

**Sujet d'épreuves de la 48^e Compétition Nationale
des Métiers**

MÉTIER N°09 SOLUTIONS LOGICIELLES EN ENTREPRISE

Module 2

Soumis par :

Paul NGO, Expert WorldSkills France

Geoffrey DA ENCARNACAO, Expert adjoint WorldSkills France

EXPLICATION DU MODULE 2

| | |
|--------------------|-------------------------------------|
| DUREE DE L'ÉPREUVE | 3 heures 30 |
| DIFFUSION DU SUJET | Découvert le jour de la compétition |

MODULE 2 : Développement d'une API REST

Introduction

Le **Parc National des Calanques de Marseille** est un site naturel d'exception. Chaque année, des milliers de visiteurs participent à des activités comme la randonnée, la plongée et les excursions en kayak, mettant en tension l'équilibre fragile de cet écosystème protégé. Face à ces enjeux, l'Établissement Public du Parc National des Calanques (EPNPC) souhaite déployer plusieurs solutions logicielles afin de répondre à l'augmentation du nombre de visiteurs, tout en assurant une gestion durable et efficace des ressources.

Dans ce module, en tant que **développeur back-end**, votre mission sera de **concevoir et d'implémenter l'API REST** qui assurera la communication entre la base de données et les applications front-end. Cette API devra répondre spécifiquement aux **besoins des applications mobiles et desktop**

Contenu

Cette session comprend les documents suivants :

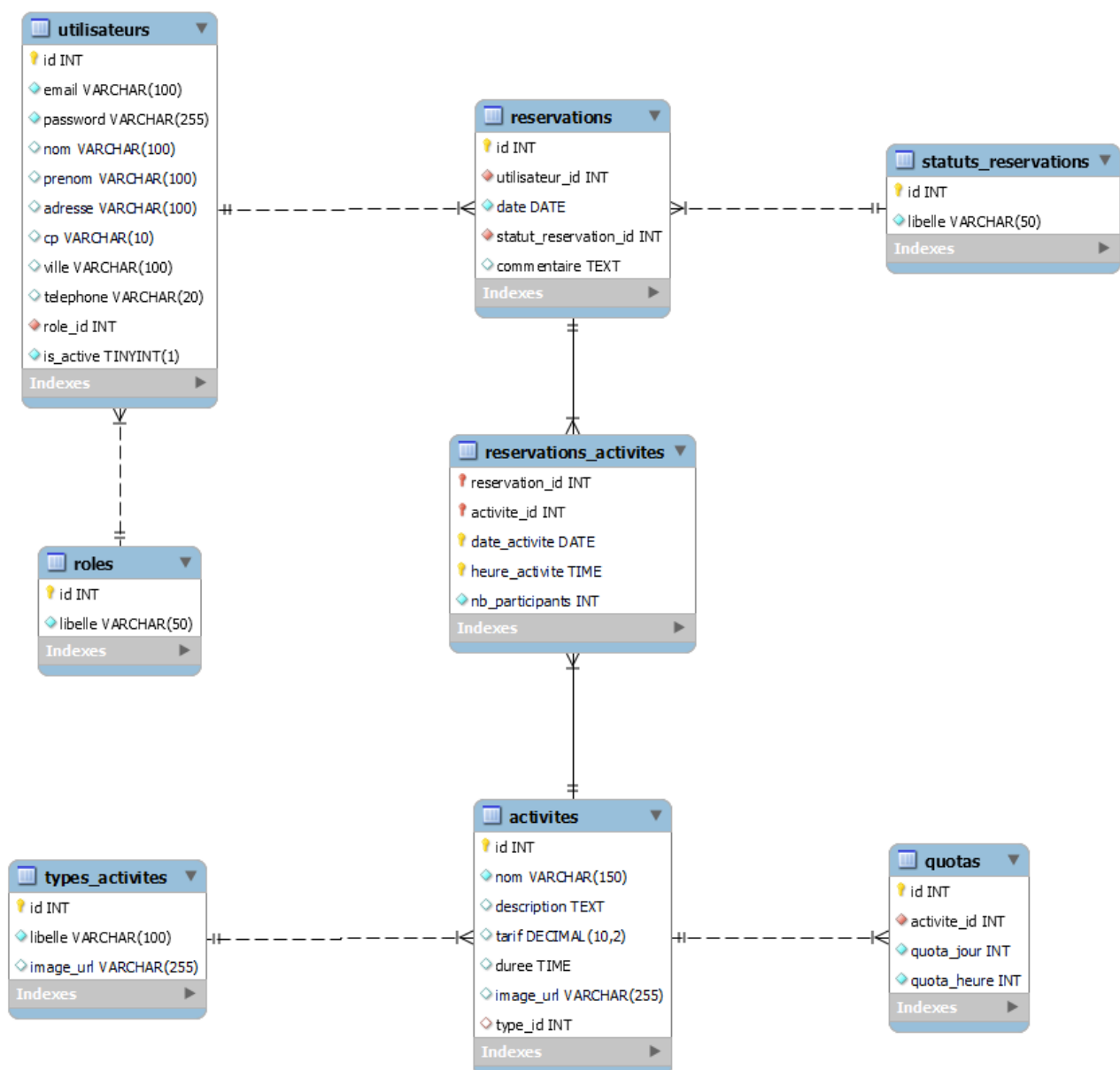
1. Instructions pour le module (PDF) : instructions détaillées décrivant les tâches à accomplir et les livrables attendus pour cette session.
2. Dossier commun : ce dossier contient des ressources supplémentaires telles que le logo, des icônes, un guide de style et d'autres éléments de conception qui peuvent être utilisés tout au long du développement de l'application.
3. Schéma de base de données (SQL) : script SQL contenant la structure des tables et des données nécessaires.
4. Image du diagramme ERD du SI

