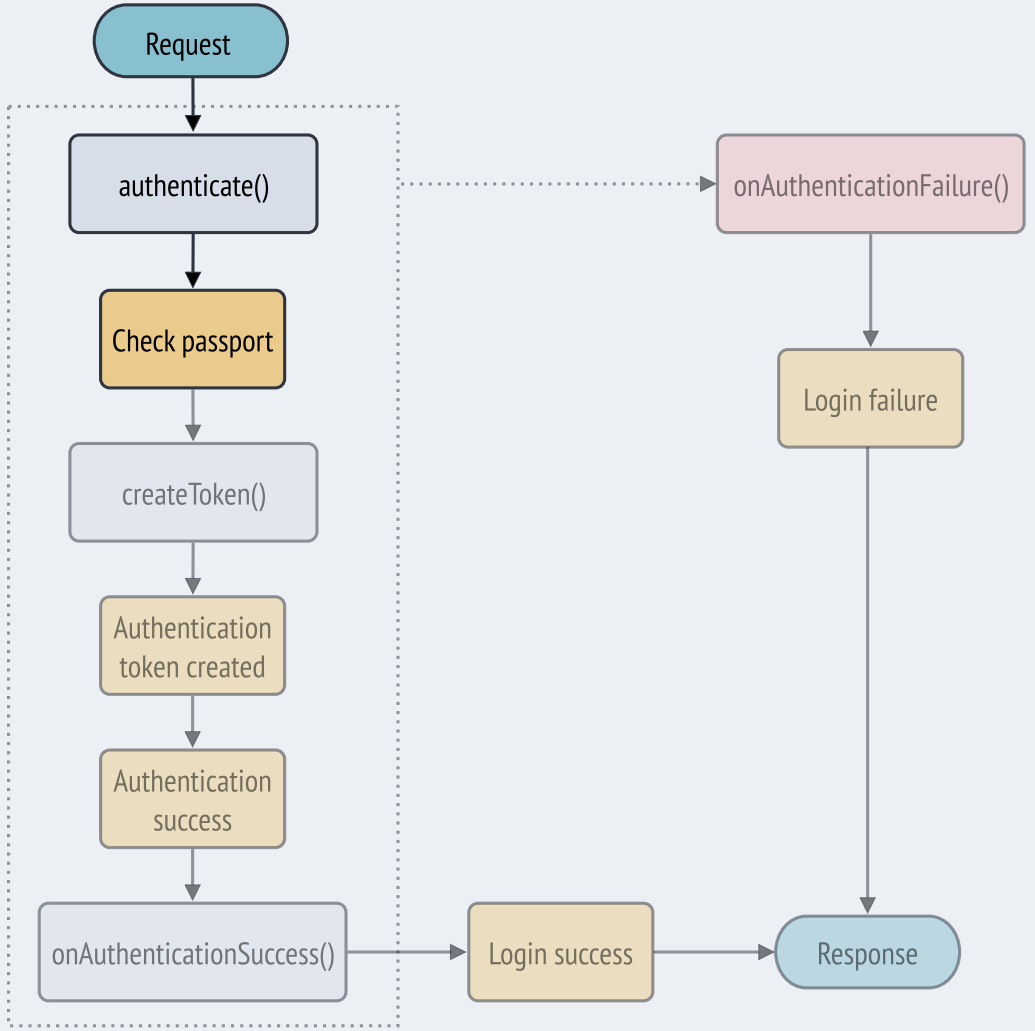
```
$ symfony console debug:firewall --events main
```

```
$ symfony console debug:firewall --events main
...

Event listeners for firewall "main"
===================================

"Symfony\Component\Security\Http\Event\LoginSuccessEvent" event
---------------------------------------------------------------


------------------------------------------------------------------------ -------
 Callable                                                                Prio
------------------------------------------------------------------------ -------
Symfony\Component\Security\...\SessionStrategyListener::onSuccessfulLogin()    0
App\EventSubscriber\LoginAttemptListener::onLoginAttempt()                     0
Symfony\Component\Security\...\PasswordMigratingListener::onLoginSuccess()     0
------------------------------------------------------------------------ -------
```

```
$ symfony console debug:firewall --events main
...

Event listeners for firewall "main"
===================================

"Symfony\Component\Security\Http\Event\LoginSuccessEvent" event
---------------------------------------------------------------

 ------------------------------------------------------------------------ -------
  Callable                                                                  Prio
 ------------------------------------------------------------------------ -------
 Symfony\Component\Security\...\SessionStrategyListener::onSuccessfulLogin()    0
 App\EventSubscriber\LoginAttemptListener::onLoginAttempt()                     0
 Symfony\Component\Security\...\PasswordMigratingListener::onLoginSuccess()     0
 ------------------------------------------------------------------------ -------
```

```
$ symfony console debug:firewall --events api
...

Event listeners for firewall "api"
==================================

"Symfony\Component\Security\Http\Event\LoginSuccessEvent" event
---------------------------------------------------------------


------------------------------------------------------------------------ -------
 Callable                                                                Prio
------------------------------------------------------------------------ -------
Symfony\Component\Security\...\SessionStrategyListener::onSuccessfulLogin()    0
Symfony\Component\Security\...\PasswordMigratingListener::onLoginSuccess()     0
------------------------------------------------------------------------ -------
```

