

Mise en contexte :

1. Le projet vise à mettre en œuvre l'**interface applicative (API)** qui sera utilisée par votre application Angular. L'API sera intégrée au projet #1 (30%) afin de mettre en œuvre l'application finale visée par le projet #3 (40%).
2. Voici ce qui vous est demandé :
 - a. Votre API doit offrir la possibilité de gérer les vidéos de votre application.
 - b. Toutes les opérations standard (CRUD) doivent être gérées. Lors de sa remise, votre API doit permettre:
 - i. **D'afficher** la liste complète des vidéos
 - ii. **D'afficher** une vidéo en particulier (selon son identifiant)
 - iii. **D'ajouter** une vidéo
 - iv. De **modifier** une vidéo
 - v. De **supprimer** une vidéo
 - vi. **D'afficher les commentaires** d'une vidéo en particulier (selon son identifiant)
 - vii. **D'ajouter un commentaire** sur une vidéo en particulier (selon son identifiant)
 - viii. **D'ajuster le score (à la hausse ou à la baisse)** d'une vidéo en particulier (selon son identifiant)
 - c. Vous êtes libres de structurer la base de données et le format JSON de l'API comme vous le désirez. Celle-ci doit contenir **au minimum 5 enregistrements** dans chacune de vos tables « principales ».

Voici tout de même mes conseils :

- i. Stockez les données des vidéos dans une seule table dans une base de données MySQL.
- ii. Ajoutez une deuxième table pour les avis
- iii. Créez votre API en PHP et assurez-vous que celui-ci envoie et reçoit les informations au format proposé dans le Projet #1 (voir [Exemple d'une vidéo JSON](#) à la page 3).
- iv. Nous reviendrons sur les techniques à utiliser pour générer du json avec des éléments imbriqués (avoir des tableaux ou des objets dans des objets). Ce n'est pas évalué dans ce projet. Si vous voulez en savoir plus, vous pouvez consulter cet exemple sur GitHub : <https://github.com/Techniques-de-programmation-3/exemple-api-cegeps>

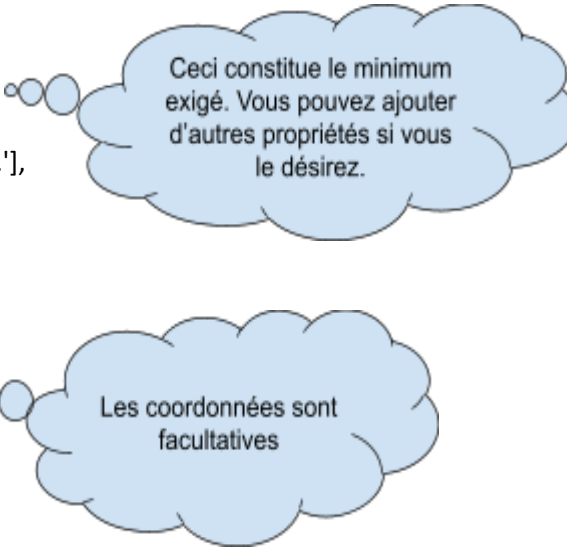
Si vous préférez utiliser d'autres technologies ou un autre format JSON, vous êtes responsable :