Description du projet et des tâches

2.1. LA BASE DE DONNÉES

Conformément aux consignes du commanditaire (EPNPC), la structure de la base de données fournie ne peut en aucun cas être modifiée. Cela inclut la suppression de tables, l'ajout ou la suppression de champs, ainsi que toute modification des types de données existants.

Afin de mieux comprendre la structure de la base, les concepteurs ont fourni un diagramme entité-relation (ERD, pour *Entity-Relationship Diagram* en anglais). Ce diagramme illustre le modèle conceptuel de la base de données et représente visuellement les entités ainsi que leurs relations.



En fonction de votre SGBD préféré, un script SQL est mis à disposition. Les scripts comprennent la structure de la base de données et les données nécessaires pour effectuer les tâches requises. Les données doivent être importées dans la base de données créée pour ce module, nommée "Module2_X" où X est le numéro du compétiteur.

2.2.USER STORIES

L'Établissement Public du Parc National des Calanques applique le cadre Scrum pour piloter le développement de ses applications logicielles. Scrum est un cadre de travail agile basé sur des itérations courtes (sprints), des revues fréquentes et une collaboration étroite entre les parties prenantes.

Le Product Owner, garant de la vision produit et de la valeur métier, définit, priorise et affine les besoins sous forme de User Stories.. Il s'assure que l'équipe de développement comprend parfaitement les attentes des utilisateurs finaux.

Le Product Owner vous fournit les User Stories suivantes :

Utilisateurs & Authentification

- En tant que client, je souhaite m'inscrire, afin de créer un compte et pouvoir réserver des activités dans le Parc National des Calanques.
- En tant que client, je souhaite me connecter, afin d'accéder à mon espace personnel.
- En tant que client, je souhaite consulter et mettre à jour mes informations personnelles (nom, prénom, adresse, téléphone, etc.), afin de garantir l'exactitude de mes données.
- En tant que client, je souhaite modifier mon mot de passe, afin de sécuriser mon compte.
- En tant que client, je souhaite désactiver mon compte, afin de ne plus apparaître comme utilisateur actif.
- En tant qu'administrateur, je souhaite créer un utilisateur, afin d'élargir ou de structurer l'équipe de gestion.
- En tant qu'administrateur, je souhaite consulter la liste de tous les utilisateurs, afin de mieux gérer les accès.
- En tant qu'administrateur, je souhaite consulter le détail d'un utilisateur, afin de connaître ses informations et son statut.
- En tant qu'administrateur, je souhaite modifier le rôle d'un utilisateur, afin d'adapter ses droits d'accès.
- En tant qu'administrateur, je souhaite activer ou désactiver un utilisateur, afin de contrôler l'accès au système.

