

Sommaire :

Présentation.....	2
Api.....	.2
Application mobile.....	.2
Installation.....	3
Base de données.....	.3
Utilisation.....	3
Api.....	.3
Application mobile.....	.5
Problèmes de fonctionnement.....	6

Présentation

Api

Cette API sert de liaison entre la base de données et l'application mobile, celle-ci permet le bon fonctionnement des requêtes effectuées sur l'application.

Elle permet des requêtes de type GET, POST, PUT, et DELETE.

On pourra récupérer un contact en particulier via son ID ou la totalité des contacts dans la base de donnée via des requêtes GET.

La méthode POST permet elle d'ajouter un nouveau contact à la base de donnée.

Une requête PUT sert à la modification d'un contact via son ID.

Enfin la méthode DELETE comme indiqué, est là pour pouvoir supprimé un contact de la base de donnée.

Application mobile

Le but de cette application, est un simple agenda de contacts, celle-ci vous permet de gérer votre liste de contact, d'afficher les détails d'un contact, d'en ajouter, de les modifier.

Au lancement, l'application va exécuter une méthode GET de tout les contacts pour vous les afficher en liste.

Il vous sera également permis d'ajouter un nouveau contact en rentrant les informations requises à l'écran.

Il vous est également possible d'afficher les détails d'un contact en appuyant sur celui souhaité, la possibilité de le supprimé ou de le modifier vous y sera présenté.

Installation

Base de données

Une fois les projets téléchargés, lancez WAMPServer et accédez à <http://localhost/phpmyadmin/> via un navigateur. Connectez-vous avec l'ID « root » sans mot de passe.

Créez ensuite une nouvelle base de données nommée « contact », cliquez sur cette dernière puis sur l'onglet « Importer ». Parcourez ensuite vos fichiers pour y trouver le fichier « contact.sql ».

La base de données est maintenant importée.

Utilisation

API

Pour pouvoir utiliser l'api, ouvrez la solution via visual studio puis lancez là, une page firefox devrait s'ouvrir (**NE PAS LA FERMER AVANT FIN D'UTILISATION /!**). Pour s'assurer du bon fonctionnement de l'api, rendez vous sur la page ouverte et rendez-vous sur ce le lien « <http://localhost:44320/api/contact> ». Une liste de contact devrait donc s'afficher.

Si cette liste apparait, vous pouvez donc tester les différentes fonctionnalités de l'API, pour cela procurez-vous « Postman », un logiciel de requête.

TOUT CA TOUJOURS EN AYANT L'APPLICATION D'API LANCE

Pour les requêtes GET :

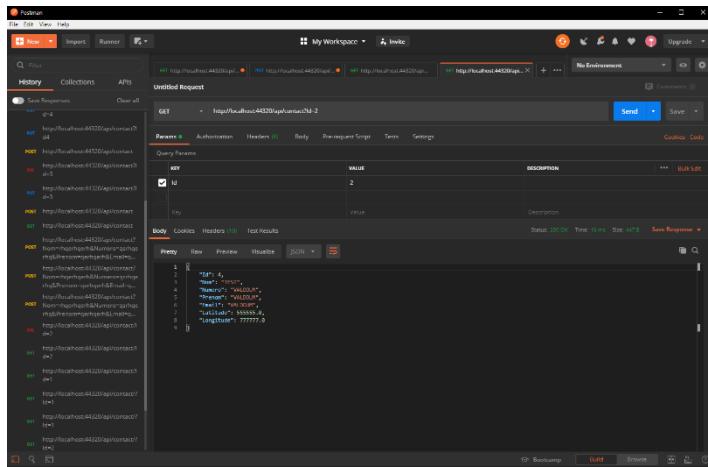
Si vous souhaitez afficher tous les contacts ->

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'History', 'Collections', and 'APIs'. Below it, a list of recent requests including 'Today' and 'April 21'. In the center, there's a 'My Workspace' section with a 'Collection' named 'Untitled Request'. This collection contains a single 'GET' request to 'http://localhost:44320/api/contact'. The 'Body' tab of the request shows a JSON object with fields like 'Name', 'Email', 'Phone', 'Address', 'Mobile', and 'LengthTime'. The 'Body' tab also includes a 'JSON Schema' definition. At the bottom, there's a preview of the response, which shows a list of contacts in JSON format. The response body is as follows:

```
[{"id": 1, "Name": "John Doe", "Email": "johndoe@example.com", "Phone": "123-4567", "Address": "123 Main St", "Mobile": "555-1234", "LengthTime": "99999"}, {"id": 2, "Name": "Jane Smith", "Email": "janesmith@example.com", "Phone": "234-5678", "Address": "456 Elm St", "Mobile": "555-2345", "LengthTime": "99999"}, {"id": 3, "Name": "Bob Johnson", "Email": "bobjohnson@example.com", "Phone": "345-6789", "Address": "789 Oak St", "Mobile": "555-3456", "LengthTime": "99999"}, {"id": 4, "Name": "Alice Williams", "Email": "alicewilliams@example.com", "Phone": "456-7890", "Address": "567 Pine St", "Mobile": "555-4567", "LengthTime": "99999"}, {"id": 5, "Name": "Mike Brown", "Email": "mikebrown@example.com", "Phone": "567-8901", "Address": "987 Cedar St", "Mobile": "555-5678", "LengthTime": "99999"}, {"id": 6, "Name": "Sarah Davis", "Email": "sarahdavis@example.com", "Phone": "678-9012", "Address": "321 Birch St", "Mobile": "555-6789", "LengthTime": "99999"}, {"id": 7, "Name": "David Wilson", "Email": "davidwilson@example.com", "Phone": "789-9123", "Address": "654 Elm St", "Mobile": "555-7890", "LengthTime": "99999"}, {"id": 8, "Name": "Emily Clark", "Email": "emilyclark@example.com", "Phone": "890-9123", "Address": "234 Cedar St", "Mobile": "555-8901", "LengthTime": "99999"}, {"id": 9, "Name": "Frank Miller", "Email": "frankmiller@example.com", "Phone": "987-9123", "Address": "456 Birch St", "Mobile": "555-9876", "LengthTime": "99999"}]
```

