

Hotel Room Service

Bienvenue sur notre pl

Nom ou ville

Nom ou ville

Hôtel Paris Centre

Adresse : 10 Rue des Fleurs

Ville : Paris



[Voir les détails](#)

1

Projet PHP/Symfony

Développement d'une plateforme de réservation hôtelière avancée



Présentation du Projet



Développement Web

Site CRUD MVC avec Symfony



Base de Données

MongoDB NoSQL pour flexibilité



Application

Système de réservation centralisé



Architecture et Données

Structure solide pour une gestion efficace

1

2

Configuration ODM

Mapping des documents MongoDB

config/packages/doctrine_mongodb.yaml

Modèle de Données

Traduction du MCD (Client, Hotel, Chambre, Réservation, Comment)

src/Document/*.php

Règles de Gestion Clés

Gestion des Réservations

Hôtel, chambres multiples, période valide

`HotelController::reserveRoom`

Catégories d'Hôtels

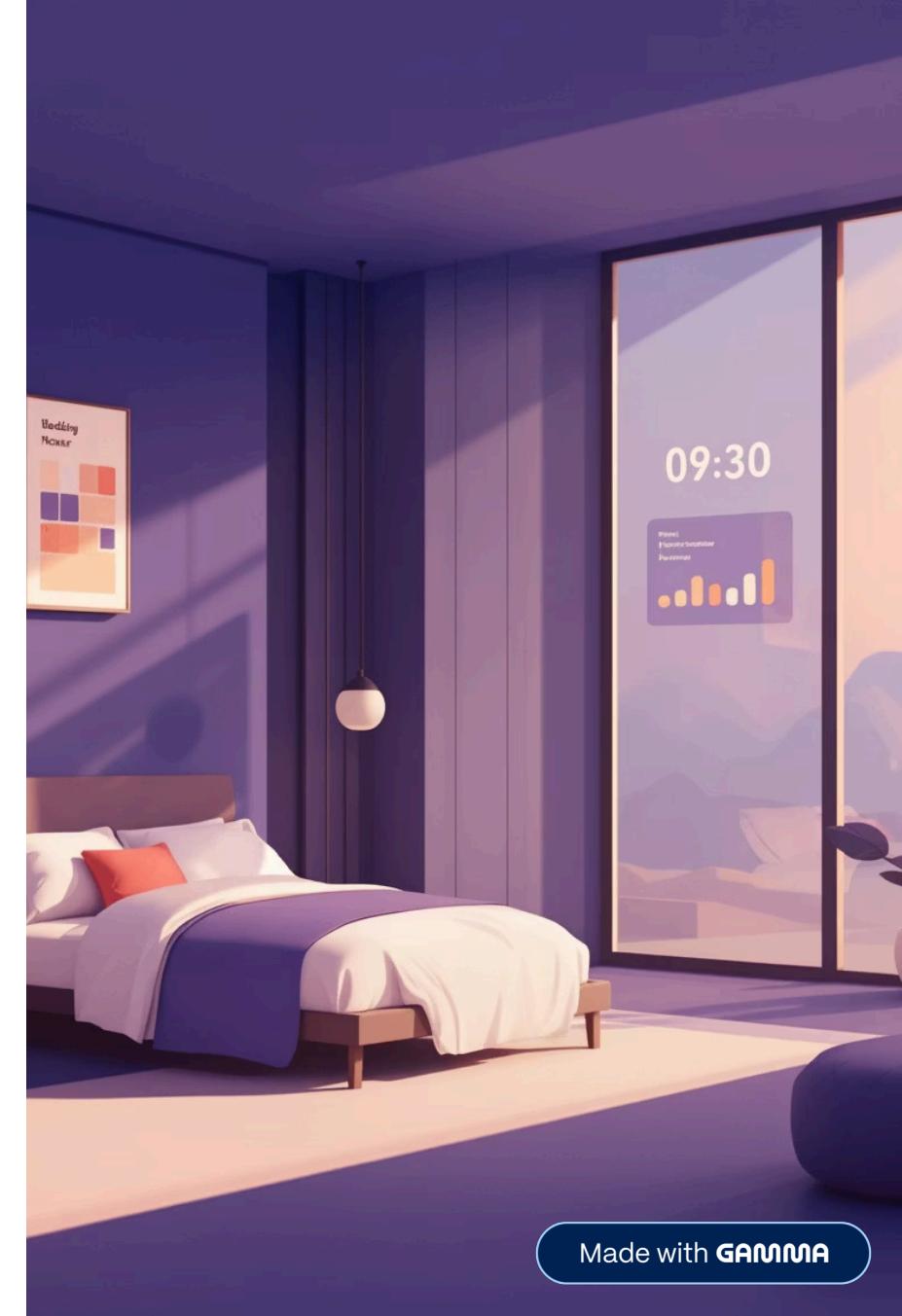
De 1 à 5 étoiles, validations rigoureuses