Rôles

- En tant qu'**admin**, je souhaite créer un nouveau rôle (ex : support), afin de définir des permissions spécifiques.
- En tant qu'**admin**, je souhaite consulter la liste des rôles existants, afin d'en avoir une vue d'ensemble.
- En tant qu'**admin**, je souhaite consulter un rôle existant, afin d'en avoir une vue d'ensemble.
- En tant qu'**admin**, je souhaite modifier un rôle, afin d'adapter l'organisation du système.

Statuts de réservation

- En tant qu'**admin**, je souhaite créer et modifier les statuts de réservation, afin de refléter les différentes étapes du processus.
- En tant qu'**admin**, je souhaite modifier le statut d'une réservation, afin d'assurer un suivi précis.

Types d'activités

- En tant que **public**, je souhaite consulter la liste des types d'activités, afin de mieux orienter mes choix.
- En tant qu'**admin**, je souhaite créer et modifier un type d'activité, afin d'adapter l'offre du parc aux besoins.

Activités

- En tant que **public**, je souhaite consulter toutes les activités avec des filtres (type, durée, tarif...), afin de choisir une activité adaptée.
- En tant que **public**, je souhaite consulter les détails d'une activité, afin de m'informer avant de réserver.
- En tant que **public**, je souhaite consulter tous les quotas en cours et les saturations prévues d'une activité, afin d'anticiper ma réservation.
- En tant qu'**admin**, je souhaite consulter, créer ou modifier une activité, afin de gérer l'offre disponible.

G. Réservations

- En tant que **client**, je souhaite créer une réservation en précisant une date et un commentaire optionnel, afin d'organiser ma visite.
- En tant que **client**, je souhaite associer une ou plusieurs activités à une réservation, en précisant la date, l'heure et le nombre de participants, afin de personnaliser mon séjour.
- En tant que **client**, je souhaite consulter le statut de mes réservations, afin de connaître leur avancement.
- En tant que **client**, je souhaite consulter une réservation, afin de préparer mon séjour.
- En tant que **client**, je souhaite annuler une réservation, afin de rester flexible dans mon organisation.
- En tant que **client**, je souhaite modifier ou annuler une activité de ma réservation, afin de gérer les imprévus.
- En tant qu'**admin**, je souhaite consulter toutes les réservations afin de piloter les flux de visiteurs.
- En tant qu'**admin**, je souhaite consulter les détails d'une réservation, afin de mieux la gérer.
- En tant qu'**admin**, je souhaite consulter toutes les activités d'une réservation, afin de piloter les flux de visiteurs.
- En tant que **système**, je souhaite empêcher la réservation d'activités en doublon à la même date pour un même utilisateur, afin d'éviter les conflits de planning.

Quotas

- En tant que **système**, je souhaite vérifier les quotas horaires et journaliers lors de la création ou modification d'une réservation, afin de respecter la capacité du parc.
- En tant qu'**admin**, je souhaite créer les quotas horaires et journaliers par activité, afin de contrôler la fréquentation.
- En tant qu'**admin**, je souhaite consulter tous les quotas, afin d'anticiper la gestion des flux de visiteurs.
- En tant qu'**admin**, je souhaite consulter les quotas en cours pour une activité, afin d'anticiper la gestion des flux de visiteurs.
- En tant qu'**admin**, je souhaite modifier les quotas en cours pour une activité, afin de réajuster la gestion des flux de visiteurs.

2.3.ENDPOINT

Un **endpoint** (ou point de terminaison) est une **URL** exposée par une API REST à laquelle un client (application front-end, mobile, script, etc.) peut envoyer des requêtes HTTP pour accéder à une ressource ou déclencher une action.

