



*Préparation du DC
en « Développement Framework Côté Client »*

Niveau : L2DSI

AU : 2024 - 2025

Enoncé :

On propose de développer une application de gestion des réclamations des clients pour un hôtel. Chaque client, se connecte sur le serveur local de l'hôtel et saisit sa demande dans la page de réclamation.

En utilisant Vue.js 3, on se propose de développer cette page et en particulier, à créer un composant qui permet de récupérer les données saisies par le client.

Chaque réclamation est caractérisée par :

- *Nom du client*
- *Numéro de sa chambre*
- *Catégorie de la réclamation*
- *Description*
- *Priorité*

- 1. Donner le code du composant qui permet à l'utilisateur de saisir son nom, le numéro de sa chambre et la description de sa réclamation.*
- 2. On propose maintenant de permettre la saisie de la catégorie de la réclamation. Ceci sera achevé à partir d'une liste qu'on affiche à l'utilisateur. Donner le code pour afficher cette liste.*

On donne le tableau suivant :

Tableau contenant les catégories des réclamations :

```
const categorieReclamation = [  
  { 'id': '1', 'titre' : 'Nettoyage' },  
  { 'id': '2', 'titre' : 'Réparation' },  
  { 'id': '3', 'titre' : 'Autres' }  
]
```

3. Si on suppose que cette liste provient d'une API dont l'URL est la suivante :
« <https://www.monhotel.tn/CategorieReclamation> », redonner le code pour récupérer cette liste à partir de cette URL au lieu du tableau.
4. On s'intéresse maintenant à la saisie de la priorité. Sachant qu'elle peut avoir trois valeurs possibles (Élevée, Normale et Basse), donner le code permettant au client de préciser la priorité à partir des boutons radios.
5. Dès la fin de la saisie de sa réclamation, on affiche en bas du formulaire un bouton nommé "Valider". Dès l'appui sur ce bouton, on affiche :
- Le message "Voici le résumé de votre réclamation : "
 - La liste des données qu'il vient de saisir. Notons que si la priorité est haute, la catégorie s'écrit en rouge et si la priorité est basse, la catégorie s'écrit en gris clair #999.
 - Un bouton nommé "Envoyer" qui fait appel à la méthode "envoyer()"

Donner le code nécessaire.

6. On s'intéresse maintenant à écrire le code de la méthode "envoyer()". Cette dernière a pour rôle d'envoyer toutes les données à notre API via l'URL donnée précédemment avec la méthode POST.

Donner le code nécessaire.

On donne :

Exemple de code récupération des données d'un serveur :

```
fetch('https://www.monhotel.tn/mesdonnees/')
  .then(response => response.json())
  .then(data => {
    console.log(data)
  })
```

Exemple de code d'envoi des données au serveur :

```
const url = "https://www.monhotel.tn/mesdonnees/";
const options = {
  method: "POST",
  headers: {
    Accept: "application/json",
    "Content-Type": "application/json;charset=UTF-8",
  },
  body: JSON.stringify({
    parametre1: 100,
```

```
        parametre2: 'val2',
      }),
    };
    fetch(url, options)
      .then((response) => response.json())
      .then((data) => {
        console.log(data);
      });
```

Bon travail...