**1. Autorisez les connexions extérieures dans WAMP**

Par défaut, WAMP Server autorise uniquement les connexions locales. Vous devez configurer Apache pour accepter les connexions externes.

1. Ouvrez le fichier de configuration Apache. Il se trouve généralement ici :

bash

Copier le code

c:/wamp64/bin/apache/apacheX.X.X/conf/extra/httpd-vhosts.conf

1. Recherchez la ligne contenant Require local ou Deny from all.
2. Modifiez-la pour autoriser les connexions à partir de toutes les adresses IP. Exemple :

Require all granted

Ou dans certains cas :

Allow from all

**2. Désactivez le pare-feu ou autorisez les connexions au port 80**

Votre pare-feu Windows peut bloquer les connexions entrantes au serveur web.

1. **Vérifiez le port utilisé par WAMP :**
   * Généralement, c'est le **port 80** ou parfois **8080** (en cas de conflit).
2. **Ajoutez une règle dans le pare-feu pour autoriser le port :**
   * Accédez à **Panneau de configuration > Pare-feu Windows Defender > Paramètres avancés**.
   * Cliquez sur **Règles de trafic entrant** > **Nouvelle règle** :
     + Type de règle : Port.
     + Appliquez aux ports **80** (ou le port utilisé par WAMP).
     + Autorisez les connexions.

### Ici cest pour un site normal maintenant voyons pour une application 3. Lancer le serveur Artisan sur l'adresse IP

Lorsque vous démarrez votre serveur Laravel avec la commande php artisan serve, vous pouvez lui spécifier une adresse IP et un port.  
Utilisez l'adresse IP de votre ordinateur (192.168.137.1) :

bash

Copier le code

**php artisan serve --host=192.168.137.1 --port=8000**

Cela rendra votre application accessible depuis d'autres appareils connectés au même réseau, comme votre téléphone.

Ensuite, dans le navigateur de votre téléphone, tapez :

arduino

Copier le code

http://192.168.137.1:8000

**4. Configurer le pare-feu si ce n’est pas fait par defaut par une alerte**

Assurez-vous que le **pare-feu Windows** autorise les connexions sur le **port 8000** :

1. Ouvrez **Paramètres avancés du pare-feu Windows Defender**.
2. Cliquez sur **Règles de trafic entrant > Nouvelle règle**.
3. Créez une règle pour le **port 8000** :
   * Type de règle : Port.
   * Port spécifique : 8000.
   * Autoriser la connexion.
   * Appliquer à tous les profils (Domaine, Privé, Public).
4. Appliquez et redémarrez votre pare-feu.

Oui, tout cela peut fonctionner si votre PC et votre téléphone sont connectés au même réseau (même point d'accès), mais il y a quelques ajustements à vérifier pour s'assurer que tout fonctionne correctement.

### Ce qu'il faut vérifier dans ce scénario :

#### 1. **Adresse IP de votre PC**

* Lorsque votre PC est connecté à un réseau autre que le point d'accès qu'il crée, il aura une nouvelle adresse IP, attribuée par le routeur ou le point d'accès Wi-Fi.
* Pour connaître l'adresse IP de votre PC dans ce réseau :
  1. Ouvrez une invite de commande (CMD) et tapez :

bash

Copier le code

ipconfig

* 1. Trouvez la **Carte réseau sans fil Wi-Fi** et notez l'**Adresse IPv4** (par exemple, 192.168.1.102).

#### 2. **Utiliser cette nouvelle adresse IP**

* Sur votre téléphone, dans le navigateur, utilisez cette nouvelle adresse IP pour accéder à votre application Laravel :

arduino

Copier le code

http://[Adresse\_IP\_du\_PC]:8000

Par exemple :

arduino

Copier le code

http://192.168.1.102:8000

#### 3. **Pare-feu Windows**

* Si votre téléphone et votre PC sont sur le même réseau mais que vous ne pouvez pas accéder à l'application, vérifiez que le pare-feu Windows n'empêche pas les connexions :
  1. Autorisez le port utilisé par Laravel (par défaut, **8000**) comme expliqué précédemment :
     + Accédez à **Pare-feu Windows Defender > Paramètres avancés > Règles de trafic entrant**.
     + Ajoutez une nouvelle règle pour autoriser les connexions entrantes sur le port **8000**.

#### 4. **Configuration de Laravel**

Si vous utilisez **PHP Artisan Serve**, vous devez relancer le serveur en utilisant l'adresse IP de votre PC (pas localhost). Par exemple :

bash

Copier le code

php artisan serve --host=192.168.1.102 --port=8000

#### 5. **Problèmes liés à Vite ou aux assets**

* Si vous utilisez Vite (ou Mix), vérifiez que les fichiers CSS et JS sont correctement générés et accessibles via l'adresse IP :
  + Mettez à jour le fichier vite.config.js (si nécessaire) pour utiliser l'adresse IP :

javascript

Copier le code

export default defineConfig({

server: {

host: '0.0.0.0', // Pour écouter toutes les connexions réseau

port: 5173, // Port de développement utilisé par Vite

}

});

* + Si vous accédez à l'application via http://192.168.1.102:8000 mais que les assets (CSS/JS) ne chargent pas, assurez-vous que :
    - Votre .env contient : APP\_URL=http://192.168.1.102:8000.
    - Recompilez les assets avec :

bash

Copier le code

npm run dev

#### 6. **Adresse IP dynamique**

Si votre routeur attribue des adresses IP dynamiques (qui changent à chaque redémarrage du réseau), vous devrez peut-être vérifier l'adresse IP de votre PC à chaque fois.