



React Native

Exploitation des API externes



La solution API

Les API sont essentielles en React Native car elles permettent à une application mobile de communiquer avec des services externes comme des bases de données, des serveurs ou des outils tiers.

Les API permettent à React-N entre autres, de :

- récupérer des données en temps réel
- envoyer des informations (formulaires, images, paiements...)
- gérer l'authentification des utilisateurs
- interagir avec des fonctionnalités avancées comme la géolocalisation ou la traduction.

En gros, les API transforment une application statique en une interface dynamique et réactive.



La solution : Stockage local

- Il est tout à fait possible d'utiliser des bases de données locales comme **SQLite** avec React Native, notamment pour stocker des données hors-ligne **ou conserver des informations persistantes sur l'appareil.**
- Cela convient bien aux applications simples ou aux fonctionnalités autonomes.
- Les API web offrent une plus grande flexibilité, car elles permettent de **synchroniser les données entre plusieurs appareils**, d'accéder à des services distants et de maintenir l'application à jour.

En résumé, SQLite est utile pour le local, mais les API sont souvent préférables pour des applications connectées, évolutives et collaboratives.



Méthodes d'accès aux API

Pour manipuler les API avec React-N, deux librairies sont disponibles :

- **L'API native de React-N : native Fetch**
- **Axios (priviligée)**

Axios est une bibliothèque largement utilisée en React Native pour effectuer des requêtes HTTP vers des API

Elle facilite la communication entre l'application mobile et les serveurs en proposant une **syntaxe claire et concise pour envoyer ou recevoir des données.**

Elle permet aussi de faciliter l'intégration des fonctionnalités dynamiques comme l'authentification, la mise à jour en temps réel ou la synchronisation avec des services web



Mise en œuvre de la bibliothèque axios

```
npm install axios
```



Exemple de code

```
import axios from 'axios';
import React, { useEffect } from 'react';
import { View } from 'react-native';

export default function Test() {

  const fetchApiData = async () => {
    const data = await axios.get('http://localhost:8000/api/resources')
      .then((response) => {
        console.log(response)
        return response.data;
      });

    console.log(data);
  }

  useEffect(() => {
    fetchApiData();
  }, [])

  return (
    <View>

    </View>
  );
};
```



Ajout des données dans la requête HTTP

```
export default function Test() {
  const fetchApiData = async () => {

    const jwtToken = 'TestToken';

    const config = {
      headers: {
        Accept: 'application/json',
        Authorization: 'Test ' + jwtToken,
      }
    }

    const data = await axios.get('http://localhost:8000/api/resources', config)
      .then((response) => {
        console.log(response)
        return response.data;
      });
  }

  useEffect(() => {
    fetchApiData();
  }, [])

  return (
    <View>
      ...
    </View>
  );
};
```



```
// GET Request
```

```
axios.get(url, config)
```

```
// POST Request
```

```
axios.post(url, data, config)
```

```
// PUT Request
```

```
axios.put(url, data, config)
```

```
// DELETE Request
```

```
axios.delete(url, config)
```

```
// PATCH Request
```

```
axios.patch(url, data, config)
```




Exemple de code - GET

```
import React, { useState, useEffect } from 'react';
import axios from 'axios';

function DataFetchingComponent() {
  const [data, setData] = useState([]);

  useEffect(() => {
    axios.get('https://jsonplaceholder.typicode.com/posts')
      .then(response => {
        setData(response.data);
      })
      .catch(error => {
        console.error('Error fetching data:', error);
      });
  }, []);

  return (
    <div>
      <h1>Posts</h1>
      <ul>
        {data.map(post => (
          <li key={post.id}>{post.title}</li>
        ))}
      </ul>
    </div>
  );
}
```



Exemple de code - POST

```
import React, { useState } from 'react';
import axios from 'axios';

function CreatePostComponent() {

  const [title, setTitle] = useState('');
  const [body, setBody] = useState('');

  const handleSubmit = (e) => {
    e.preventDefault();
    const post = { title, body };

    axios.post('https://jsonplaceholder.typicode.com/posts', post)
      .then(response => {
          console.log('Post created:', response.data);
});

    .catch(error => {
      console.error('Error creating post:', error);
    });
  };
};
```



Exemple de code - POST

```
return (
  <div>
    <h1>Create Post</h1>
    <form onSubmit={handleSubmit}>
      <div>
        <label>Title:</label>
        <input
          type="text"
          value={title}
          onChange={ (e) => setTitle(e.target.value) }
        />
      </div>
      <div>
        <label>Body:</label>
        <textarea
          value={body}
          onChange={ (e) => setBody(e.target.value) }
        />
      </div>
      <button type="submit">Submit</button>
    </form>
  </div>
);
}

export default CreatePostComponent;
```