Security events cycle



https://symfony.com/doc/current/security/#security-events

Security Passports

Contains all information used for authentication

- UserBadge
- PasswordCredentials

- CsrfTokenBadge
- LdapBadge
- RememberMeBadge

https://symfony.com/doc/current/security/custom_authenticator.html#security-passport

```yaml
# config/packages/security.yaml
security:
    # ...

    firewalls:
        api:
            pattern: '^/api'
            json_login:
                check_path: 'api_login'
```


https://symfony.com/doc/current/security/#json-login

```
POST /api/login
{
    "username": "wouter",
    "password": "secr$t"
}
```

```
POST /api/login
{
    "username": "wouter",
    "password": "secr$t",
    "csrf_token": "684f..."
}
```

```php
use Symfony\Component\EventDispatcher\Attribute\AsEventListener;
use Symfony\Component\Security\Http\Event\CheckPassportEvent;

class CsrfJsonLoginListener
{
    #[AsEventListener(
        event: CheckPassportEvent::class,
        priority: 2048,
        dispatcher: 'security.event_dispatcher.api'
    )]
    public function onCheckPassportEvent(CheckPassportEvent $event)
    {
        // ...
    }
}
```

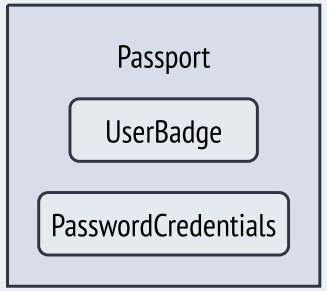
```php
use Symfony\Component\EventDispatcher\Attribute\AsEventListener;
use Symfony\Component\Security\Http\Event\CheckPassportEvent;

class CsrfJsonLoginListener
{
    #[AsEventListener(
        event: CheckPassportEvent::class,
        priority: 100,
        dispatcher: 'security.event_dispatcher.api'
    )]
    public function onCheckPassportEvent(CheckPassportEvent $event)
    {
        // ...
    }
}
```

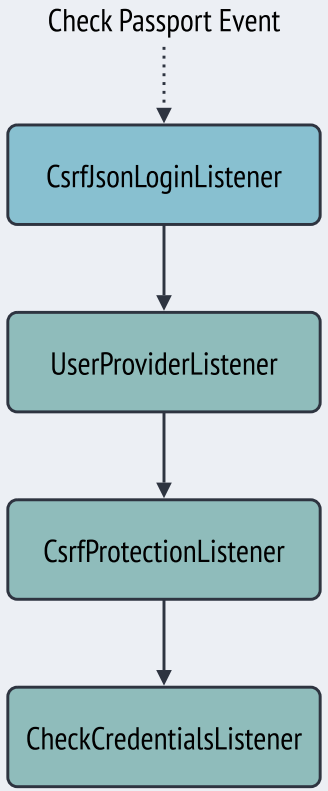
```php
// ...
public function onCheckPassportEvent(CheckPassportEvent $event)
{
    if (!$event->getAuthenticator() instanceof JsonLoginAuthenticator) {
        return;
    }



}
```

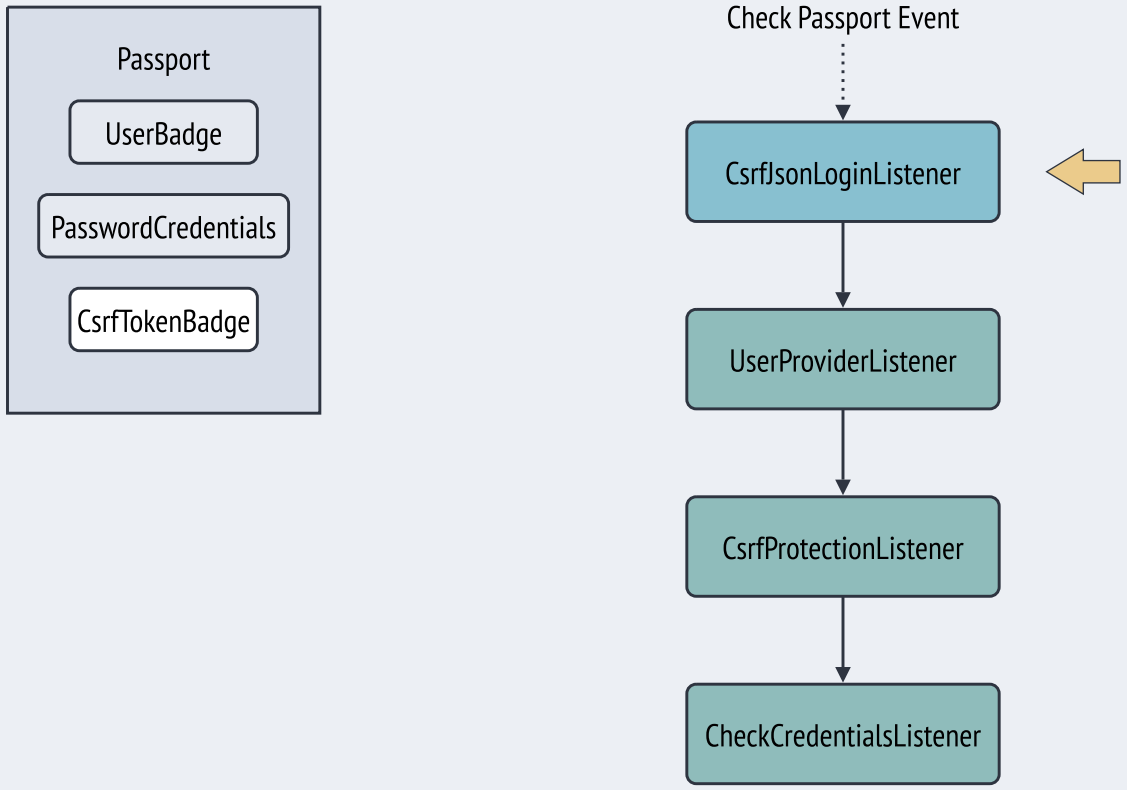
```php
// ...
public function onCheckPassportEvent(CheckPassportEvent $event)
{
    if (!$event->getAuthenticator() instanceof JsonLoginAuthenticator) {
        return;
    }

    $passport = $event->getPassport();




}
```

