



Collège de
Maisonneuve

Automne 2022

Plan de cours

420-941-MA gr. 21624

Services Web

Programme : LEA.8F Sécurité des applications mobiles, Web et de bureau
Formation continue

Pondération : 2-4-3

Préalables : 420-921-MA Base de données
420-931-MA Programmation Web Front-End

Professeur : Antonio Tavares

Courriel : atavares@cmaisonneuve.qc.ca

Présentation du cours

Le cours sur les services Web couvre les composants logiciels qui gèrent les interactions des programmes logiciels distants communiquant sur le Web. Après avoir suivi ce cours, les étudiants devraient être en mesure de décrire l'interfonctionnement des protocoles Web, de concevoir et de développer des interfaces de programmation d'applications Web (API), des services Web et le code client qui interagit avec ces services. Les étudiants devraient également être capables de préparer l'environnement de développement informatique, de déployer et de documenter les services Web, et de contrôler la qualité du code développé. Ce cours complète la formation des étudiants aux compétences en programmation Web en étendant leurs compétences en programmation côté serveur pour couvrir des sujets liés à la conception, à la mise en œuvre et à la sécurisation des services Web, y compris les communications de processus à processus.

Ce cours complète la formation des étudiants aux compétences en programmation Web en étendant leurs compétences en programmation côté serveur pour couvrir des sujets liés à la conception, à la mise en œuvre et à la sécurisation des services Web, y compris les communications de processus à processus. Ce cours est préalable au cours 420-951-MA Applications Web Transactionnelles.

Compétence(s) développée(s) dans ce cours :

NO	Énoncé	Éléments
00SV	Développer des services d'échange de données.	<ul style="list-style-type: none">• Analyser le projet de développement de l'application• Préparer l'environnement de développement informatique• Préparer la base de données• Programmer la logique d'application pour le service• Programmer une application de mise à l'essai en utilisant le service• Contrôler la qualité du service• Participer au déploiement du service• Produire la documentation

À la fin du cours comme dernier travail pratique

Dans un environnement d'équipe, les étudiants conçoivent, implémentent et testent un service Web de leur propre pensée et conception. Ces services Web sont déployés sur des comptes de service cloud gratuits ou sur un serveur local offrant des capacités de test similaires. Les services sont testés à l'aide d'un service d'intégration continue (CI). Les étudiants documentent leur API de service Web dans un rapport final et présentent leur implémentation finale à tous les étudiants de la classe.

Déroulement du cours

Semaine	Théorie	Laboratoire
1 à 2	<p>Installation, configuration et administration des outils de développement.</p> <p>Outils</p> <ul style="list-style-type: none"> • Visual Studio Code • Node • Ts-node • Nodemon • Express • Typescript • Librairie RxJS <p>Structure d'une application côté serveur</p> <ul style="list-style-type: none"> • Dossiers du projet • Modèle MVC (Modèle-Vue-Contrôleur) 	<p>Lectures, recherches internet</p> <p>Installation de l'environnement</p> <p><u>Exercices formatifs</u></p>
3 à 6	<p>Langages de programmation côté serveur</p> <p>Vue d'Ensemble de ces langages et leur utilisation</p> <ul style="list-style-type: none"> • JavaScript ES6 • Utilisation de RxJS • Typescript <p>Programmation asynchrone</p> <ul style="list-style-type: none"> • Callbacks • Promises • Async/await <p>Structure d'une application côté serveur</p> <ul style="list-style-type: none"> • Dossiers du projet • Modèle MVC (Modèle-Vue-Contrôleur) <p>Modèle de communication client/serveur</p> <ul style="list-style-type: none"> • Protocole HTTP (requête/réponse HTTP) • Validation des réponses http • Création d'une requête HTTP <ul style="list-style-type: none"> ○ AJAX ○ Fetch ○ JSON ○ XML 	<p><u>Exercices formatifs</u></p> <p><u>Examen 1</u></p> <p><u>Laboratoire 1</u></p>