L'équipe de développement vous fournit une liste **complète** de tous les endpoints, organisée par domaine et avec méthodes, routes, rôles et descriptions :

Authentification & Utilisateurs

| Méthode | Endpoint | Rôle | Description |
|---------|------------------|--------|-------------------------------------|
| POST | /api/auth/signup | Public | Inscription d'un nouvel utilisateur |
| POST | /api/auth/login | Public | Connexion |

| Méthode | Endpoint | Rôle | Description |
|---------|-------------------------------|--------|---|
| GET | /api/users/me | Client | Voir ses informations |
| PUT | /api/users/me | Client | Modifier ses informations personnelles |
| PUT | /api/users/me/password | Client | Modifier son mot de passe |
| PUT | /api/users/me/deactivate | Client | Désactiver son compte |
| POST | /api/users | Admin | Créer un utilisateur (avec rôle) |
| GET | /api/users | Admin | Lister tous les utilisateurs (filtrable par rôle) |
| GET | /api/users/{user_id} | Admin | Voir un utilisateur |
| PUT | /api/users/{user_id}/role | Admin | Modifier le rôle d'un utilisateur |
| PUT | /api/users/{user_id}/activate | Admin | Activer / désactiver un utilisateur |

Rôles

| Méthode | Endpoint | Rôle | Description |
|---------|----------------------|-------|-----------------------|
| POST | /api/roles | Admin | Créer un rôle |
| GET | /api/roles | Admin | Lister tous les rôles |
| GET | /api/roles/{role_id} | Admin | Voir un rôle |
| PUT | /api/roles/{role_id} | Admin | Modifier un rôle |

Statuts de réservation

| Méthode | Endpoint | Rôle | Description |
|---------|---------------------------------------|-------|--------------------------------|
| POST | /api/reservation-statuses | Admin | Créer un statut de réservation |
| GET | /api/reservation-statuses | Admin | Lister tous les statuts |
| PUT | /api/reservation-statuses/{status_id} | Admin | Modifier un statut |

Types d'activités

| Méthode | Endpoint | Rôle | Description |
|---------|--------------------------|--------|-----------------------------------|
| GET | /api/activity-types | Public | Lister tous les types d'activités |
| POST | /api/activity-types | Admin | Créer un type d'activité |
| PUT | /api/activity-types/{id} | Admin | Modifier un type d'activité |

Activités

| Méthode | Endpoint | Rôle | Description |
|---------|--|--------|--|
| GET | /api/activities | Public | Lister toutes les activités (filtres : type, durée, tarif) |
| GET | /api/activities/{id} | Public | Voir le détail d'une activité |
| GET | /api/activities/{activity_id}/availability | Public | consulter la disponibilité des quotas des activités |
| POST | /api/activities | Admin | Créer une activité |
| PUT | /api/activities/{id} | Admin | Modifier une activité |

Réervations

| Méthode | Endpoint | Rôle(s) | Description |
|---------|--|----------------|--|
| POST | /api/reservations | Client | Créer une réservation (date, commentaire) |
| GET | /api/reservations/my | Client | Lister ses réservations |
| GET | /api/reservations/{reservation_id} | Client / Admin | Voir une réservation (Client → la sienne, Admin → n'importe laquelle) |
| PUT | /api/reservations/{reservation_id} | Client | Modifier sa propre réservation |
| PUT | /api/reservations/{reservation_id}/status | Client / Admin | Modifier le statut d'une réservation (Le client ne peut que modifier le statut de sa propre réservation) |
| GET | /api/reservations/{reservation_id}/activities | Client / Admin | Voir les activités d'une réservation (selon rôle) |
| PUT | /api/reservations/{reservation_id}/activities/{act_id} | Client / Admin | Modifier une activité dans sa propre réservation |
| DELETE | /api/reservations/{reservation_id}/activities/{act_id} | Client | Supprimer une activité d'une propre réservation |
| GET | /api/reservations | Admin | Lister les réservations |

