

Front formulaires





Ajout d'une annonce



Création de la page

Crée une nouvelle page **new.tsx** dans le dossier **ad** et consulte la via l'url **/ad/new**

The screenshot displays a web browser and a code editor side-by-side. The browser on the left shows the URL `http://localhost:3000/ad/new` and a placeholder page with the text "Here we will create a new ad". The code editor on the right shows the file structure of `new.tsx - frontend_boilerplate` with `new.tsx` selected under the `ad` directory. The code in `new.tsx` is as follows:

```
1 const NewAd = () => {  
2   return <p>Here we will create  
3 };  
4  
5 export default NewAd;  
6
```



Création du formulaire

Commençons par créer 1 input avec 1 label et 1 bouton

Titre de l'annonce:


```
src > pages > ad > TS new.tsx > ...
1  const NewAd = () => {
2    return (
3      <>
4        <label>
5          Titre de l'annonce: <br />
6          <input className="text-field" name="titre" />
7        </label>
8        <button className="button">Submit</button>
9      </>
10   );
11 };
12
13 export default NewAd;
14
```



Création du formulaire

Remplaçons le fragment (`<></>`) par une balise `<form></form>`

```
1  const NewAd = () => {  
2    return (  
3      <form>  
4        <label>  
5          Titre de l'annonce: <br />  
6          <input className="text-field" name="titre" />  
7        </label>  
8        <button className="button">Submit</button>  
9      </form>  
10   );  
11 };  
12  
13 export default NewAd;  
14
```

Que se passe-t-il au clic sur le bouton submit ?



Gestion de la soumission

Ajoute une prop onSubmit et empêche le fonctionnement de base du formulaire:

```
const NewAd = () => {  
  return (  
    <form  
      onSubmit={(e) => {  
        e.preventDefault();  
      }}  
    >  
  )  
}
```



Récupération de la donnée dans le js

Récupérons maintenant les valeurs entrées dans les différents inputs de notre formulaire:

```
src > pages > ad > TS new.tsx > [x] NewAd
1  const NewAd = () => {
2    return (
3      <form
4        onSubmit={(e) => {
5          // Prevent the browser from reloading the page
6          e.preventDefault();
7
8          // Read the form data
9          const form = e.target;
10         const formData = new FormData(form as HTMLFormElement);
11
12         // Or you can work with it as a plain object:
13         const formJson = Object.fromEntries(formData.entries());
14         console.log(formJson);
15       }}
16    >
```

[source : https://react.dev/reference/react-dom/components/input#reading-the-input-values-when-submitting-a-form](https://react.dev/reference/react-dom/components/input#reading-the-input-values-when-submitting-a-form)



Récupération de la donnée dans le js

Ajoute tous les champs nécessaires à la création d'une annonce, sauf la catégorie, et vérifie que tu récupères bien la valeur dans la console:

The screenshot displays a web application interface on the left and a browser's developer console on the right. The web application has a header with 'TGC' and 'Pub' labels, and a search icon. Below the header, there are category links: 'Ameublement', 'Électroménager', and 'Photo'. The main form area contains two input fields: 'Titre de l'annonce:' with the value 'Je vends ma 308' and 'Prix:' with the value '35000'. A 'Submit' button is located below the price field. The browser's developer console on the right shows the 'Console' tab selected, displaying a log entry: `{titre: 'Je vends ma 308', price: '35000'}`.



Sélection de la catégorie

Les catégories étant être créés côté, on doit d'abord fetcher les catégories pour les rendre accessible dans notre formulaire.