`src/Document/Hotel.php`

Chambres

Typées et tarifées, gestion complète

`ChambreController`



Fonctionnalités Publiques



Recherche Intégrée

Par nom/ville, période de séjour

HotelController::home/search



Réservation Facile

Sans paiement, login requis

HotelController::reserveRoom



Authentification

Inscription, connexion, mot de passe oublié

UserController::register/login

```
#[Route('/search', name: 'hotel_search', methods: ['GET'])]
public function search(Request $request, DocumentManager $dm): Response
{
    $query = trim(string($request->query->get('q')));
    $dateDebut = $request->query->get('date_debut');
    $dateFin = $request->query->get('date_fin');

    $dateDebutObj = DateTime::createFromFormat('Y-m-d', $dateDebut);
    $dateFinObj = DateTime::createFromFormat('Y-m-d', $dateFin);

    if ($dateDebutObj <= null || $dateFinObj <= null) {
        $dateDebutObj = new \DateTime('2018-01-01');
        $dateFinObj = new \DateTime('2018-12-31');
    }

    if ($dateDebutObj > $dateFinObj) {
        $this->addFlash('error', 'La date de fin doit être postérieure à la date de début.');
        return $this->redirectToRoute('hotel_index');
    }

    $qb = $dm->getRepository(Hotel::class)->createQueryBuilder();
    $qb->where('h.date_debut >=:dateDebut AND h.date_fin <=:dateFin');
    $qb->setParameter('dateDebut', $dateDebutObj);
    $qb->setParameter('dateFin', $dateFinObj);
}
```

```
#[Route('/hotel/{id}/reserve', name: 'hotel_reserve', methods: ['POST'])]
public function reserveRoom(string $id, Request $request, DocumentManager $dm): Response
{
    $hotel = $dm->getRepository(Hotel::class)->find($id);

    if (!$hotel) {
        throw $this->createNotFoundException('Hôtel non trouvé.');
    }

    if (!$this->isGranted('IS_AUTHENTICATED_FULLY')) {
        $this->addFlash('error', 'Vous devez être connecté pour réserver une chambre.');
        return $this->redirectToRoute('app_login');
    }

    $chambreIds = (array) $request->request->all('chambre_ids');
    if (empty($chambreIds)) {
        $this->addFlash('error', 'Veuillez sélectionner au moins une chambre.');
        return $this->redirectToRoute('hotel_show', ['id' => $hotel->getId()]);
    }

    $dateDebut = new \DateTime($request->request->get('date_debut'));
    $dateFin = new \DateTime($request->request->get('date_fin'));
```

```
/**
 * Gère l'inscription d'un nouvel utilisateur (client).
 */
#[Route('/register', name: 'app_register', methods: ['GET', 'POST'])]
public function register(Request $request, UserPasswordEncoderInterface $passwordHasher, DocumentManager $dm): Response
{
    if ($request->isMethod('POST')) {
        $nom = $request->request->get('nom');
        $email = $request->request->get('email');
        $telephone = $request->request->get('telephone');
        $password = $request->request->get('password');
        $confirmPassword = $request->request->get('confirm_password');

        // Vérifiez que les mots de passe correspondent
        if ($password != $confirmPassword) {
            $this->addFlash('error', 'Les mots de passe ne correspondent pas.');
            return $this->redirectToRoute('app_register');
        }

        // Vérifiez l'unicité de l'email
        $existingEmail = $dm->getRepository(Client::class)->findOneBy(['email' => $email]);
        if ($existingEmail) {
            $this->addFlash('error', 'Cet email est déjà utilisé.');
            return $this->redirectToRoute('app_register');
        }
    }
}
```