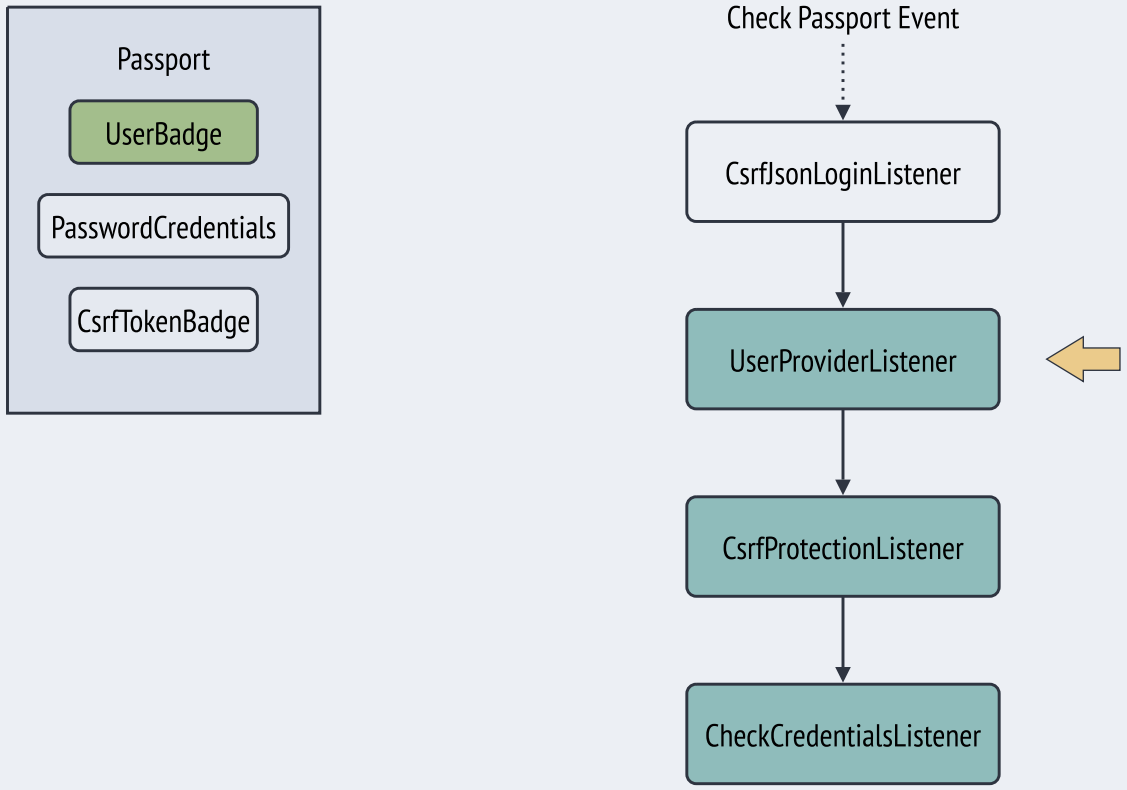
```php
// ...
public function onCheckPassportEvent(CheckPassportEvent $event)
{
    if (!$event->getAuthenticator() instanceof JsonLoginAuthenticator) {
        return;
    }

    $passport = $event->getPassport();
    $passport->addBadge(
        new CsrfTokenBadge(
            'authenticate',
            $this->requestStack->getMainRequest()->get('csrf_token'),
        )
    );
}
```

