Mise en place

OdooAsso — Projet de modules Odoo pour la gestion associative

Contexte

Odoo présente une orientation naturelle vers l'entreprise et un niveau de complexité d'usage trop important pour les utilisateurs d'associations. L'objectif de ce projet est de regrouper tous les besoins en quelques modules cohérents pour simplifier au maximum la gestion, et de produire un travail de qualité suffisante pour être partagé avec d'autres associations.

Stack cible : Odoo 16 Community, déployé en Docker. **Modules tiers utilisés :** modules OCA (vertical association, membership, partner...), modules Deodoo.

Architecture du dépôt

Structure monorepo inspirée du standard OCA :

```
OdooAsso/                          # dépôt public GitHub/GitLab
├── README.md
├── projet.md
├── .pre-commit-config.yaml        # linting OCA
├── setup.cfg
├── oca_dependencies.txt           # versions des modules OCA requis
├── asso_base/                     # socle — groupes, menus racine, mixins
```

```
├── asso_membership/              # gestion des membres et cotisations
├── asso_event/                   # événements et activités
├── asso_accounting/              # comptabilité simplifiée (loi 1901)
├── asso_portal/                  # portail adhérent
└── asso_website/                 # front public
```

Principe de séparation

`asso_base` déclare les groupes d'accès, les menus racine et les mixins réutilisables. Tous les autres modules en dépendent. Les modules métier sont indépendants les uns des autres autant que possible, pour permettre une adoption partielle.

Namespace

Le préfixe `asso_` est choisi pour être neutre et réutilisable (pas de nom d'association spécifique). Si le projet vise explicitement le contexte français (loi 1901, plan comptable associatif), le préfixe `fr_asso_` peut être envisagé.

Stratégie de simplification UX

C'est le cœur de la valeur ajoutée du projet. Odoo est simplifié à trois niveaux :

1. Groupes d'accès dédiés

Des groupes propres sont créés, sans réutiliser les groupes Odoo natifs :

- `group_asso_user` — membre du bureau, usage courant
- `group_asso_manager` — responsable, accès étendu
- `group_asso_admin` — administrateur technique

Cela permet de masquer tout ce qui n'est pas pertinent pour une association.

2. Menus simplifiés

- Les menus natifs Odoo sont désactivés (`active="False"`) pour les groupes `asso_*`

- Une arborescence plate et orientée tâches les remplace :
  - Membres → Activités → Finances → Communication

3. Vues allégées

- Les vues natives sont héritées pour masquer les champs inutiles
- L'attribut `groups` est utilisé sur les champs avancés (visibles admins seulement)

4. Wizards pour les actions courantes

Les opérations fréquentes sont regroupées en wizards pour éviter la navigation multi-menus : inscription en masse, relances cotisations, exports, etc.

Qualité et partageabilité

Outils

Outil	Rôle
`pre-commit` (config OCA)	Linting XML, Python, manifest
Tests unitaires Odoo	Validation des modèles et flux métier
`oca_dependencies.txt`	Déclaration explicite des dépendances OCA

Lancer les tests

```
odoo-bin --test-enable --stop-after-init -i asso_membership
```

Conventions

- Chaque module déclare ses dépendances OCA explicitement dans son `__manifest__.py`
- Les données de configuration (`data/`) sont séparées des données de démonstration (`demo/`)
- Les champs custom sont préfixés avec le nom du module (`membership_date_end` plutôt que `x_date_end`)

Environnement de développement

Le projet tourne en Docker. Ajouter un `docker-compose.override.yml` pour monter le dépôt directement dans le `addons_path` :

```
services:
  odoo:
    volumes:
      - ./OdooAsso:/mnt/extra-addons
```

Travailler en mode `--dev xml` pour recharger les vues sans redémarrer le serveur.

Modules — périmètre fonctionnel

`asso_base`

- Groupes d'accès et règles de sécurité
- Menus racine et structure de navigation
- Modèles abstraits et mixins partagés
- Configuration générale de l'association

`asso_membership`

- Gestion des adhérents (hérite `res.partner`)
- Types d'adhésion et tarifs
- Suivi des cotisations et renouvellements
- Relances automatiques

`asso_event`

- Événements et activités de l'association
- Inscriptions et présences
- Lien avec les adhérents

`asso_accounting`

- Simplification de la comptabilité pour le contexte associatif (loi 1901)
- Plan comptable associatif
- Suivi des subventions

`asso_portal`

- Espace adhérent en ligne
- Consultation et renouvellement d'adhésion
- Inscription aux événements

`asso_website`

- Pages publiques de l'association
- Formulaire d'adhésion en ligne
- Agenda public des événements

Étapes de démarrage

1. Initialiser le dépôt Git avec la structure monorepo
2. Configurer `pre-commit` avec la config OCA standard
3. Lister les fonctionnalités par module en partant des douleurs utilisateurs actuelles
4. Créer `asso_base` : groupes, menus racine, modèles abstraits
5. Migrer progressivement les développements existants dans cette structure

---

Revision #1
Created 2026-03-23 18:21:07 UTC by Nicolas
Updated 2026-03-23 18:28:30 UTC by Nicolas