Espace Client Personnalisé

Gérez vos séjours en toute simplicité.

- Visualisation des réservations
- Annulation sécurisée (CSRF)
- Ajout de commentaires

ReservationController::userReservations

```
/**  
 * Affiche les réservations de l'utilisateur connecté.  
 */  
#[Route('/mes-reservations', name: 'user_reservations', methods: ['GET'])]  
public function userReservations(DocumentManager $dm): Response  
{  
    $user = $this->getUser();  
  
    if (!$user) {  
        $this->addFlash('error', 'Vous devez être connecté pour accéder à vos réservations.');//  
        return $this->redirectToRoute('app_login');//  
    }  
  
    $reservations = $dm->getRepository(Reservation::class)->findBy(['client' => $user]);  
  
    return $this->render('reservation/index.html.twig', [  
        'reservations' => $reservations,  
    ]);  
}
```

Bienvenue sur notre plateforme

Ajouter un hôtel

Nom ou ville

Nom ou ville

Hôtel Paris Centre

Adresse : 10 Rue des Fleurs

Ville : Paris



Voir les détails

Modifier

Supprimer

1

Tableau de Bord Administrateur

1

CRUD Hôtels

Gestion complète des établissements

2

CRUD Chambres

Avec pagination et recherche avancée

3

CRUD Clients

Pagination, recherche et détails

4

CRUD Réservations

Filtrage, création, édition et suppression

Sécurité et Authentification

Accès contrôlé et protection des données

- **Rôles Utilisateur:** ROLE_USER / ROLE_ADMIN
- **Firewall Symfony:** Restrictions de routes
- **Navbar Conditionnelle:** Visibilité selon le rôle

Gestion des utilisateurs Mes réservations Se déconnecter

config/packages/security.yaml

Expérience Utilisateur

The screenshot shows a mobile application interface for "Hotel Room Service". At the top, there's a header with the app's name and a menu icon (three horizontal lines). Below the header, a large text area says "Bienvenue sur notre plateforme de réservation d'hôtels". Underneath this, there's a green button labeled "Ajouter un hôtel". Further down, there are two input fields: one labeled "Nom ou ville" with the placeholder "Nom ou ville" and another labeled "Date de début".

Responsive Design

Adaptation sur tous les appareils avec Bootstrap 5

Aucun utilisateur trouvé

Mot de pa!

Gestion des Exceptions

Messages d'erreur clairs et retours utilisateur

Qualité et Tests

Assurer la robustesse et la fiabilité du projet.

- **Tests Fonctionnels:** Login, liste clients admin
- **Dette Technique:** SonarQube à implémenter

tests/controller/SecurityControllerTest.php

```
class ClientControllerTest extends WebTestCase
{
    private function createAdminUser()
    {
        $user = new \App\Document\Client();
        $user->setNom('Marc Zuckerberg');
        $user->setEmail('marc@example.com');
        $user->setRoles(['ROLE_ADMIN']);
        $user->setPassword('password');

        try {
            $dm = static::getContainer()->get('doctrine_mongodb')->getManager();
            $dm->persist($user);
            $dm->flush();
        } catch (\Throwable $e) {
            $this->markTestSkipped('MongoDB non disponible pour le test : ' . $e->getMessage());
        }
    }

    return $user;
}
```