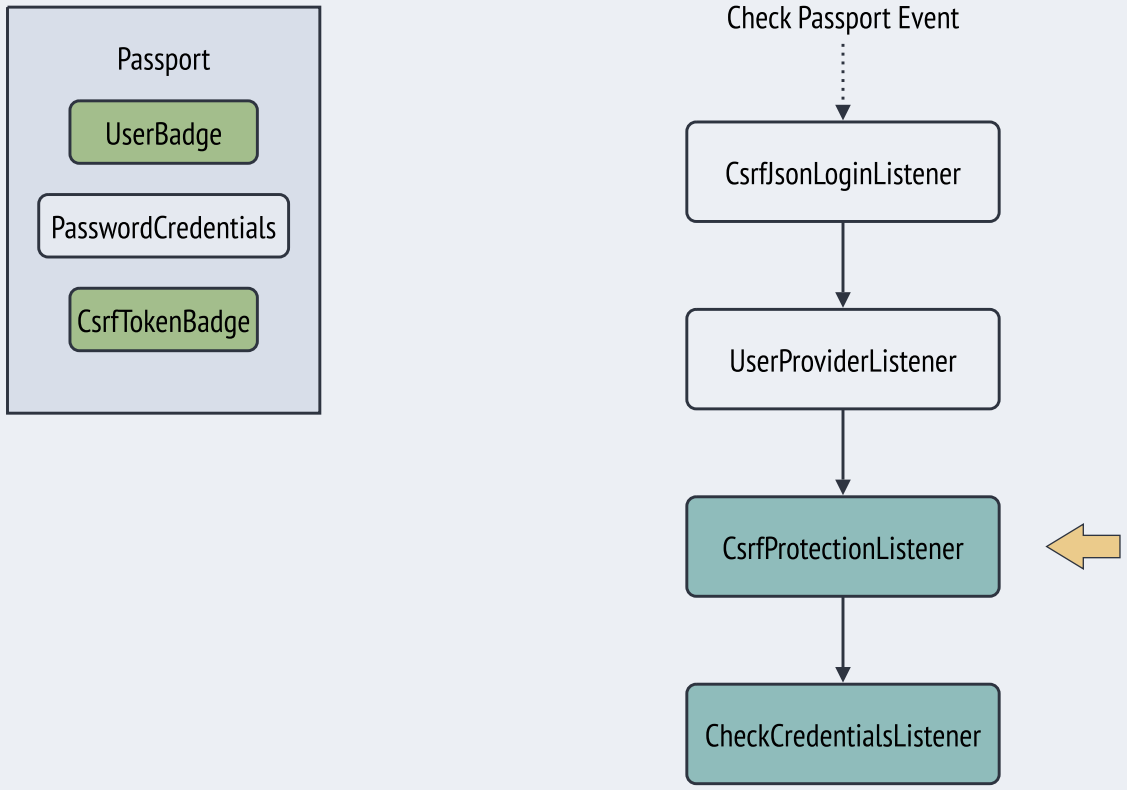
```php
// ...
public function onCheckPassportEvent(CheckPassportEvent $event)
{
    if (!$event->getAuthenticator() instanceof JsonLoginAuthenticator) {
        return;
    }

    $passport = $event->getPassport();
    $passport->addBadge(
        new CsrfTokenBadge(
            'authenticate',
            $this->requestStack->getMainRequest()->get('csrf_token'),
        )
    );
}
```

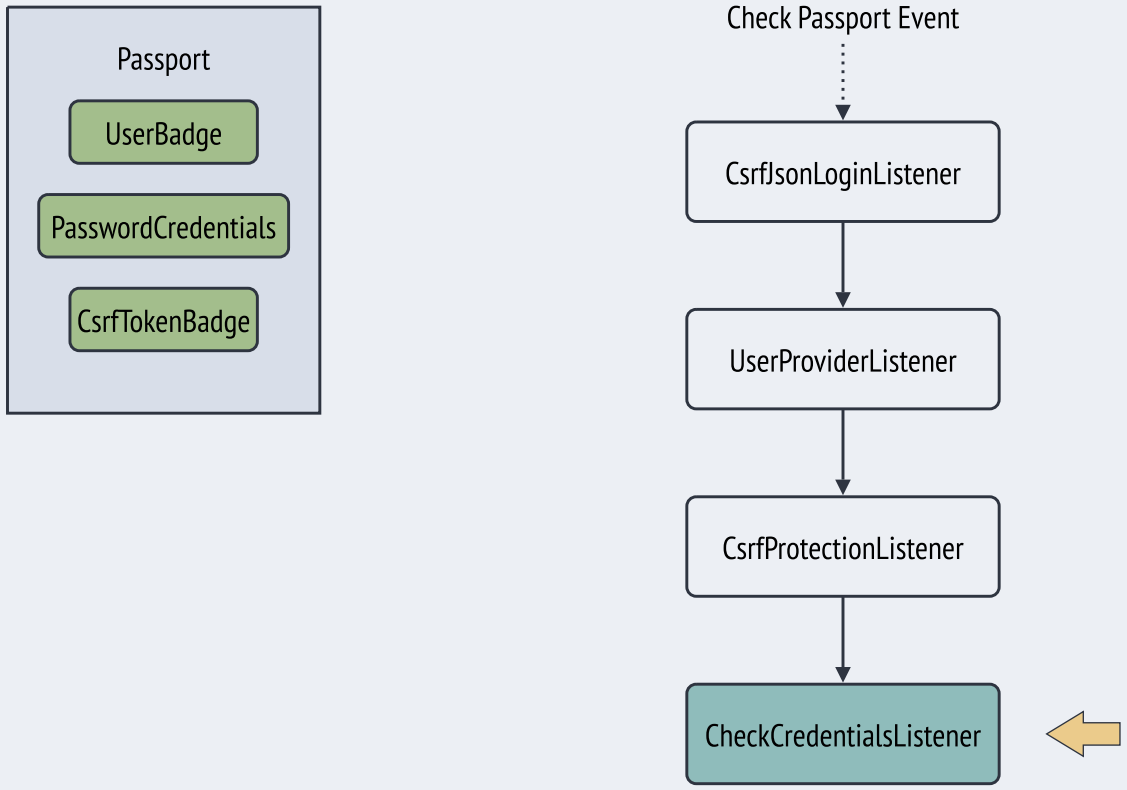

Passport
UserBadge
PasswordCredentials

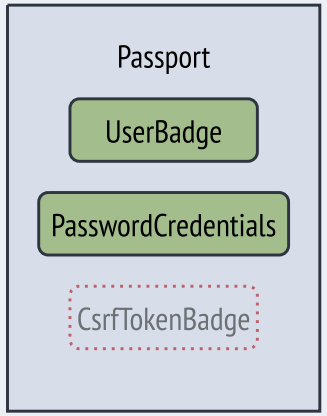

Check Passport Event
CsrfJsonLoginListener
UserProviderListener
CsrfProtectionListener
CheckCredentialsListener

