Laboratoire: Architecture N\_Tier

# Objectif(s)

* Structurer un projet multi-tier

# Créer La branche pour la séance

## Création de la branche de la séance

1. Dans Gitkraken, clonez votre repository de laboratoire
2. Créez la branche de la séance **S12\_N\_Tier**

# Structurer la solution

## Déplacer les éléments existants

1. Déplacez les **interfaces** des Repositories et du UnitOfWork (seulement les interfaces) dans un dossier **Repositories**, dans le projet **Jungle\_Models**
2. Corrigez les références

# Ajouter des fonctionnalités services

## Fonctionalité Travel\_reservation

Ajoutez des classes de models, des propriétés ou des annotations si nécessaire.

Si ce n’est pas déjà fait :

1. Ajoutez un projet **Jungle\_Service** (blibliothèque de classe)
2. Ajoutez une classe dans le projet appelée **Reservation\_Service**
3. Ajoutez le code (1 ou plusieurs méthodes) permettant au Customer de réserver un voyage :
   * Pour un nombre de personnes donné qui ne doit pas dépasser le nombre de places restantes
   * Pour les voyages dans le futur (minimum 2 semaines)

## Fonctionalité Travel\_Recommendation

1. Ajoutez une classe dans le projet appelée **Recommendation\_Service**
2. Ajoutez le code (1 ou plusieurs méthodes) permettant au Counseller d’ajouter/modifier/supprimer une recommendation pour un Travel
   * 1 seule recommendation par Travel
   * Pour les Travels à partir d’aujourd’hui (pas passé)

# Projet web api

NOTE : vous pouvez faire la partie client avec le framework de votre choix (Agular ou autre), faites des interfaces client de base.

## Configurer le projet Web API

1. Ajoutez un projet Web API appelé **Jungle\_API**
2. Ajoutez les liaisons nécessaires afin que ce projet utilise les éléments (models, repositories, DbContext, etc) existants

## Fonctionnalités

1. Visualiser les travels (customer)
2. Réserver un travel (customer)
3. Ajouter une recommendation sur un Travel (Counseller)