

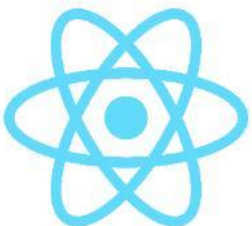
# CONSOMMATION DES APIS EN REACT

**COURS : INTRODUCTION A REACT JS**

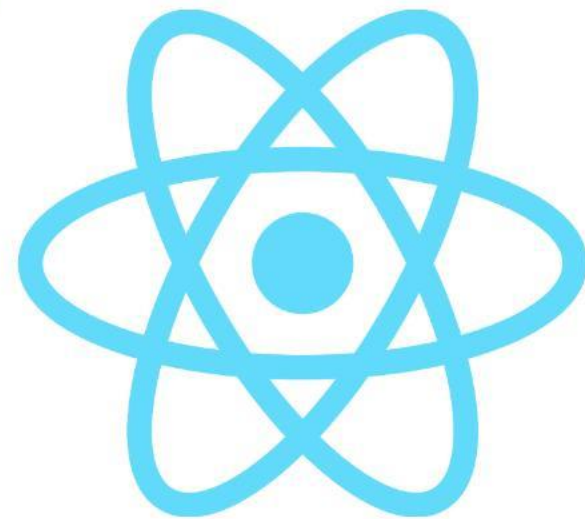
**Enseignant** : M. Florent BALKOULGA




université  
virtuelle  
Burkina Faso




# INTRODUCTION






De nombreux projets sur le web doivent s'interfacer avec une API REST à un certain stade de leur développement. Axios est un client HTTP léger basé sur le service `$http` dans Angular.js v1.x et est similaire à l'API native JavaScript Fetch.



Axios est basé sur Promise, ce qui vous permet de profiter des avantages d'async de JavaScript et await pour un code asynchrone plus lisible.

Vous pouvez également intercepter et annuler des demandes, et il existe une protection intégrée côté client contre la falsification des demandes intersites.



Dans ce cours, vous explorerez des exemples d'utilisation d'Axios pour accéder à la populaire API JSON Placeholder au sein d'une application React.

# Ajouter Axios au projet

Dans cette section, vous ajouterez Axios notre projet,

Pour ajouter Axios au projet, ouvrez votre terminal et placez-vous dans le répertoire dans votre projet.

Exécutez ensuite cette commande pour installer Axios :

```
npm install axios
```

Ensuite, vous devrez importer Axios dans le fichier dans lequel vous voulez l'utiliser.

# Faire une requête GET

Dans cet exemple, vous créez un nouveau composant et vous y importez Axios pour envoyer une demande GET.

Dans le dossier src de votre projet React, ajoutez le code suivant au composant App:

```
import React from 'react';

import axios from 'axios';

export default class PersonList extends React.Component {
  state = {
    persons: []
  }

  componentDidMount() {
    axios.get('https://jsonplaceholder.typicode.com/users')
      .then(res => {
        const persons = res.data;
        this.setState({ persons });
      })
  }

  render() {
    return (
      <ul>
        { this.state.persons.map(person =>
          <li>{person.name}</li>)}
      </ul>
    )
  }
}
```

# Faire une requête GET

Tout d'abord, vous importez React et Axios afin que les deux puissent être utilisés dans le composant. Ensuite, vous vous branchez sur le `componentDidMount` lifecycle hook et effectuez une demande GET.

Vous utilisez `axios.get(url)` avec une URL provenant d'un point final de l'API pour obtenir une promesse qui renvoie un objet de réponse. À l'intérieur de l'objet de réponse, il y a des données auxquelles on attribue la valeur `person`.