- v. D'intégrer des objets et tableaux imbriqués dans votre format JSON.
 - vi. D'intégrer les technologies non couvertes dans le cours Infrastructure Web sans garantie d'obtenir un support pédagogique sur les technologies non couvertes dans le programme d'études.
 - vii. D'adapter votre application lors de la mise en oeuvre du Projet #3 pour que celle-ci soit en mesure d'interagir avec votre API.
 - viii. De fournir une procédure d'installation simple et rapide OU un environnement me permettant d'évaluer votre projet et d'en tester son fonctionnement.
- d. Vous devez également démontrer le bon fonctionnement des requêtes effectuées à l'API. À cet effet, votre remise devra contenir un dossier « captures » qui contiendra les captures suivantes, **numérotées** (je dois **clairement** voir l'**URL**, les **données en entrées** si applicable, et le **résultat** de la requête) :
1. **Affichage** de la **liste des vidéos** (minimum 5) initialement présents dans la base de données, affichées au format JSON dans un **navigateur Web** (Chrome, Firefox, Safari, etc.)
 2. **Affichage** de **première vidéo** de la liste **capturée au #1, selon son identifiant** (id), affichée au format JSON dans un **navigateur Web** (Chrome, Firefox, Safari, etc.)
 3. **Ajout d'une vidéo** avec données en entrée **valides** (format JSON), dans une application permettant de tester des API (Postman, Thunder Client, etc.)
 4. **Ajout d'une vidéo** avec données en entrée **invalides** (format JSON), dans une application permettant de tester des API (Postman, Thunder Client, etc.)
 5. **Édition de la vidéo capturée au #2** avec données en entrée **valides** (format JSON), dans une application permettant de tester des API (Postman, Thunder Client, etc.)
 6. **Édition de la vidéo capturée au #2** avec données en entrée **valides** (format JSON), **sans identifiant dans l'URL**, dans une application permettant de tester des API (Postman, Thunder Client, etc.)
 7. **Édition de la vidéo capturée au #2** avec données en entrée **invalides** (format JSON), dans une application permettant de tester des API (Postman, Thunder Client, etc.)
 8. **Suppression de la seconde vidéo** de la liste **capturée au #1**, dans une application permettant de tester des API (Postman, Thunder Client, etc.)
 9. **Suppression d'une vidéo sans identifiant**, dans une application permettant de tester des API (Postman, Thunder Client, etc.)
 10. **Affichage** des commentaires **de la vidéo capturée au #2**, dans une application permettant de tester des API (Postman, Thunder Client, etc.)
 11. **Ajout d'un commentaire de la vidéo capturée au #2**, dans une application permettant de tester des API (Postman, Thunder Client, etc.)
 12. **Ajustement du score de la vidéo capturée au #2**, dans une application permettant de tester des API (Postman, Thunder Client, etc.)

13. **Affichage** de la **liste des vidéos** suites aux opérations précédemment effectuées, affichées au format JSON dans une application permettant de tester des API (Postman, Thunder Client, etc.)
14. **Affichage** des autres données que vous avez ciblées, affichées au format JSON dans une application permettant de tester des API (Postman, Thunder Client, etc.)

Exemple d'une vidéo JSON (format proposé) :

```
{
  nom: 'Vidéo ABC',
  description: 'Lorem ipsum....',
  code: 'ABC000',
  categories: ['horreur', 'humour', '...'],
  auteur: {
    nom : 'nom...',
    utilisateur: '@utilisateur',
    verifie: false,
    coordonnees : {
      courriel: '...',
      facebook: '...',
      instagram: '...',
      twitch: '...',
      site_web : '...'
    },
    description : '...'
  },
  date_publication : '2023-01-01',
  duree : 125,
  nombre_vues: 500,
  score: 2500,
  sous-titres: 'ss',
  avis: [
    {
      note: 10,
      commentaire: '...'
    },
    {
      note: 0,
      commentaire: '...'
    }
  ]
}
```



Dates de remise :

- ✓ Du temps sera accordé dans les périodes du 9 et 16 janvier pour réaliser le projet.
- ✓ Le projet doit être remis sur **Github** avant le **30 janvier à 23h59**.

À remettre:

- ✓ Lien GitHub du projet, contenant:
 - o Un script de la base de données, avec quelques enregistrements (minimum 5 par table « principale »)
 - o Le code source de votre API
 - o Les captures d'écran prouvant le bon fonctionnement de votre API

Grille d'évaluation:

Critère	Pondération
1. Détermination correcte des tâches à effectuer	/ 2
2. Utilisation correcte des différents langages et outils de développement (Serveur Web, Navigateur Web, Application permettant de tester des API, langage de programmation interprété côté serveur, etc.)	/ 4
3. Programmation appropriée de l'interface applicative (2 points par commande, 2 points pour l'utilisation la page index.html pour l'ensemble des méthodes)	/ 18
4. Information adéquatement sécurisée (utilisation de requêtes préparées et validation des données reçues)	/ 6
NOTE	/ 30

Pénalités lors de la remise des travaux :

Chaque journée de retard entraîne une pénalité de 10 %.

Aucun travail ne peut être remis après 7 jours de retard, la note zéro (0) sera alors attribuée.