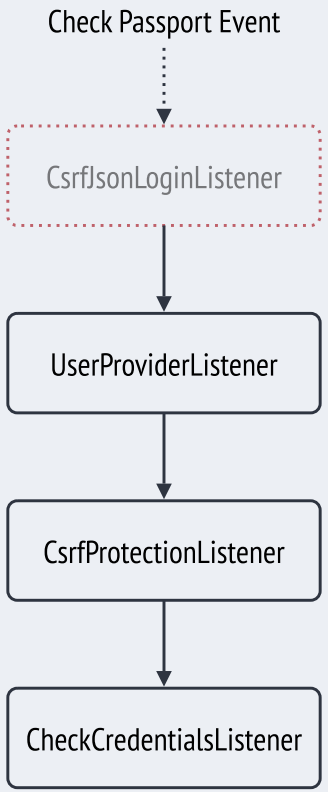

Passport
UserBadge
PasswordCredentials
CsrfTokenBadge
Check Passport Event
CsrfJsonLoginListener
UserProviderListener
CsrfProtectionListener
CheckCredentialsListener


Passport
UserBadge
PasswordCredentials
CsrfTokenBadge
Check Passport Event
CsrfJsonLoginListener
UserProviderListener
CsrfProtectionListener
CheckCredentialsListener


Passport
UserBadge
PasswordCredentials
CsrfTokenBadge
Check Passport Event
CsrfJsonLoginListener
UserProviderListener
CsrfProtectionListener
CheckCredentialsListener

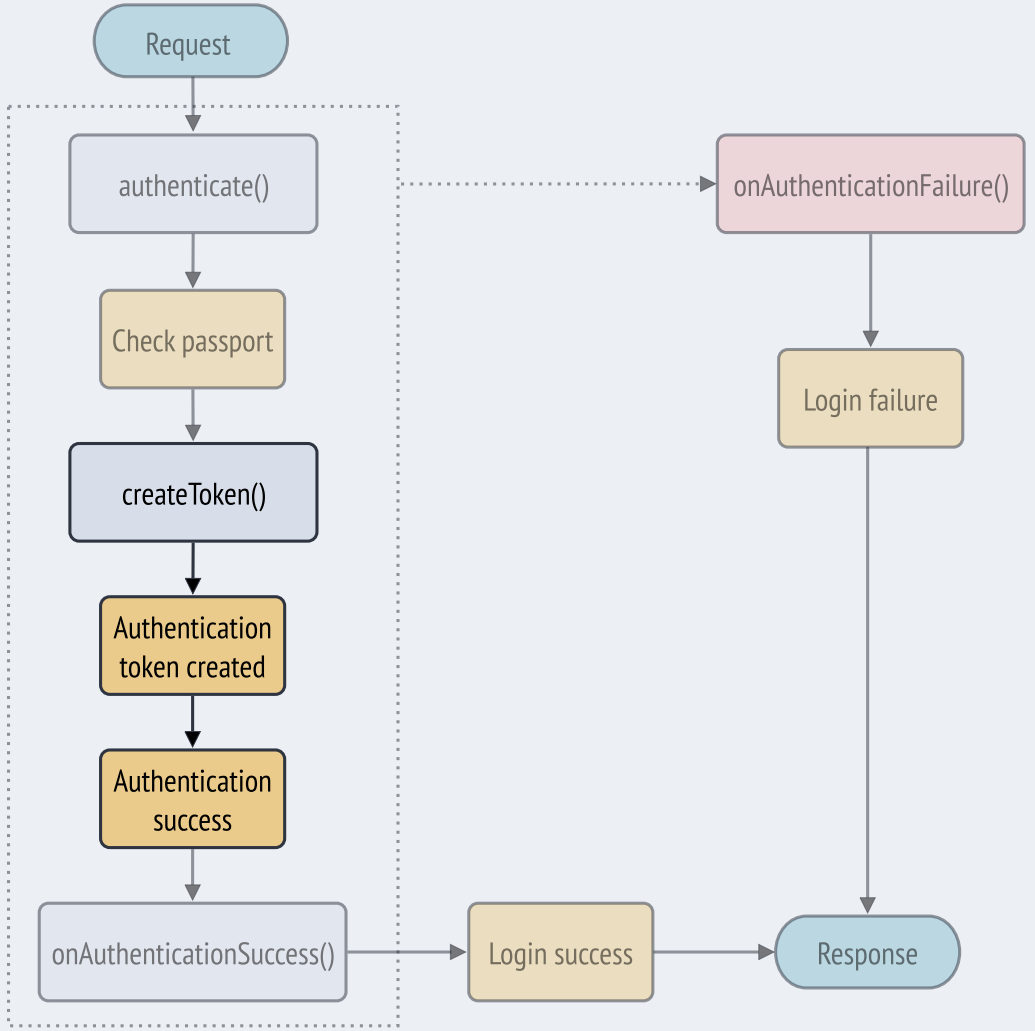

Passport
UserBadge
PasswordCredentials
CsrfTokenBadge
Check Passport Event
CsrfJsonLoginListener
UserProviderListener
CsrfProtectionListener
CheckCredentialsListener