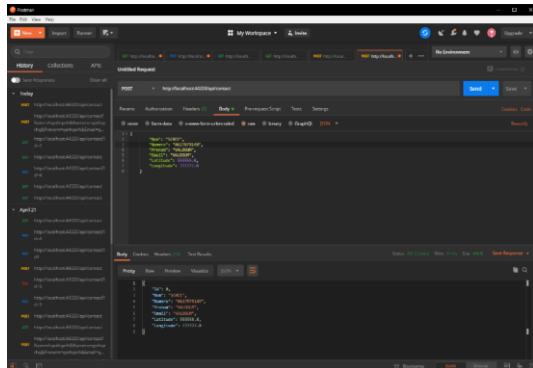
DOCUMENTATION APPLICATION MOBILE XAMARIN + API

Si vous souhaitez afficher un contact via son ID ->

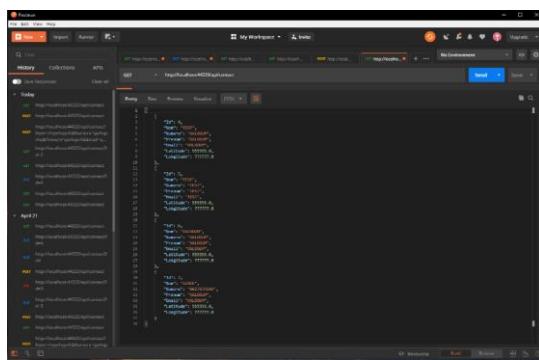


Pour les requêtes POST :

Vous devrez passer la requête en POST et vous rendre dans la section « body » pour vos paramètres, passez bien le traitement du texte en « raw ».



Effectuez ensuite une requête global GET, votre contact devrait apparaître.



DOCUMENTATION APPLICATION MOBILE XAMARIN + API

Pour les requêtes PUT :

Entrez l'ID du contact que vous souhaitez modifier et faire les mêmes manipulations que pour la méthode POST

The screenshot shows the Postman interface with an 'Untitled Request' tab. The method is set to 'PUT' and the URL is 'http://localhost:44320/api/contact/3'. The 'Body' tab is selected, showing a JSON payload:

```
1: {  
2:     "Name": "John Doe TEST",  
3:     "Email": "Johndoe@TEST.COM",  
4:     "Phone": "1234567890",  
5:     "Address": "123 Main St",  
6:     "Longitude": 777777.0  
7: }
```

Below the body, the 'Headers' tab shows 'Content-Type: application/json'. The status bar at the bottom indicates a successful response: 'Status: 200 OK Time: 0ms Size: 111.0'.

Vous devriez voir vos modifications après un appel GET

The screenshot shows the Postman interface with an 'Untitled Request' tab. The method is set to 'GET' and the URL is 'http://localhost:44320/api/contact'. A 'Params' section contains a key-value pair: 'id' with value '3'. The 'Body' tab shows the response body, which is identical to the PUT request above:

```
1: {  
2:     "Name": "John Doe TEST",  
3:     "Email": "Johndoe@TEST.COM",  
4:     "Phone": "1234567890",  
5:     "Address": "123 Main St",  
6:     "Longitude": 777777.0  
7: }
```

The status bar at the bottom indicates a successful response: 'Status: 200 OK Time: 0ms Size: 111.0'.

Enfin pour la requête DELETE :

The screenshot shows the Postman interface with an 'Untitled Request' tab. The method is set to 'DELETE' and the URL is 'http://localhost:44320/api/contact/3'. The 'Body' tab shows the response body, which is identical to the previous GET request:

```
1: {  
2:     "Name": "John Doe TEST",  
3:     "Email": "Johndoe@TEST.COM",  
4:     "Phone": "1234567890",  
5:     "Address": "123 Main St",  
6:     "Longitude": 777777.0  
7: }
```

The status bar at the bottom indicates a successful response: 'Status: 200 OK Time: 0ms Size: 111.0'.

Après une requête GET, votre contact devrait avoir disparu de la liste.

Application mobile

/!\ Toujours en développement suite a quelques soucis /!

Problèmes de fonctionnement

Suite à un souci de réponse de la méthode asynchrone, l'application mobile n'est donc pas fonctionnelle

