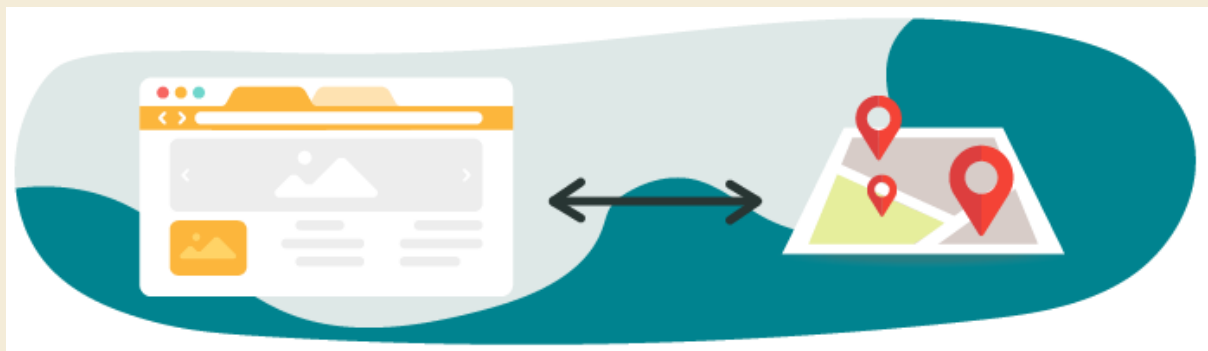


Récupérez des données sur une API interne

5–7 minutes



Avant de commencer ce chapitre, petit rappel : **une API n'est pas forcément liée à un webservice**. Votre navigateur comporte de très nombreuses API internes :

- Le LocalStorage utilise l'API du Web Storage pour stocker des informations sur votre navigateur ;
- L'API Canvas fournit un moyen de dessiner via le JavaScript et le HTML.

Récupérez des données de géolocalisation

Dans le cadre du projet Façadia, nous allons utiliser l'API de géolocalisation sur la page de création d'un formulaire. Vous pouvez trouver [la documentation de cette API sur le MDN](#).



► Technologies► References & Guides► Feedback

Search MDN

Sign in

Technologies web pour développeurs ► Référence Web API ► Geolocation API (API de géolocalisation)

Change language | View in English

Cette page a été traduite à partir de l'anglais par la communauté. Vous pouvez également contribuer en rejoignant la communauté francophone sur MDN Web Docs.

Table of contents

- L'objet `geolocation`
- Décrire une position
- Gérer les erreurs
- Exemple interactif
- Demander la permission
- Compatibilité des navigateurs
- Voir aussi

Related Topics

Geolocation API (API de géolocalisation)

 **Contexte sécurisé**

Cette fonctionnalité est uniquement disponible dans des [contextes sécurisés](#) (HTTPS), pour certains navigateurs qui la prennent en charge.

L'API **Geolocation** (pour géolocalisation) permet à un utilisateur d'indiquer sa localisation à une application web s'il le souhaite. Pour des raisons de vie privée, l'application doit demander la permission à l'utilisateur de manipuler ces informations.



La page du MDN concernant l'API de Géolocalisation

Cela dit, comme toujours, avant de se lancer pleinement dans le code, il est important de découvrir comment fonctionne cette API en dehors du projet. Une fois cette API bien comprise, vous pourrez voir comment elle a été intégrée au projet.

Découvrez l'API de géolocalisation

J'ai une bonne nouvelle à vous annoncer.



La documentation du MDN sur l'API de géolocalisation est très **très** claire ! Elle va vous montrer comment l'intégrer facilement au projet, comment gérer les cas d'erreurs et elle comporte même un exemple : c'est un peu le rêve du développeur !

Dans la vidéo ci-dessous, je vais vous montrer comment utiliser l'API. Voici la branche relative à cette partie : [partie-4/chapitre-2/section-1](#). La page de test se trouve dans le [dossier pages/geolocation.html](#)

Quelques éléments à retenir sur l'API avant de poursuivre :

- L'API de géolocalisation se trouve dans l'objet `navigator.geolocation`. Tous les navigateurs ne la prennent pas en charge.
- Une fois que vous avez pu tester si elle était disponible, la méthode `getCurrentPosition` vous permet de récupérer votre position actuelle. C'est cette méthode que nous allons utiliser dans le projet Façadia.
- Vous pouvez aussi suivre l'évolution de la position grâce à la méthode `watchPosition`.