Passport
UserBadge
PasswordCredentials
CsrfTokenBadge


Check Passport Event
CsrfJsonLoginListener
UserProviderListener
CsrfProtectionListener
CheckCredentialsListener

```yaml
# config/packages/security.yaml
security:
    # ...

    firewalls:
        api:
            pattern: '^/api'
            json_login:
                check_path: 'api_login'
            required_badges: ['CsrfTokenBadge']
```

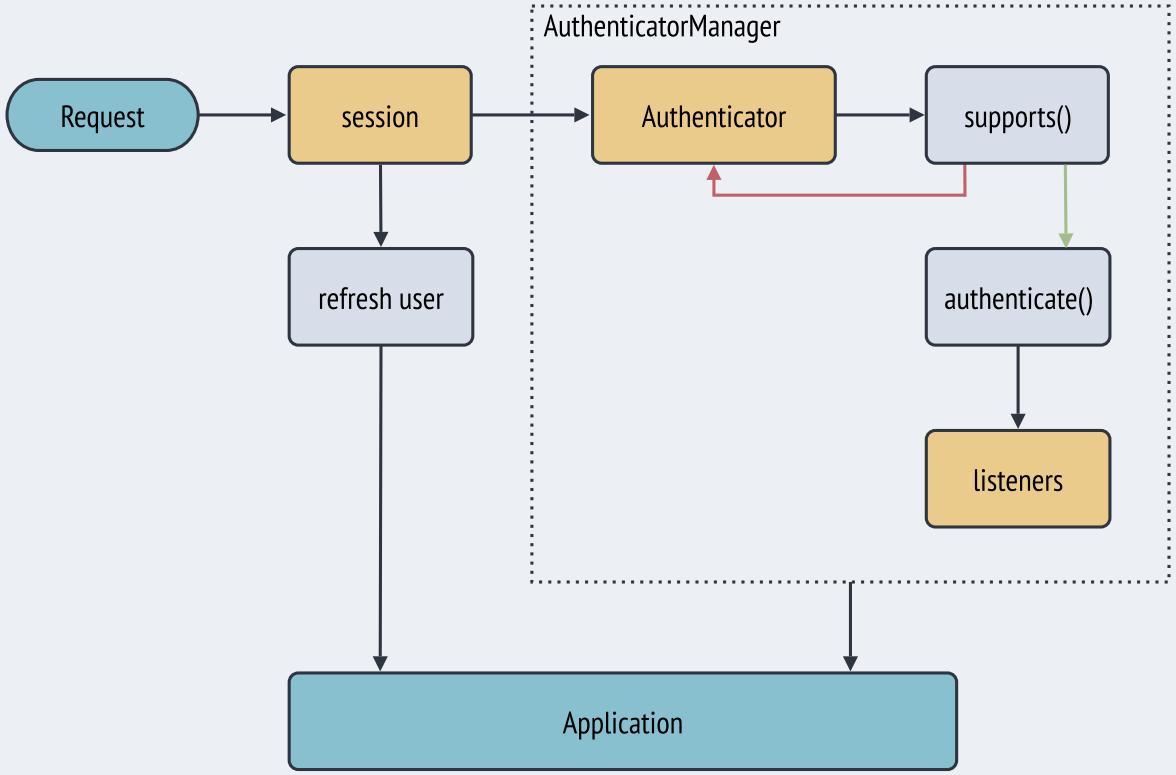
Security events cycle



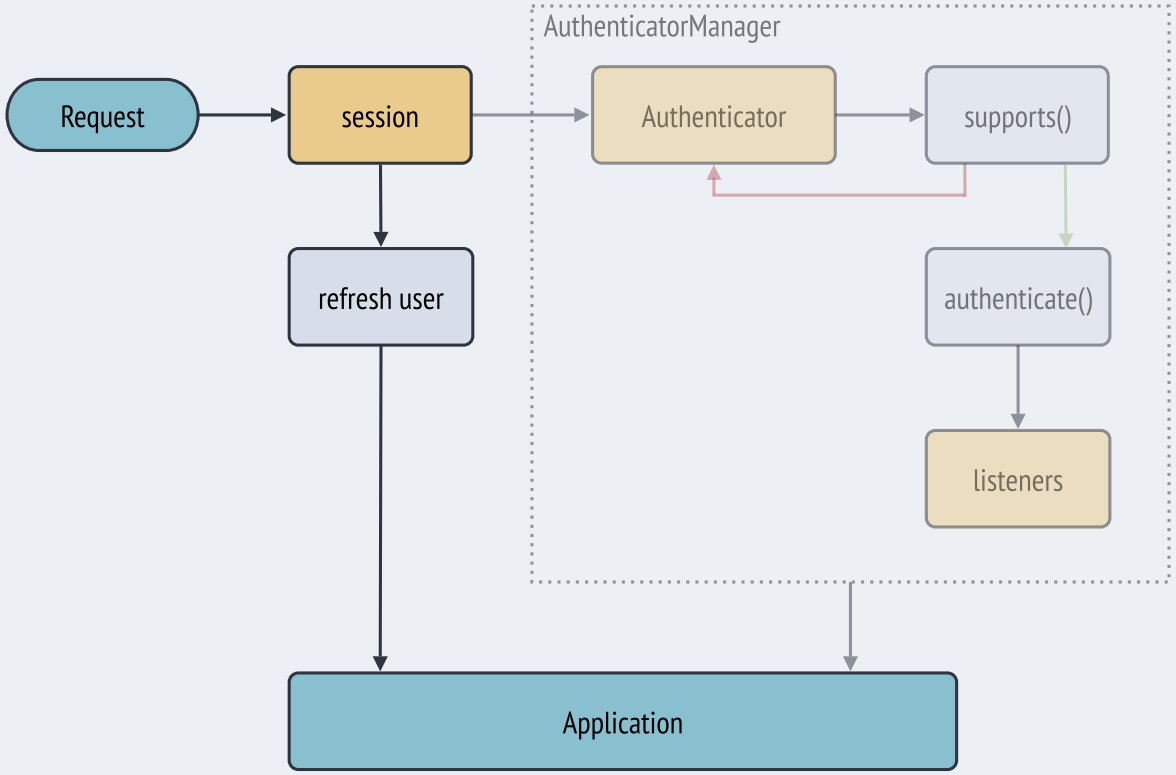
https://symfony.com/doc/current/security/#security-events

