Guidance

Concepts

Resources

How Resources work

The `resources.json` file defines the content for resource cards, which include various learning materials like articles, events, and courses.

Example structure:

```
{
  "id": 10,
  "title": "Managing Change in Organisations",
  "description": "A seminar discussing change management strategies for
business success.",
  "categories": ["Change", "Leadership"],
  "thumbnail": "https://example.com/thumbnail.jpg",
  "alt": "Thumbnail showing a business leader adapting to change",
  "duration": 60,
  "type": "Event",
  "events": [
    {
      "location": "Liverpool Regional Centre",
      "start_date": "14/03/2025",
      "end_date": "14/03/2025",
      "href": "https://example.com/"
    }
  ]
}
```

Each resource is displayed as a card in the UI. The `type` determines the icon used, and the `events` field allows event-based resources to list occurrences with locations and dates. Clicking a resource navigates to its details page using `href`.

Components

Card Section

How to Use the CardSection Component

The `CardSection` component is a flexible and user-friendly way to display a collection of cards, such as resources or events. It supports filtering, pagination, and an optional carousel layout. By default and without filters applied, the card section is designed to show all resource cards.

Features

- **Filtering:** Supports multiple filters such as category, type, duration, location, and search query.
- **Pagination:** Allows users to navigate through multiple pages of content when enabled.
- **Carousel Mode:** Displays cards in a horizontally scrolling layout instead of a grid.
- **Responsive Layout:** Automatically adjusts card layout based on screen size.
- **Dynamic Content Loading:** Updates dynamically when new filters are applied.
- **Custom Callbacks:** Supports custom functions for handling user interactions like pagination and "View All" actions.

Props Explained

Prop	Type	Description
`id`	`string`	Unique identifier for the section.
`title`	`string`	Section title displayed at the top.
`description`	`string`	Optional description below the title.
`cards`	`array`	Array of objects representing the cards (courses, events, etc.).
`filters`	`object`	Filtering rules to refine the displayed cards.
`paginated`	`boolean`	Enables pagination when `true`.
`useCarousel`	`boolean`	Uses a carousel instead of a grid layout when `true`.
`onViewAll`	`function`	Callback for handling "View All" button actions.

Filtering

The component allows filtering based on various criteria:

- `byId`: Show specific courses by ID.
- `byCategory`: Filter courses by category.
- `byMaxDuration` / `byMinDuration`: Show courses within a duration range.
- `byType`: Filter by course type.
- `byLocation`: Show only courses available in a specific location.
- `byQuery`: Search courses by title or description.

Example:

```
filters={{ byCategory: ["Business"], byMaxDuration: 120 }}
```

Using Carousel Mode

Set `useCarousel` to `true` to enable a horizontal scrolling layout instead of a grid.

```
<CardSection useCarousel={true} />
```

With these features, `CardSection` provides a customizable and interactive way to showcase resources.

Carousel

How to Use the Carousel Component

The `Carousel` component is a dynamic and interactive way to display a rotating set of slides. It loads slides from an external JSON file and provides navigation controls.

Features

- **Automatic Slide Rotation:** The carousel automatically cycles through slides.
- **Manual Navigation:** Users can navigate using left/right arrows.
- **Keyboard Navigation:** Supports arrow key controls.
- **Focus Management:** Ensures accessible navigation by adjusting `tabindex` dynamically.
- **Lazy Loading:** Loads images efficiently to improve performance.
- **Dynamic Content:** Automatically updates when `slides.json` changes.

How the Carousel Works

The `Carousel` component fetches slide data from `slides.json`, which is structured as an array of slide objects. Each slide contains information like headings, descriptions, images, and a call-to-action button.

Example slide data:

```
[
  {
    "heading": "Change Hub",
    "body": "Change is a constant in leadership.",
    "image": "https://example.com/image1.jpg",
    "alt": "Leadership concept",
    "cta": { "label": "Read more", "href": "discover.html?id=1" }
  }
]
```

Each slide is displayed as a separate element in the carousel. Users can navigate through them using arrow buttons or keyboard controls.

**Understanding **`href`

The `href` property in slides and other links defines the destination URL when a user clicks a button or link. For example, in `slides.json`, the `cta` object contains `href`, which determines where the "Read more" button leads.

Example:

```
{
  "label": "Read more",
  "href": "discover.html?id=1"
}
```

This means clicking the button will navigate to `discover.html?id=1`.

Pages

Articles

How Article Pages work

The `articles.json` file determines the routes for displaying article pages, which are converted from Markdown to HTML.

Example structure:

```
[
  {
    "id": 1,
    "title": "9 Leadership Capabilities",
    "image": "https://example.com/image2.jpg",
    "page": "pages/9-capabilities.html"
  }
]
```

Each entry corresponds to a specific article. The `page` property contains the path to the HTML page generated from a Markdown file. This allows dynamic linking to detailed content pages.

