

Day 3 – API Integration Report – Tech Store PK

The API integration for Tech Store PK used Template 4 API, aligning it with Template 8 schema. Adjustments ensured accurate fields like name, price, and category with validation rules. A custom (import-data.mjs) script streamlined data migration to Sanity CMS, securing sensitive data with (.env.local) and modularizing functions. Postman testing validated schema alignment and data accuracy. All steps, including API understanding, schema validation, and submission preparation, were documented and successfully completed for a scalable backend.

Steps for Day 3:

API Integration Process:

- Template API Used: Template 4.
- Template Number: 8.

Screenshots of:

- API calls.
- Data successfully displayed in the frontend.
- Sanity schema fields.
- Images are attached in documentation in Github Repo.

Adjustments Made to Schema:

The schema was adjusted to align with the API data structure. Below is the updated schema.

Schema:

```
export default {  
  name: 'product',  
  type: 'document',  
  title: 'Product',  
  fields: [  
    {  
      name: 'name',  
      type: 'string',  
      title: 'Name',  
      validation: (Rule: any) => Rule.required().error('Name is required'),  
    },  
    {  
      name: 'slug',  
      type: 'slug',  
      title: 'Slug',  
      validation: (Rule: any) => Rule.required().error('Slug is required'),  
      options: {  
        source: 'name',  
        maxLength: 200,  
      },  
    },  
    {  
      name: 'image',  
      type: 'image',  
      title: 'Image',  
      options: {
```

```
    hotspot: true,  
  },  
  description: 'Upload an image of the product.',  
},  
{  
  name: 'price',  
  type: 'string',  
  title: 'Price',  
  validation: (Rule: any) => Rule.required().error('Price is required'),  
},  
{  
  name: 'description',  
  type: 'text',  
  title: 'Description',  
  validation: (Rule: any) =>  
    Rule.max(150).warning('Keep the description under 150 characters.'),  
},  
{  
  name: 'discountPercentage',  
  type: 'number',  
  title: 'Discount Percentage',  
  validation: (Rule: any) =>  
    Rule.min(0).max(100).warning('Discount must be between 0 and 100.'),  
},  
{  
  name: 'isFeaturedProduct',  
  type: 'boolean',
```

```
    title: 'Is Featured Product',
  },
  {
    name: 'stockLevel',
    type: 'number',
    title: 'Stock Level',
    validation: (Rule: any) => Rule.min(0).error('Stock level must be a positive number.'),
  },
  {
    name: 'category',
    type: 'string',
    title: 'Category',
    options: {
      list: [
        { title: 'Chair', value: 'Chair' },
        { title: 'Sofa', value: 'Sofa' },
      ],
    },
    validation: (Rule: any) => Rule.required().error('Category is required'),
  },
],
};
```

Migration Steps and Tools:

1. Created a (script/import-data.mjs) file tailored to the environment for seamless data migration.
2. Adjusted the migration script to align with the API and schema requirements.

3. Used .env.local to protect API keys and sensitive data.
4. Ensured descriptive variable names and modularized functions for reusability.
5. Validated the migration into Sanity CMS.
6. Documented complex logic with detailed comments.

Testing Process:

- Used Postman to test the API before implementation.
- Verified data accuracy and alignment with the schema post-migration.

Self-Validation Checklist:

- **API Understanding:** ✓.
- **Schema Validation:** ✓
- **Data Migration:** ✓
- **API Integration in Next.js:** ✓
- **Submission Preparation:** ✓

The API integration and data migration processes for Tech Store PK were successfully completed. All steps were validated and documented thoroughly.