Quotas

| Méthode | Endpoint | Rôle | Description |
|---------|------------------|-------|-------------------------------------|
| GET | /api/quotas | Admin | Lister tous les quotas |
| GET | /api/quotas/{id} | Admin | Voir le détail d'un quota |
| POST | /api/quotas | Admin | Créer un quota (horaire/journalier) |
| PUT | /api/quotas/{id} | Admin | Modifier un quota |

2.4. DEVELOPPEMENT BACK-END DE L'API REST

Mettre en place les endpoints RESTful accessibles au rôle « Public / Client » pour couvrir les besoins suivants :

1. Gestion de la création de compte et de l'authentification
2. Consultation des types d'activités et des activités
3. Gestion des quotas journaliers et horaires lors de la création de réservations
4. Création, modification et consultation des réservations
5. Suivi des statuts de réservation

Remarque : seuls les endpoints listés en **rouge** dans la section 2.3 doivent être implémentés pour ce module.

Exigences techniques :

1. Contrôle des données :
 - Assurez-vous que l'application envoie le bon type de données à la base de données(chiffres, lettres, dates, etc.).
2. Gestion des erreurs :
 - Renvoyez les bons messages à l'application en cas de problème.
3. Authentification :
 - Assurez-vous que seules les bonnes personnes peuvent utiliser l'API. Le login doit être l'email de l'utilisateur et le mot de passe par défaut est "P@ssw0rd".
4. Bonnes pratiques :
 - Référez vous au document "Web API Design Best Practices.pdf" pour vous aider.
 - Toutes les validations côté client doivent également être effectuées côté API pour assurer l'intégrité des données.

Livrables :

Code source de l'API + README dans un ZIP.

- **Nom du fichier :** Module2_API_X.zip.(X étant le numéro du compétiteur)

2.5. ÉCRIRE UN TEST UNITAIRE

Les tests unitaires constituent une étape cruciale dans la conception et la mise en œuvre des logiciels. Ils améliorent non seulement l'efficacité du code, mais rendent également le code plus robuste et réduisent les régressions lors du développement et de la maintenance ultérieurs.

Objectif :

Vérifier que certaines routes de l'API accessibles au **rôle Client** respectent les droits d'accès définis. Le client ne doit **pas pouvoir** effectuer d'actions réservées à l'administrateur (gestion des utilisateurs, rôles, quotas, etc.), mais doit pouvoir :

- consulter ses propres informations et réservations,
- créer et modifier ses propres réservations,
- consulter la liste des activités et leurs disponibilités.

Règles métier à tester :

1. Le client **peut** accéder aux informations de son profil (GET /api/users/me)
 2. Le client **ne peut pas** accéder à la liste complète des utilisateurs (GET /api/users/).
 3. Le client **ne peut pas** créer de rôle utilisateur (POST /api/roles)
 4. Le client **peut** consulter la liste de ses propres réservations (GET /api/reservations/my).
 5. Le client **ne peut pas** consulter la liste de toutes les réservations (GET /api/reservations/).
- Nettoyer la donnée après le test pour ne pas laisser de traces dans la base
 - Documenter la démarche dans le Readme.txt avec :
 - Comment exécuter le test.
 - Les attentes et messages d'erreur.
 - La logique de nettoyage après test.

2.6.DOCUMENTATION DE L'API

Créer / Générer une documentation détaillée et exploitable pour les développeurs, incluant :

1. Une collection Postman ou une documentation Swagger décrivant chaque endpoint, avec méthode, paramètres, réponses, exemples JSON et messages d'erreur.

OU

2. Des exemples de requêtes et de réponses au format JSON.

Pour chaque endpoint, fournir :

- Méthode HTTP
- URL de l'endpoint : chemin d'accès à la ressource
- Description concise : rôle ou fonction du endpoint
- Corps de la requête (JSON), si applicable
- Exemples de réponses JSON pour les cas de succès et d'erreur, avec les codes HTTP associés
- Paramètres éventuels : dans l'URL ou en query string, s'ils sont requis

Livrables :

Documentation exhaustive de l'API

- **Nom du fichier** : Module2_API_X_Documentation (X étant le numéro du compétiteur)