***Grille d’évaluation du projet final***

1. Développement de votre API et utilisation de méthodes de requêtes http (GET, PUT, POST, DELETE).

-Retrouver dans: FULLSTACKHEARTSTONE>HeartStone>my-heartstone-api

-Base Donnée SQL Server vers Node.js

-Retrouver dans: FULLSTACKHEARTSTONE>HeartStone>my-heartstone-api>Data>CartesEvents>index.js

-Node.js vers Express

-Retrouver dans: FULLSTACKHEARTSTONE>HeartStone>my-heartstone-api>Controler>CartesControler.js

1. Développement d’un client qui consomme votre API créée de type application bureau avec la plateforme .NET (Projet WPF par exemple).

-Retrouver dans: FULLSTACKHEARTSTONE>HeartStone>my-heartstone-app

-Express vers React

-Retrouver dans: FULLSTACKHEARTSTONE>HeartStone>my-heartstone-app>src>Page>PageHearthStone.js.

1. Développement d’un deuxième client qui consomme votre API créée de type application Web (Projet REACT par exemple).

-Retrouver dans le laboratoire final du Cours Dev. Application Console

1. Utilisation des principes de *« routing »* côté client et serveur.

- Retrouver dans: FULLSTACKHEARTSTONE>HeartStone>my-heartstone-app>src>Page>PageHearthStone.js.

1. Test de votre API avec Postman ou REST Client par exemple –écrire au moins 10 tests.

-Installer extension rest

- Retrouver dans: FULLSTACKHEARTSTONE>HeartStone>my-heartstone-api>testRest.rest

-Test Postman

-Retrouver dans : <https://documenter.getpostman.com/view/24381246/2s93JnVSdu#e32700e7-0628-4227-b44f-8fc60bd130f1>

6-Organisation de votre application et respect des bonnes pratiques.

-Chaque application est décomposer de façon à être le plus claire et simple possible, pour pouvoir permettre de future collaboration.

-my-heartstone-api est composé des répertoires suivants :

-Controller

-Data

-Routes

-Tous les fichiers de configuration sont retrouvées dans le répertoire

my-heartstone-api

--my-heartstone-app est composé des répertoires suivants :

-src

-Components

-NavigationPrincipale

-Pages

-Ressource

-UI

7-Gestion des erreurs pour les requêtes défectueuses

-Retrouver dans: FULLSTACKHEARTSTONE>HeartStone>my-heartstone-api>Data>CartesEvents>index.js

-Retrouver dans: FULLSTACKHEARTSTONE>HeartStone>my-heartstone-api>Controler>CartesControler.js

-Retrouver dans: FULLSTACKHEARTSTONE>HeartStone>my-heartstone-app>src>Page>PageHearthStone.js.

8- Ajout de commentaires à votre code.

- Des commentaires pertinent sont retrouver aux débuts de chaque module

-Expliquant la responsabilité de chacun de c’est module

9-Génération de la documentation avec « Swagger » par exemple.

-Documentation Postman

--Retrouver dans : <https://documenter.getpostman.com/view/24381246/2s93JnVSdu#e32700e7-0628-4227-b44f-8fc60bd130f1>

10- Présentation orale du projet (Présentation de l’ API ) et des perspectives de votre projet. Par exemple, les étudiantes peuvent critiquer leur travail et présenter les parties à améliorer dans le futur.

***Guide du dévelopeur***

-Documentation Postman retrouver dans : <https://documenter.getpostman.com/view/24381246/2s93JnVSdu#e32700e7-0628-4227-b44f-8fc60bd130f1>

Types d'API

Privé

Une API privée est utilisée en interne au sein d'une équipe, créée par des développeurs qui travaillent pour l'entreprise ou l'organisation. Cela peut aider à créer des API hautement personnalisées pour les besoins de votre entreprise et à adapter l'API au besoin. Les API privées ne peuvent pas être utilisées ou travaillées par des développeurs externes, car elles sont uniquement destinées à votre équipe interne. Ces API ont tendance à se concentrer sur l'aide à la productivité et à la gestion de projet de votre entreprise sur plusieurs applications.