Basic Markdown Rules

Markdown is a lightweight markup language used to format text.

- **Headings:** `#` for H1, `##` for H2, etc.
- **Bold:** `**bold text**`
- **Italic:** `*italic text*`
- **Links:** `[text](URL)`
- **Lists:** `- Item` for unordered, `1. Item` for ordered lists.
- **Images:** `![alt text](image-url)`

Example Markdown content:

```
# Leadership Skills

Learn how to be an effective leader.
```

```
## Key Points

- Adaptability
- Communication
- Decision Making

[Read More](pages/leadership.html)
```

This is converted to structured HTML for display on the website.

Discovery Page

How the Discovery Page Works

The `discover.json` file defines the content for sections on the Discovery Page, including banners, descriptions, and card sections. Each section is visually represented as a banner with supporting text and grouped cards.

Example structure:

```
[
    {
      "id": 1,
      "title": "Change Hub",
      "image": "https://images.pexels.com/photos/2566581/pexels-photo-
2566581.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=1",
      "description": "Change is a constant in leadership, and navigating it
effectively is essential for success. The Change Hub is your go-to resource
for building the skills needed to lead yourself, your team, and your
projects through transformation.",
      "cardSections": [
        {
            "title": "Leading Others Through Change",
            "description": "Effective leaders inspire and support their
teams through uncertainty.",
            "filters": {
              "byCategory": ["Change"],
              "topN": 4,
              "byId": [3]
            },
            "useCarousel": true,
            "onViewAll": "search.html?category=Change"
        },
        {
            "title": "Leading Myself Through Change",
            "description": "Effective leaders inspire and support their
teams through uncertainty.",
            "filters": {
              "byCategory": ["Change"],
              "topN": 4
```

```
            },
            "useCarousel": true,
            "onViewAll": "search.html?category=Change"
        },
        {
            "title": "Managing Change in Projects",
            "description": "Effective leaders inspire and support their
teams through uncertainty.",
            "filters": {
              "byCategory": ["Change"],
              "topN": 4
            },
            "useCarousel": true,
            "onViewAll": "search.html?category=Change"
        }
      ]
    }
  ]
```

Each section is displayed as a banner with an image, title, and description, followed by one or more card sections. Card sections group related content such as resources or articles.

Search Page

How the Search Page Works

The Search Page allows users to explore and filter content dynamically. It provides an intuitive way to search for resources, courses, articles, or events using various filters. The content displayed on this page is driven by the `resources.json` file.

Features

- **Search Bar**: Users can input keywords to find relevant content.
- **Filters**:
  - **Categories**: Filter content by categories (e.g., "Leadership", "Change").
  - **Types**: Filter content by type (e.g., "Article", "Event").
  - **Locations**: Filter event-based content by location (e.g., "London", "Liverpool").
- **Dynamic Updates**: The page dynamically updates results as filters or search terms are applied, without requiring a page reload.
- **Pagination**: Users can navigate through multiple pages of results.
- **Responsive Design**: The layout adjusts seamlessly for different screen sizes.

Categories, Types, and Locations

The **categories**, **types**, and **locations** shown on the Search Page are dynamically derived from the `resource.categories`, `resource.types`, and `resource.events.location` fields in the `resources.json` file. This means:

1. Any updates to the categories, types, or locations in `resources.json` are automatically reflected on the Search Page.
2. These filters are generated as unique sets from the data.

Categories

Categories are taken from the `categories` field of each resource. For example:

```
"categories": ["Leadership", "Change"]
```

This will display both "Leadership" and "Change" as filter options.

Types

Types are taken from the `type` field of each resource. For example:

```
"type": "Event"
```

This will display "Event" as a type filter.

Locations

Locations are derived from the `location` field within the `events` array of event-based resources. For example:

```
"events": [
    {
        "location": "Liverpool Regional Centre",
        "start_date": "14/03/2025",
        "end_date": "14/03/2025"
    }
]
```

This will display "London" as a location filter.

Dynamic Filtering

Filters allow users to refine their search results instantly. For example:

- Selecting a category like *"Leadership"* will show only resources tagged with that category.
- Choosing a type like *"Article"* will limit results to articles.
- Selecting a location like *"London"* will display only events happening in London.

Filters can be combined for more specific searches, and the results update dynamically as filters are applied or removed.

How It Appears on the Page

The Search Page includes:

1. A **search bar** at the top for keyword-based searches.
2. **Filter sections** for categories, types, and locations:
    - Each section can be expanded or collapsed for better navigation.
3. A **results area** displaying filtered content as cards