Vous pouvez également obtenir d'autres informations sur la demande, telles que le code de statut sous `res.status` ou plus d'informations à l'intérieur de `res.request`.

```
import React from 'react';

import axios from 'axios';

export default class PersonList
  extends React.Component {
  state = {
    persons: []
  }

  componentDidMount() {

    axios.get(`https://jsonplaceholder
r.typicode.com/users`)
      .then(res => {
        const persons = res.data;
        this.setState({ persons });
      })

    render() {
      return (
        <ul>
          {
            this.state.persons.map(person =>
              <li>{person.name}</li>
            )
          }
        </ul>
      )
    }
  }
}
```

# Faire une demande POST

Dans cette étape, vous utiliserez Axios avec une autre méthode de requête HTTP appelée POST.

Supprimer le code précédent dans App et ajoutez les éléments suivants pour créer un formulaire qui permet à l'utilisateur d'entrer des données et ensuite POSTer le contenu à une API :

```
import React from 'react';
import axios from 'axios';

export default class PersonList extends React.Component {
  state = {
    name: '',
  }

  handleChange = event => {
    this.setState({ name: event.target.value });
  }

  handleSubmit = event => {
    event.preventDefault();

    const user = {
      name: this.state.name
    };

    axios.post('https://jsonplaceholder.typicode.com/users', { user })
      .then(res => {
        console.log(res);
        console.log(res.data);
      })
  }

  render() {
    return (
      <div>
        <form onSubmit={this.handleSubmit}>
          <label>
            Person Name:
            <input type="text" name="name" onChange={this.handleChange} />
          </label>
          <button type="submit">Add</button>
        </form>
      </div>
    )
  }
}
```

# Faire une demande POST

Dans la fonction `handleSubmit`, vous empêchez l'action par défaut du formulaire. Ensuite, vous mettez à jour l'état en entrée de l'utilisateur.

Utiliser POST vous donne le même objet de réponse avec des informations que vous pouvez utiliser dans un appel `then`.

Pour compléter la demande POST, vous devez d'abord saisir l'entrée de l'utilisateur. Ensuite, vous ajoutez les données en même temps que la demande POST, qui vous donnera une réponse. Vous pouvez alors utiliser `console.log` pour la réponse, qui devrait montrer l'entrée `user` dans le formulaire.



# Faire une demande DELETE

Dans cet exemple, vous verrez comment supprimer des éléments d'une API en utilisant `axios.delete` et en passant une URL comme paramètre.

Changez le code du formulaire de l'exemple POST pour supprimer un utilisateur au lieu d'en ajouter un nouveau :

```
import React from 'react';
import axios from 'axios';

export default class PersonList extends React.Component {
  state = {
    id: '',
  }

  handleChange = event => {
    this.setState({ id: event.target.value });
  }

  handleSubmit = event => {
    event.preventDefault();

    axios.delete(`https://jsonplaceholder.typicode.com/users/${this.state.id}`)
      .then(res => {
        console.log(res);
        console.log(res.data);
      })
  }

  render() {
    return (
      <div>
        <form onSubmit={this.handleSubmit}>
          <label>
            Person ID:
            <input type="text" name="id" onChange={this.handleChange} />
          </label>
          <button type="submit">Delete</button>
        </form>
      </div>
    )
  }
}
```

# Utiliser une instance de base dans Axios

Dans cet exemple, vous verrez comment vous pouvez mettre en place une instance de base dans laquelle vous pouvez définir une URL et tout autre élément de configuration.

Créez un fichier séparé nommé api.js:

Exportez une nouvelle instance axios avec ces valeurs par défaut :

Une fois que l'instance par défaut est configurée, elle peut être utilisée dans le composant PersonList. Vous importez la nouvelle instance comme ceci :

```
import axios from 'axios';

export default axios.create({
  baseURL:
    `http://jsonplaceholder.typicode.com/`
});
```

