C'est parti. En tant qu'expert technique du projet, voici la documentation de référence complète pour la configuration du fichier .json.

Cette notice est destinée aux administrateurs ou développeurs souhaitant créer ou modifier la structure de l'application **AdminForm**.

📚 Documentation Technique : Configuration JSON

Le fichier de configuration (ex: data.json) pilote l'intégralité de l'interface : la grille, les types d'interactions, la navigation et les données.

1. Structure Globale

Le fichier JSON doit respecter strictement la structure suivante : un objet racine contenant deux tableaux obligatoires.

JSON

```
{
  "columns": [ ... ],  // Définition des métadonnées (Structure)
  "rows": [ ... ]      // Les données brutes (Contenu)
}
```

- **columns** : Définit l'en-tête, le type de champ (texte, liste, IA...) et les comportements.
- **rows** : Contient les valeurs. Chaque ligne doit correspondre exactement à l'ordre des colonnes.

---

2. Définition des Colonnes (columns)

C'est un tableau d'objets. Chaque objet représente une colonne.

Propriétés Communes

Chaque colonne **doit** avoir :

- id (String) : Identifiant unique (utilisé par l'IA pour cibler les données).
- label (String) : Le titre affiché en haut du tableau.
- type (String) : Le comportement de la colonne (voir liste ci-dessous).
- params (Object, *optionnel*) : Paramètres spécifiques selon le type.

📋 Liste des Types et Paramètres

Voici le détail technique pour chaque type de colonne supporté.

Type : question

- **Fonction :** Affiche un texte en lecture seule (gras). Utilisé pour les intitulés ou instructions.
- **Paramètres :** Aucun.
- **Exemple :**
  JSON
  `{ "id": "task", "label": "Tâche à faire", "type": "question" }`

Type : reponse

- **Fonction :** Affiche une zone de texte (textarea) modifiable par l'utilisateur.
- **Paramètres :** Aucun.
- **Exemple :**
  JSON
  `{ "id": "user_comment", "label": "Vos observations", "type": "reponse" }`

Type : chapitre

- **Fonction :** Affiche une valeur textuelle ET génère le menu de navigation principal (Niveau 1) dans la barre latérale.
- **Paramètres :** Aucun.
- **Exemple :**
  JSON
  `{ "id": "phase", "label": "Phase Projet", "type": "chapitre" }`

Type : sous-chapitre

- **Fonction :** Similaire à chapitre, mais génère le sous-menu (Niveau 2) pour filtrer plus finement.
- **Paramètres :** Aucun.
- **Exemple :**
  JSON
  `{ "id": "step", "label": "Étape", "type": "sous-chapitre" }`

Type : combo (Liste déroulante)

- **Fonction :** Affiche un menu sélecteur. Change la couleur de fond de la cellule selon la valeur choisie.
- **Paramètres (params) :**
  - options (Array) : Liste des choix textuels possibles.
  - colors (Object) : Dictionnaire associant "Valeur": "CodeCouleurCSS".
- **Exemple :**
  JSON

```
{
 "id": "status",
 "label": "État",
 "type": "combo",
 "params": {
  "options": ["OK", "KO", "En cours"],
  "colors": {
   "OK": "#dcfce7",    // Vert clair
   "KO": "#fee2e2",    // Rouge clair
   "En cours": "#fef9c3" // Jaune clair
  }
 }
}
```

Type : qcm (Liste de contrôle)

- **Fonction :** Affiche une liste de cases à cocher (Checkboxes) interactives.
- **Paramètres :** Aucun dans columns. La structure des cases est définie dans les données (rows).
- **Exemple :**
  JSON

```
{ "id": "checks", "label": "Vérifications", "type": "qcm" }
```

Type : ia (Intelligence Artificielle)

- **Fonction :** Affiche un bouton pour générer du contenu via un LLM.
- **Paramètres (params) :**
  - requete (String) : L'instruction (prompt) envoyée à l'IA.
  - cibles (Array of Strings) : Liste des id des autres colonnes dont le contenu sera envoyé à l'IA comme contexte.
- **Exemple :**
  JSON

```
{
  "id": "ai_help",
  "label": "Assistant",
  "type": "ia",
  "params": {
    "requete": "Analyse la tâche et le statut, puis propose une solution.",
    "cibles": ["task", "status"] // L'IA lira le contenu des colonnes 'task' et 'status' de la ligne active
  }
}
```

---

3. Format des Données (rows)

La section rows est un tableau de tableaux (Matrice).

- L'ordre des valeurs dans une ligne suit strictement l'ordre de déclaration des columns.
- Indice 0 de rows = Indice 0 de columns.

Formats de valeurs par type

Type de Colonne	**Format attendu dans rows**	**Exemple de valeur**
question	String	"Installer le serveur"
reponse	String	"Installation réussie." (ou "" si vide)
chapitre	String	"Phase 1"
combo	String	"OK" (Doit correspondre à une des options)
ia	String	"" (Vide au départ, rempli par l'IA plus tard)
qcm	**Array of Objects**	*Voir détail ci-dessous*

⚠️ Le cas spécifique du QCM

Contrairement aux autres qui stockent du texte, le QCM stocke une structure complexe JSON directement dans la cellule.

Format d'un objet QCM :

JSON

```
[
 { "label": "Texte de la case 1", "checked": true },  // true = coché/barré
 { "label": "Texte de la case 2", "checked": false }  // false = vide
]
```

---

4. Exemple Complet (Template)

Voici un fichier .json complet et fonctionnel résumant cette documentation.

JSON

```
{
 "columns": [
  {
   "id": "chap",
   "label": "Module",
   "type": "chapitre"
  },
  {
   "id": "tache",
   "label": "Action requise",
   "type": "question"
  },
  {
   "id": "checklist",
   "label": "Contrôles",
   "type": "qcm"
  },
```

```
    {
      "id": "prio",
      "label": "Priorité",
      "type": "combo",
      "params": {
        "options": ["Basse", "Haute"],
        "colors": { "Haute": "#ffcccc" }
      }
    },
    {
      "id": "avis_ia",
      "label": "Conseil IA",
      "type": "ia",
      "params": {
        "requete": "Donne un risque potentiel pour cette tâche",
        "cibles": ["tache", "checklist"]
      }
    }
  ],
  "rows": [
    [
      "Serveur Web",                          // Colonne 1: Chapitre
      "Mise à jour Apache",                   // Colonne 2: Question
      [                                       // Colonne 3: QCM (Tableau)
        { "label": "Backup config", "checked": true },
        { "label": "Stop service", "checked": false }
      ],
      "Haute",                                // Colonne 4: Combo
      ""                                      // Colonne 5: IA (Vide)
    ]
  ]
}
```