Commence par récupérer les catégories au chargement du composant et affiche-les dans la console

```
TS new.tsx U X
src > pages > ad > TS new.tsx > [⌕] NewAd > <function>
1  import axios from "axios";
2
3  type category = {
4    id: number;
5    name: string;
6  };
7
8  const NewAd = () => {
9    const fetchCategories = async () => {
10      const result = await axios.get<category[]>(  
11        "http://localhost:4000/category"  
12      );  
13      console.log(result);  
14    };  
15    fetchCategories();  
16    return (  
17      <form  
18      ⚡ onsubmit={e => {
```



Rendre l'affichage réactif

Une fois les catégories récupérées, on veut que l'interface se mette à jour et propose de sélectionner la catégorie de l'annonce. On va donc enregistrer ces catégories dans un state.

```
TS new.tsx U X
src > pages > ad > TS new.tsx > [🔍] NewAd > <function>
1  import axios from "axios";
2  import { useState, useEffect } from "react";
3
4  type category = {
5    id: number;
6    name: string;
7  };
8
9  const NewAd = () => {
10    const [categories, setCategories] = useState<category[]>([]);
11    useEffect(() => {
12      const fetchCategories = async () => {
13        const result = await axios.get<category[]>("http://localhost:4000/category");
14      };
15      setCategories(result.data);
16    };
17    fetchCategories();
18  }, []);
19  return (
20    <form
21      onSubmit={(e) => {
```



Map sur les catégories

Nous allons maintenant afficher les catégories dans un `<select>` et mapper sur le tableau pour ajouter 1 option par catégorie:

```
42   <label>
43     Prix: <br />
44     <input type="number" className="text-field" name="price" />
45   </label>
46   <br />
47   <select name="category">
48     {categories.map((el) => (
49       <option value={el.id} key={el.id}>
50         {el.name}
51       </option>
52     ))}
53   </select>
54   <button className="button">Submit</button>
55 </form>
```



Résultat

Nous pouvons maintenant choisir la catégorie dans un select et toutes les données du formulaire sont au format json, prêtes à être envoyés à notre backend

TGC

Publier

Ameublement • Électroménager • Photographie • Informatique • Téléphonie • Vélos

Titre de l'annonce:

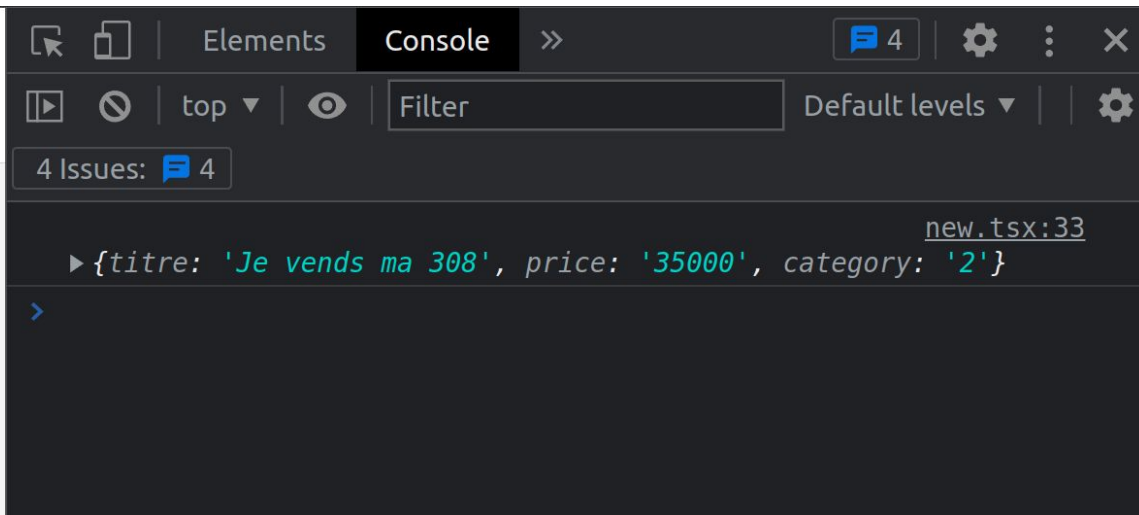
Je vends ma 308

Prix:

35000

Vehicles ▾

Submit





A toi de jouer !

- Réalise toutes ces étapes et crée la page et le formulaire permettant d'ajouter des annonces depuis le front
- **Bonus:** utilise <https://react-hook-form.com/> pour la gestion de ton formulaire et ajoute une validation côté front au clic sur le bouton submit