Dans le cadre de l'API de WeatherStack, on avait mocké les données. Ce n'est pas fait ici, est-ce gênant ?

Bonne question !

Dans le cas de cette API, il n'y a pas de limite de consommation : vous pouvez demander votre position autant de fois que vous le souhaitez. Il n'y a donc pas vraiment besoin de mocker les données.



Intégrez l'API de géolocalisation dans le projet fil rouge

Maintenant que l'API de géolocalisation n'a plus de secrets pour vous, il est temps de s'intéresser à son intégration dans notre projet Façadia ! Pour rappel, nous nous servons de cette API sur la page de création d'un capteur.

La page de création d'un capteur, avec la géolocalisation

Dans la vidéo ci-dessous, je vais vous montrer comment l'API est intégrée sur notre projet. J'utiliserai le code source de la branche main.

Maintenant que vous avez vu comment l'API fonctionnait et était utilisée dans notre projet Façadia, nous pouvons nous intéresser aux messages d'erreur.



Comprenez les messages d'erreur d'une API interne

J'ai une bonne et une mauvaise nouvelle !

La mauvaise nouvelle, c'est que la gestion des erreurs des API internes n'est pas la même d'une API à l'autre. Les messages

d'erreur ne seront pas les mêmes ainsi que les cas d'erreur.

Cela dit, et c'est la bonne nouvelle, la documentation du MDN sur les API internes sera souvent d'excellente facture ; elle vous permettra de bien comprendre comment utiliser l'API et comment traiter les erreurs de façon élégante.

Nous allons voir ici un exemple d'API pour justement vous permettre de mieux comprendre et parcourir la documentation. Je vous invite d'ailleurs à aller en voir d'autres (n'oubliez pas que [cette page vous donne toutes les API internes possibles](#)) et à essayer de comprendre comment les erreurs sont gérées.

Découvrez comment les erreurs sont gérées sur l'API de géolocalisation

Si vous vous rendez sur [la documentation de cette API sur le MDN](#), vous trouverez un paragraphe appelé "Gérer les erreurs".

On vous y explique que vous allez devoir passer une callback dans la méthode `getCurrentPosition`. Cela vous donnera donc le code suivant :

```
// Ma callback de succès
function success(position) {
    // J'ai récupéré la position et les coordonnées
    GPS, je peux faire quelque chose avec
}

// Ma callback d'erreur
function error() {
    // Quelque chose n'a pas marché : je traite de
    façon élégante l'erreur
}

// La méthode getCurrentPosition
navigator.geolocation.getCurrentPosition(success,
```

```
error);
```

Plutôt pratique, non ?

Est-ce obligatoire de traiter les cas d'erreur ?

Oui et même deux fois oui !

L'absence de traitement d'un cas d'erreur peut vous causer de gros soucis : votre programme peut planter et il peut planter de façon silencieuse ce qui sera un cauchemar à déboguer. Donc, s'il vous plaît, faites-vous une fleur et rajoutez la gestion des messages d'erreur.



À vous de jouer !



Avant de passer à la dernière partie de ce cours, c'est l'heure du cas pratique ! Il semblerait que la géolocalisation soit cassée et ne fonctionne plus correctement comme semble l'indiquer [cette issue Github](#). Prenez le temps de la regarder et d'essayer de trouver le bug par vous-même avant de regarder la vidéo ci-dessous.

En résumé

- Comme les API externes, les API internes ont aussi une façon de traiter les erreurs. Le meilleur moyen de découvrir comment elles sont traitées est de parcourir la documentation.
- Le MDN est une excellente source d'information pour comprendre comment fonctionne une API et comment elle gère les cas d'erreur.
- Il est important de traiter vos messages d'erreur ! En effet, un message d'erreur non traité peut compliquer très fortement le débogage.

Vous savez maintenant déboguer des API externes et internes, félicitations ! Il est maintenant temps de passer à la dernière partie du cours : déboguer des applications.