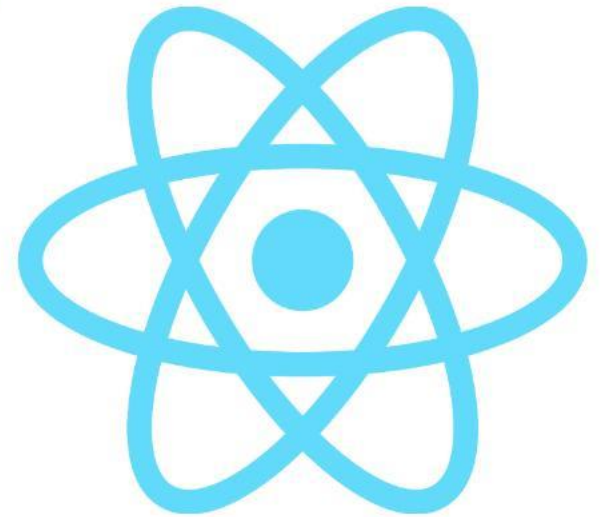
```
import React from 'react';

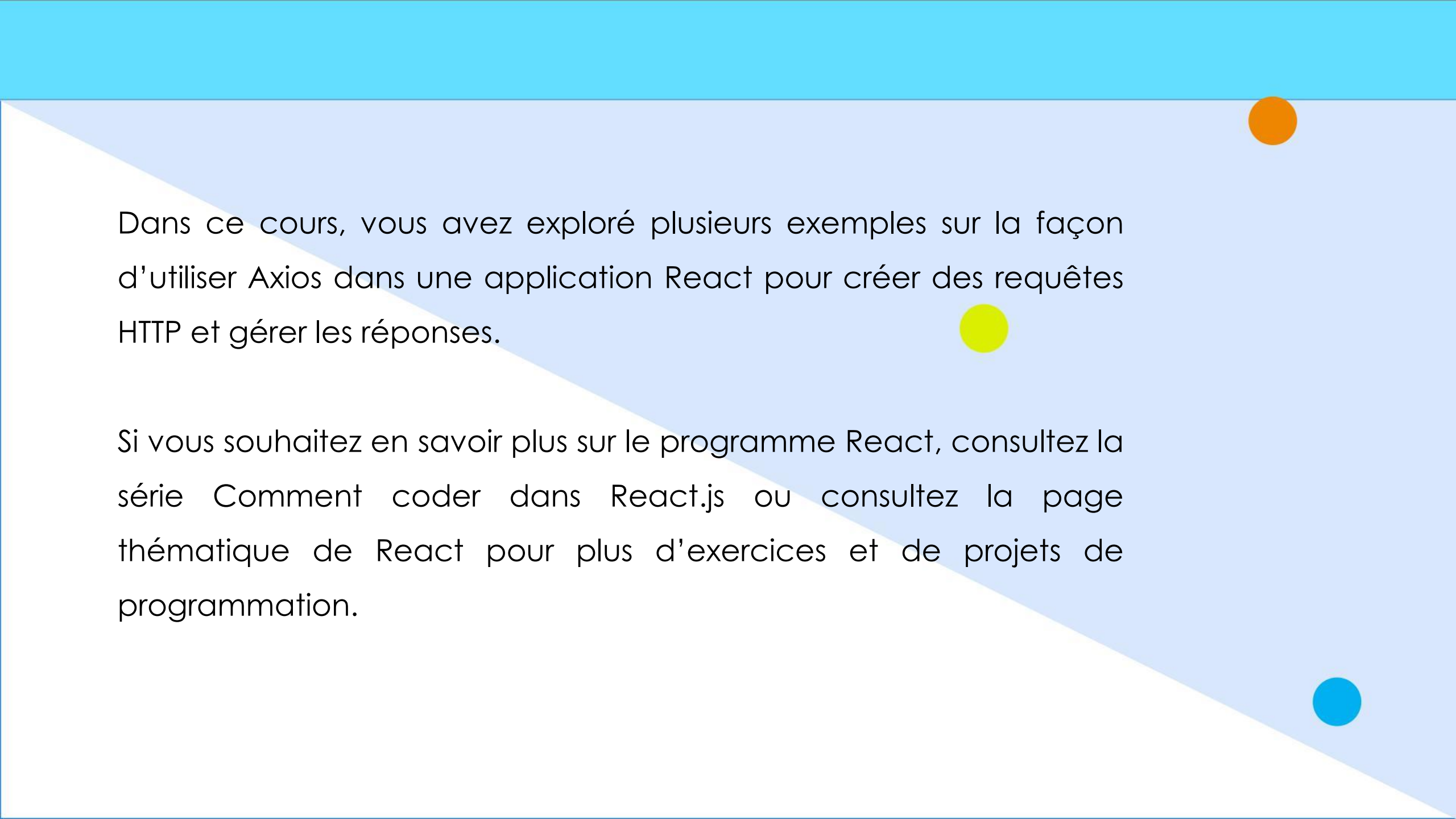
import API from '../api';

export default class PersonList
  extends React.Component {
  handleSubmit = event => {
    event.preventDefault();

    API.delete(`users/${this.state.id}`)
      .then(res => {
        console.log(res);
        console.log(res.data);
      })
  }
}
```

