

Docs » Symfony Bundles » Platform Bundle » Installation

Installation

Install the bundle, create entities, configure services, and run migrations.

Anonymous · 09/09/2026

Installation

1. Require the bundle

```
composer require derafu/symfony-platform-bundle
```

If the bundle is consumed as a local path repository during development:

```
{
  "repositories": [
    { "type": "path", "url": "../packages/DerafuPlatformBundle" }
  ],
  "require": {
    "derafu/symfony-platform-bundle": "*"
  }
}
```

2. Register the bundle

If Symfony Flex did not register it automatically:

```
// config/bundles.php
return [
    // ...
    Derafu\PlatformBundle\PlatformBundle::class => ['all' => true],
];
```

3. Create the concrete entities

The bundle provides abstract `Base* MappedSuperclass` classes. The app must create concrete entities that extend them. By convention, entities are grouped by domain under `App\Entity\`.

Identity entities (required)

```
// src/Entity/Auth/User.php
namespace App\Entity\Auth;

use Derafu\PlatformBundle\Identity\Entity\BaseUser;
use Doctrine\ORM\Mapping as ORM;

#[ORM\Entity]
#[ORM\Table(name: 'auth_users')]
class User extends BaseUser {}
```

Repeat for every identity entity:

Base class	Suggested concrete	Suggested table
BaseUser	App\Entity\Auth\User	auth_users
BaseRole	App\Entity\Auth\Role	auth_roles
BasePermission	App\Entity\Auth\Permission	auth_permissions
BaseUserRole	App\Entity\Auth\UserRole	auth_user_roles
BaseRolePermission	App\Entity\Auth\RolePermission	auth_role_permissions
BaseOrganization	App\Entity\Auth\Organization	auth_organizations
BaseOrganizationMembership	App\Entity\Auth\OrganizationMembership	auth_organization_memberships
BaseOrganizationInvitation	App\Entity\Auth\OrganizationInvitation	auth_organization_invitations
BaseVerificationToken	App\Entity\Auth\VerificationToken	auth_verification_tokens
BaseApiKey	App\Entity\Auth\ApiKey	auth_api_keys
BaseJwtToken	App\Entity\Auth\JwtToken	auth_jwt_tokens
BaseLoginRecord	App\Entity\Auth>LoginRecord	auth_login_records
BaseImpersonationRecord	App\Entity\Auth\ImpersonationRecord	auth_impersonation_records
BaseTeam	App\Entity\Auth\Team	auth_teams
BaseTeamMembership	App\Entity\Auth\TeamMembership	auth_team_memberships
BaseLocale	App\Entity\Auth\Locale	auth_locales
BaseTimezone	App\Entity\Auth\Timezone	auth_timezones

All base classes live under `Derafu\PlatformBundle\Identity\Entity\`.

Apps entities (required if using the Apps module)

Base class	Suggested concrete	Suggested table
BaseUserAppInstallation	App\Entity\Apps\UserAppInstallation	apps_user_installations
BaseOrganizationAppInstallation	App\Entity\Apps\OrganizationAppInstallation	apps_organization_installations

Base classes: `Derafu\PlatformBundle\Apps\Entity\`.

Notifications entities (required if using the Notifications module)

Base class	Suggested concrete	Suggested table
BaseInAppNotification	App\Entity\Notif\InAppNotification	notif_inapp
BaseUserWebhookEndpoint	App\Entity\Notif\UserWebhookEndpoint	notif_user_webhook_endpoints
BaseOrganizationWebhookEndpoint	App\Entity\Notif\OrganizationWebhookEndpoint	notif_org_webhook_endpoints
BaseUserWebhookDelivery	App\Entity\Notif\UserWebhookDelivery	notif_user_webhook_deliveries
BaseOrganizationWebhookDelivery	App\Entity\Notif\OrganizationWebhookDelivery	notif_org_webhook_deliveries
BaseUserNotificationPreference	App\Entity\Notif\UserNotificationPreference	notif_user_preferences
BaseUserDigestPreference	App\Entity\Notif\UserDigestPreference	notif_user_digest_preferences

Base classes: Derafu\PlatformBundle\Notifications\Entity\.

4. Create the concrete repositories

For each entity, create a repository that calls the bundle's base with your concrete class:

```
// src/Repository/Auth/UserRepository.php
namespace App\Repository\Auth;

use App\Entity\Auth\User;
use Derafu\PlatformBundle\Identity\Repository\BaseUserRepository;
use Doctrine\Persistence\ManagerRegistry;

class UserRepository extends BaseUserRepository
{
    public function __construct(ManagerRegistry $registry)
    {
        parent::__construct($registry, User::class);
    }
}
```

Base repositories exist for all identity, apps, and notifications entities.

5. Configure the bundle

If you follow the `App\Entity\Auth\*`, `App\Entity\Apps\*`, and `App\Entity\Notif\*` naming conventions, **most configuration is optional** — the defaults match. See [Configuration](#) for the full reference.

Minimum required configuration when the Notifications module is active:

```
# config/packages/derafu_platform.yaml
derafu_platform:
    notifications:
```

```
from_email: '%env(MAILER_FROM_EMAIL)%'  
from_name: '%env(MAILER_FROM_NAME)%'
```

6. Configure Symfony Security

Register the bundle's authenticators in your security firewall:

```
# config/packages/security.yaml  
security:  
    firewalls:  
        main:  
            form_login:  
                login_path: derafu_platform_identity_login  
                check_path: derafu_platform_identity_login  
            custom_authenticators:  
                - Derafu\PlatformBundle\Identity\Security\ApiKeyAuthenticator  
                - Derafu\PlatformBundle\Identity\Security\MagicLinkAuthenticator
```

7. Configure Messenger (for async notifications)

The Notifications module dispatches messages via Symfony Messenger. Route them to a transport for async delivery:

```
# config/packages/messenger.yaml  
framework:  
    messenger:  
        transports:  
            async: '%env(MESSENGER_TRANSPORT_DSN)%'  
        routing:  
            'Derafu\PlatformBundle\Notifications\Messenger\*': async
```

8. Run migrations

```
bin/console doctrine:migrations:diff  
bin/console doctrine:migrations:migrate
```

9. Seed initial data

The bundle does not ship seed data. Create the roles your app needs and seed locales/timezones:

```
INSERT INTO auth_roles (code, name) VALUES  
('system.admin', 'System Administrator'),
```

```
('org.owner',      'Organization Owner'),  
( 'org.admin',     'Organization Admin'),  
( 'org.member',    'Organization Member');
```

For locales and timezones the bundle provides a `SeedCommand` that you can trigger with the `derafu:platform:seed` console command, provided your app registers seeders implementing `SeederInterface`.

Last updated on 09/09/2026