7 à 8	<p>Services Web (vue d'ensemble)</p> <ul style="list-style-type: none"> • Définition, contexte et objectif • Introduction à la communication IPC (Communication inter-processus par socket) • SOA vs ROA (architecture orientée services vs ressources) • Différences entre SOAP et RESTful • Émergence de GraphQL et différence/complémentarité avec RESTful • Modèle de messagerie, espaces de noms, SOAP sur HTTP, enveloppe, en-tête et corps du message, erreurs SOAP • Langage de description des services : WSDL, UDDI, modèle d'informations WSDL, sémantique des services, description des messages • Normes liées aux technologies des services Web (groupe de travail sur l'architecture des services Web – un sous-comité du W3C) • ROA-ressources Web, schéma d'URL, types de médias 	<p><u>Exercices formatifs</u></p> <p>Laboratoire 2</p> <p><u>Examen 2</u></p>
9 à 15	<p>Conception et implémentation d'API RESTful</p> <ul style="list-style-type: none"> • Points de terminaison publics, transmission de paramètres dans les appels d'API et liste avec HATEOS • Concevoir et mettre en œuvre des API RESTful qui fournissent différents types de ressources (texte brut et flux binaires) • Programmation modulaire dans le contexte des services Web : <ul style="list-style-type: none"> ○ JSON en sortie ○ Routage ○ Contrôleurs recevant une entrée JSON ○ Modèles et conception appropriée de la base de données • Réception et traitement des requêtes client • Composition de services Web (alias mashup) • Intégration tierce (par exemple, CDN) • Si le temps le permet, services Web dans des environnements de micro services 	<p><u>Laboratoire 3</u></p> <p><u>ÉXAMEN DE SYNTHÈSE</u></p>

Activités d’enseignement et d’apprentissage

Les principales méthodes pédagogiques utilisées dans ce cours sont :

Des exercices formatifs.

Travaux pratiques, réalisés en classe et en dehors des heures de cours prévues à l’horaire, Ces travaux pratiques sont réalisés soit individuellement soit en équipe.

Évaluation formative

L’évaluation formative se fera à l’aide d’exercices formatifs réalisés en classe ou en dehors des heures de cours.

Évaluation sommative

L’évaluation sommative s’effectuera au moyen de trois examens et de trois travaux pratiques.

Laboratoires 1	10%
Laboratoires 2	15%
Laboratoires 3	15%
Examen 1	15% *
Examen 2.....	15% *
Épreuve finale	30% *

* Les évaluations annotées d’une étoile (*) font partie de l’ensemble des évaluations dont le seuil de passage pour l’ensemble de ces évaluations est de 50%. Ces évaluations se font de manière individuelle sous environnement contrôlé, en classe.

CRITÈRES D’ÉVALUATION DE L’ÉPREUVE FINALE

L’épreuve finale, qui compte pour 30% de la note du cours, sera constituée d’un l’examen pratique en classe dont les étudiants ont droit à toutes leurs notes de cours.

Critères d’évaluation et pondération (sur 100%)

À partir d’un énoncé sur la création d’un service Web

• Configurer l’environnement de développement	10%
• Création de la base de données	15%
• Création des interfaces côté client	10%
• Gestion des requêtes côté client	15%
• Création d’un API RESTful pouvant fournir plusieurs types de ressources	35%
• Validations, gestion des exceptions et patrons de conception	10%
• Vidéo des tests de l’application	5%

Médiagraphie

Notes de cours en LEA

Sites intéressants pour le cours (d'autres dans LEA)

<https://developer.mozilla.org/fr/docs/Web/JavaScript>

https://lia.disi.unibo.it/materiale/JS/developer.mozilla.org/en-US/docs/Web/JavaScript/New_in_JavaScript/ECMAScript_6_support_in_Mozilla.html

<https://rxjs.dev/guide/overview>

<https://www.learnrxjs.io/>

<https://www.typescriptlang.org/>

<https://code.visualstudio.com/docs/languages/typescript>

https://www.w3schools.com/typescript/typescript_intro.php