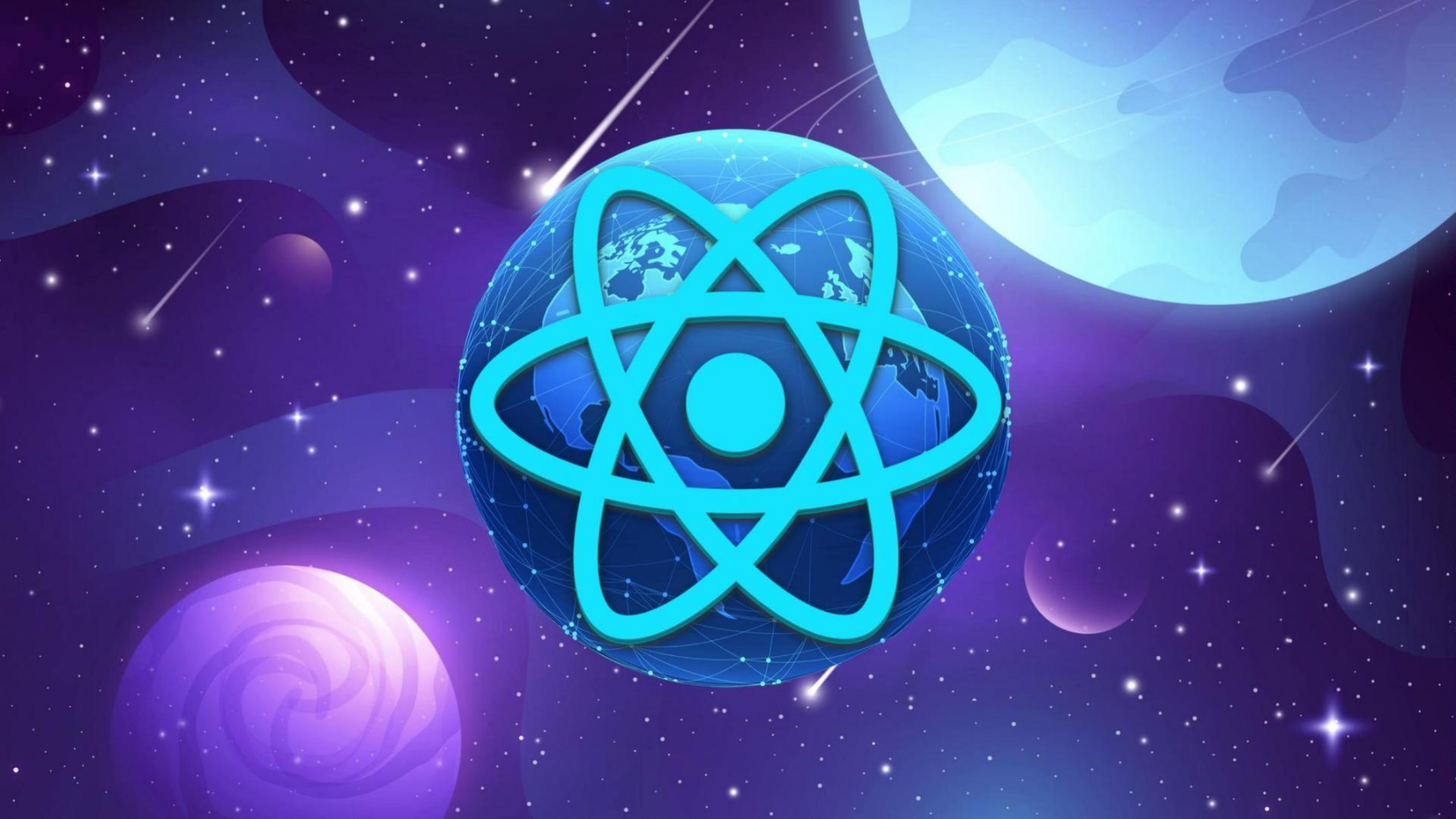
# CONCLUSION



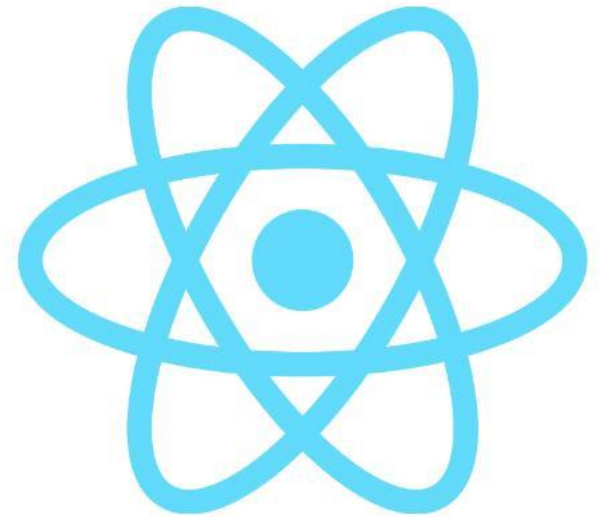


Dans ce cours, vous avez exploré plusieurs exemples sur la façon d'utiliser Axios dans une application React pour créer des requêtes HTTP et gérer les réponses.

Si vous souhaitez en savoir plus sur le programme React, consultez la série [Comment coder dans React.js](#) ou consultez la page thématique de React pour plus d'exercices et de projets de programmation.



# RESSOURCES





# Ressources

- ✓ <https://blog.arcoptimizer.com/consommer-les-api-rest-en-reaction-avec-fetch-et-axios>
  - ✓ <https://www.savoirdanslavie.com/how-to-consume-apis-in-react-using-fetch-and-axios/>
  - ✓ <https://fr.legacy.reactjs.org/docs/faq-ajax.html>
  - ✓ <https://techblog.ingeniance.fr/appeler-une-api-dans-un-composant-fonctionnel-react/>
  - ✓ <https://zephyrnet.com/fr/consommer-des-apis-de-repos-en-r%C3%A9action-avec-fetch-et-axios/>
  - ✓ [https://developer.mozilla.org/fr/docs/Learn/JavaScript/Client-side web APIs/Introduction](https://developer.mozilla.org/fr/docs/Learn/JavaScript/Client-side_web_APIs/Introduction)
  - ✓ [https://developer.mozilla.org/fr/docs/Learn/JavaScript/Client-side web APIs/Fetching data](https://developer.mozilla.org/fr/docs/Learn/JavaScript/Client-side_web_APIs/Fetching_data)
  - ✓ <https://www.synonyme-du-mot.com/les-articles/comment-utiliser-une-api-avec-react>
- 
- ✓ <https://www.youtube.com/watch?v=YaHTPRquCbU>
  - ✓ <https://www.youtube.com/watch?v=pgWcJygAk4s>
  - ✓ <https://www.youtube.com/watch?v=HUTF01zmcDY>
  - ✓ <https://www.youtube.com/watch?v=uVk3kHLBZ6c>



**“ A LA PROCHAINE POUR LE  
COURS SUIVANT ”**

