

## Zadanie rekrutacyjne

Projektem do wykonania jest katalog produktów firmy odzieżowej.

Użytkownik obsługujący katalog powinien móc przeglądać dostępne drzewo kategorii, dodawać nowe, edytować oraz usuwać istniejące.

W oparciu o dostarczone źródło danych (REST API), należy zbudować aplikację kliencką, która umożliwi wykonywanie operacji **CRUD**. Aplikacja powinna być wykonana w formie dashboard-u, aby w łatwy sposób zarządzać danymi.

### Opis akcji w REST API:

- Tworzenie kategorii  
POST {adresApi}/categories
- Edycja kategorii  
PUT {adresApi}/categories/{categoryId}
- Usunięcie kategorii  
DELETE {adresApi}/categories/{categoryId}
- Pobranie listy kategorii  
GET {adresApi}/categories

Adres API: [http://dotnet.demos.i-sklep.pl/rest\\_api/shop\\_api/v1](http://dotnet.demos.i-sklep.pl/rest_api/shop_api/v1)

Login: rest

Hasło: gBict?3J

Autoryzacja: Basic Auth

Przykład ciała zapytania HTTP dla akcji PUT oraz POST:

```

{
  "category":
  {
    "parent_id": 1,
    "is_visible": true,
    "name": "Test Category",
    "description": "",
    "picture_filename": "",
    "ordering": 1
  }
}

```

Przykład odpowiedzi serwera:

```

{
  "version": "v1",
  "success": true,
  "data": {
    "category": {
      "id": 74,
      "parent_id": 1,
      "is_visible": true,
      "name": "Test Category",
      "description": "",
      "picture_filename": "",
      "ordering": 1,
      "source_id": null,
      "symbol": null
    }
  },
  "error": null
}

```

W polu „data” otrzymujemy dane zwrotne z serwera, które dodatkowo są pakowane w „category” dla pojedynczej kategorii oraz „categories” dla tablicy kategorii (wykorzystywane w metodzie GET).

## Model kategorii

W powyższych przykładach widzimy model danych wysyłany i zwracany z serwera w formacie JSON, który powinien być mapowany w aplikacji klienckiej.

Najbardziej znaczące pola w klasie Category:

- int Id
- int ParentId
- bool IsVisible
- string Name
- string Description
- int Ordering

## Podsumowanie

Wybór dotyczący dodatkowym bibliotek czy frameworków jest dowolny. Ważne jest, żeby aplikacja była oparta na jednym z frameworków JS – Angular/React/Vue.js/StencilJS

Kod źródłowy jest miejscem, w którym można się wykazać znajomością dobrych praktyk programistycznych i wzorców projektowych, także **jakość kodu** jest ważnym punktem tego zadania :)

Jeśli chodzi o wygląd aplikacji, to wystarczy prosty i przejrzysty layout, ale tutaj też wybór zostawiamy Tobie. Dodatkowy framework CSS na pewno pójdzie tylko na plus :)