Last resort:
Custom authenticators

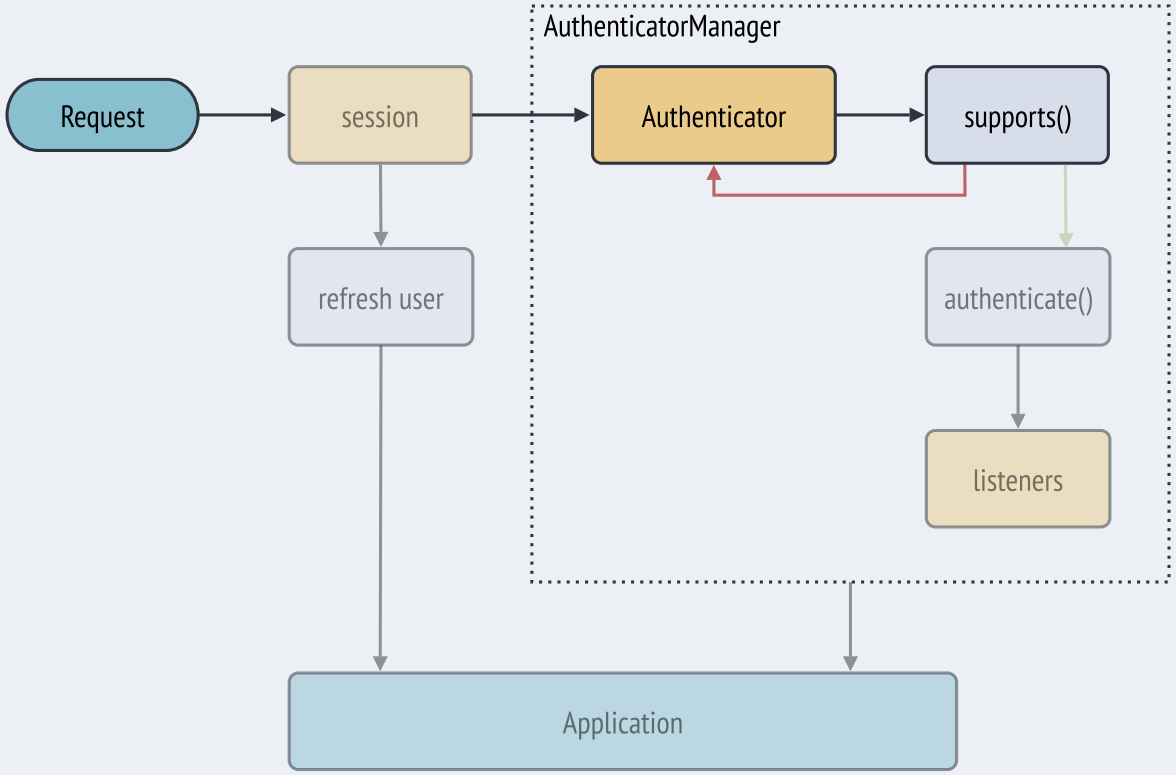
Authentication process



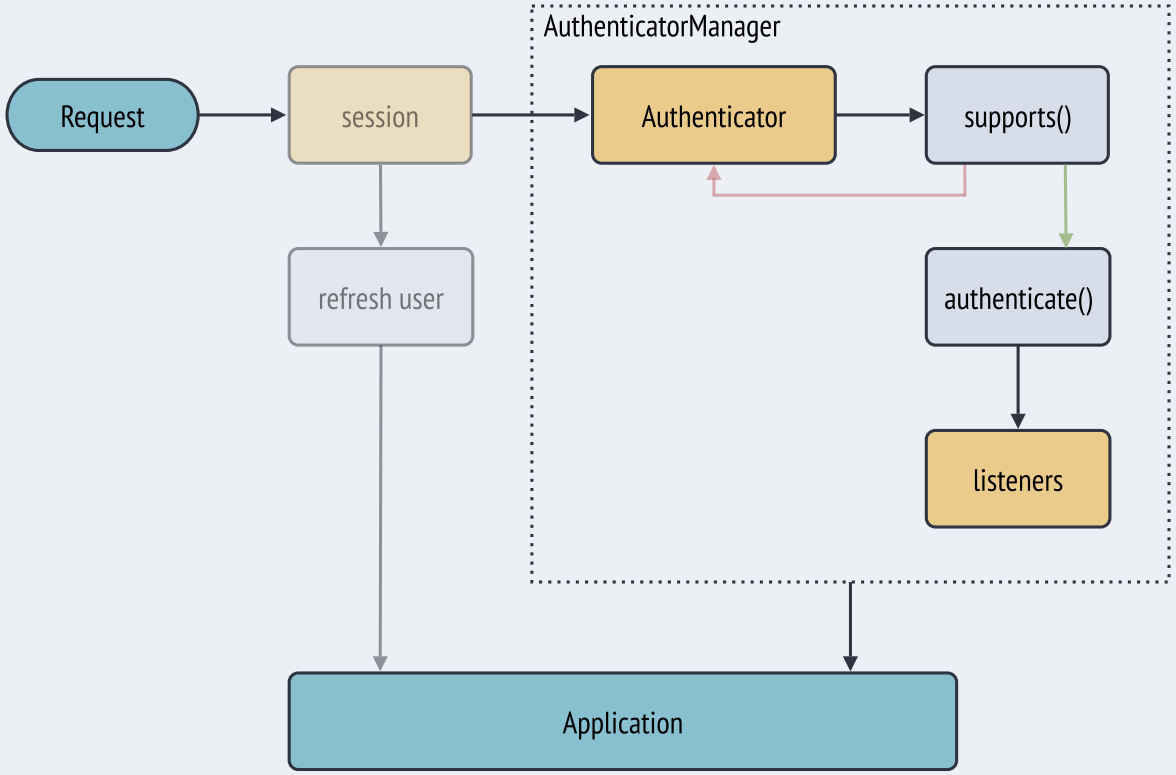
Authentication process



Authentication process



Authentication process



```xml
<?xml version="1.0" encoding="UTF-8"?>
<request>
    <auth>
      <username>wouter</username>
      <password>secr$t</password>
    </auth>

    <!-- ... -->
</request>
```

```php
use Symfony\Component\Security\Http\Authenticator\AbstractAuthenticator;

class XmlLoginAuthenticator extends AbstractAuthenticator
{
    public function supports(Request $request): ?bool
    {
        return 'xml' === $request->getFormat();
    }
}
```


https://symfony.com/doc/current/security/custom_authenticator.html

```php
class XmlLoginAuthenticator extends AbstractAuthenticator
{
    // ...
    public function authenticate(Request $request): Passport
    {
        $xml = new \SimpleXmlElement($request->getContent());









    }
}
```

```php
class XmlLoginAuthenticator extends AbstractAuthenticator
{
    // ...
    public function authenticate(Request $request): Passport
    {
        $xml = new \SimpleXmlElement($request->getContent());

        return new Passport(
            new UserBadge($xml->auth->username),
            new PasswordCredentials($xml->auth->password)
        );
    }
}
```

```php
class XmlLoginAuthenticator extends AbstractAuthenticator
{
    // ...
    public function authenticate(Request $request): Passport
    {
        $xml = new \SimpleXmlElement($request->getContent());
        if (!isset($xml->auth)) {
            throw new CustomUserMessageAuthenticationException(
                'No <auth> element found.'
            );
        }

        return new Passport(
            new UserBadge($xml->auth->username),
            new PasswordCredentials($xml->auth->password)
        );
    }
}
```

```php
class XmlLoginAuthenticator extends AbstractAuthenticator
{
    // ...

    public function onAuthenticationSuccess(