Url de base : http://localhost:4000/api

Endpoints disponibles pour consommations

GET

-http://localhost:4000/api/GetCartes

-http://localhost:4000/api/GetCarte/:id

POST

-http://localhost:4000/api/AjouterCarte

PUT

-http://localhost:4000/api/UpdateCarte/:id

DEL

-http://localhost:4000/api/DeleteCarte/:id

### GETGetTouteLesCartes

Cette Opération GetTouteLesCarte, vous permet de retrouver toutes les cartes qui se retrouvent dans la base de Donnée HearthStone.

Le URL est :

<http://localhost:4000/api/GetTouteLesCartes>

-Une réponse Json est reçu avec tous les cartes.

-status code 200, si la requête est bien exécuter

-status code 404, si une erreur survient

les Headers dans la réponse reçu sont :

Id : int auto incrementer

Rareter: String

Nom : String

Cout : int

Vie: int

Attack: int

### GETGetUneCarte

Cette Opération GetUneCarte, vous permet de retrouver Une carte qui se retrouve dans la base de Donnée HearthStone.

Le URL est :

<http://localhost:4000/api/GetCarte/:id>

paramètre:

id: Int

L'Utilisateur dois entrer le chiffre qui correspond à l'Id de la carte qu'il recherche.

Pour Voir Les Cartes disponibles se référé à l'opération GetTouteLesCartes

-Une réponse Json est reçu avec la carte désiré.

-status code 200, si la requête est bien exécuter

-status code 404, si une erreur survient

Les Headers dans la réponse reçu sont :

Id : int auto incrementer

Rareter: String

Nom : String

Cout : int

Vie: int

Attack : int

### POSTAjouterUneCarte

Cette Opération AjouterUneCarte, vous permet d'ajouter une nouvelle carte, à la base de données HearthStone.

Le URL est :

[http://localhost:4000/api/](http://localhost:4000/api/GetCarte/:id)AjouterCarte:

L'Utilisateur dois fournir un array d'array un object Json dans le body de la demande.

-Les Headers suivant doivent se retrouver dans l'object Json:

Rareter: String

Nom: String

Cout: int

Vie: int

Attack: int

-Une réponse Json est reçu avec le id de la carte qui vous avez ajouté à la base de données.

-statuts code 200, si la requête est bien exécutée

Les Headers dans la réponse reçu sont :

Id : int

### PUTUpdateUneCarte

Cette Opération UpdateUneCarte, vous permet modifier une carte existante, dans base de données HearthStone.

Le URL est:

<http://localhost:4000/api/UpdateCarte/:id>

paramètre:

id: int

L'Utilisateur dois entrer le chiffre qui correspond à l'Id de la carte qu'il aimerait modifier.

L'Utilisateur dois fournir un array d'array un object Json dans le body de la demande.

-Les Headers suivant doivent se retrouver dans l'object Json:

Id: Int

Rareter: String

Nom: String

Cout: int

Vie: int

Attack: int

-Les Headers suivant doivent se retrouver dans l'object Json:

Id: int

Rareter: String

Nom: String

Cout: int

Vie: int

Attack: int

-Une réponse Json est reçu avec la carte modifier dans la base de donner HearthStone.

-status code 200, si la requête est bien exécutée

-status code 404, si la le id ne correspond pas à une carte existante dans la base de donnée.

les Headers dans la réponse reçu sont :

Id : int auto incrementer

Rareter: String

Nom : String

Cout : int

Vie: int

Attack: int

### DELETEDeleteUneCarte

Cette Opération DeleteUneCarte, vous permet modifier une carte existante, dans base de données HearthStone.

Le URL est :

<http://localhost:4000/api/DeleteCarte/:id>

paramètre:

id: int

L'Utilisateur dois entrer le chiffre qui correspond à l'Id de la carte qu'il aimerait modifier.

Réponse:

-Status code 200, si la requête est bien exécutée

-Status code 404, si le id ne correspond pas à une carte existante dans la base de donnée.