

Développement Web : JavaScript & Web API Fetch

Examen

Introduction :

Vous êtes développeur dans une entreprise et on vous demande d'aider le responsable technique en développant toute une suite d'outils permettant d'accélérer le développement d'une application Web.

Objectif :

Savoir manipuler des dates, des expressions régulières, et les API Web DOM & Fetch dans le langage JavaScript

Partie 1 (5 points) : Champ de formulaire

Dans un document HTML, afficher un champ de formulaire permettant de mettre à disposition de vos utilisateurs un champ de texte et un bouton d'envoi dans un formulaire. Ce champ de formulaire doit être vérifié selon les règles suivantes (afficher une seule erreur générique si le champ ne respecte pas ces règles avant l'envoi d'une requête) :

- Ne doit pas être vide
- Doit comporter uniquement des caractères alphabétiques
- Ne doit comporter aucun symboles, hormis des espaces
- Ne doit comporter qu'un seul espace à la fois (plusieurs espaces consécutifs interdits)
- Chaque mot séparés par des espaces doit commencer par une majuscule, les autres caractères en minuscule

Partie 2 (5 points) : Requête asynchrone HTTP

Lorsque le champ de formulaire est valide, envoyer une requête HTTP asynchrone sans recharger la page, et récupérer la météo de la ville entrée dans le champ de texte par le biais de l'API OpenWeatherMap. Vous devez créer un compte avant de pouvoir accéder à cette API. Vous afficherez les informations suivantes dans le format de votre choix (les températures doivent être récupérée en degrés celsius) :

- Condition météorologique (weather.description)
- Une icône sous la forme d'une image (weather.icon)
- La température (main.temp)
- La température ressenti (main.feels_like)
- La date au format français (Y/m/d H:m:s) à laquelle cette requête a été effectuée (dt)
- La date au format français du lever de soleil
- La date au format français du coucher de soleil

Partie 3 (5 points) : Chargement, annulation et erreur

Il doit être possible, via un affichage supplémentaire (animation de chargement, texte de chargement, ...) de pouvoir avoir un état du chargement de la requête asynchrone HTTP. Vous pouvez vous aider des outils de développement afin de simuler une connexion lente.

Il doit également être possible d'annuler la requête HTTP si l'utilisateur le décide (et si la requête est suffisamment lente pour le permettre). Pour cela, un bouton doit être affiché uniquement si la réponse liée à la requête HTTP n'est pas encore arrivée.

Enfin, en cas d'erreur liée à la requête HTTP asynchrone (réseau trop lent, timeout, mauvaise URL, annulation, etc...) une erreur doit être affichée en rouge et petit à un endroit suffisamment visible pour l'utilisateur sur la page.

Bonus (5 points)

- Code d'exemple permettant de tester toutes les fonctions
- Code bien commenté
- Code uniquement en anglais (les commentaires en français)
- Aucune erreur à l'exécution du code
- Récupération automatique de la météo au chargement de la page via l'API Geolocation et les coordonnées GPS (faire une requête en fonction de la latitude et longitude au lieu d'un nom de ville)