        Request $request, TokenInterface $token, string $firewallName
    ): ?Response
    {
        return null;
    }

    public function onAuthenticationFailure(
        Request $request, AuthenticationException $exception
    ): ?Response
    {
        return new XmlResponse($exception->getMessageKey(), 401);
    }
}
```

Guard **AuthenticatorInterface**

```
supports(): bool

getCredentials(): mixed
getUser(): UserInterface
checkCredentials(): bool

onAuthenticationSuccess(): ?Response
onAuthenticationFailure(): ?Response

supportsRememberMe(): bool
```

New **AuthenticatorInterface**

```
supports(): ?bool

authenticate(): Passport


onAuthenticationSuccess(): ?Response
onAuthenticationFailure(): ?Response
```

```yaml
# config/packages/security.yaml
security:
    # ...

    firewalls:
        api:
            pattern: '^/api'
            custom_authenticators:
                - 'App\Security\XmlLoginAuthenticator'
```

Contribute your authenticator back

Support in Symfony 5

```yaml
# config/packages/security.yaml
security:
    enable_authenticator_manager: true

    # ...
```


https://symfony.com/doc/5.2/security/experimental_authenticators.html

A few BC breaks

- Anonymous users no longer exist
- Authentication providers become Authenticators

But:

- All built-in authenticators are compatible
- Guard authenticators are supported (for Symfony <6)

https://symfony.com/doc/5.2/security/experimental_authenticators.html

Let's make Symfony 6 even better!

Symfony 6 ideas

- Advanced makers
- Modern authenticators
- Support for authentication factors (sudo mode)

Makers

- Don't write boilerplate code
- Fully own the code afterwards

Centralization of code

- Use built-in authenticators
- Use passport badges
- Contribute back new and cool authenticators

Security events

- "Monitor" the authentication
- Customize the authentication checks
- Customize the authentication responses

Thanks!

https://github.com/wouterj-nl/security-winterworld21

https://wouterj.nl/2021/12/sfwinterworld21-security