Scientific Citation App 项目说明文档

一、项目概述

Scientific Citation App 是一个科学引文查询应用，它允许用户输入 DOI（数字对象标识符）来获取相关的科学文献信息。这个应用使用了 React 作为前端框架, Material-UI 作为 UI 库，并使用了一些其他的库和工具来实现其功能。

二、技术选型

1. 前端框架：React18

React 作为前端框架具有高效的渲染性能，丰富的生态系统，以及易于理解和使用的组件化开发模式。React 的虚拟 DOM 技术可以有效地减少不必要的 DOM 操作，提高页面渲染性能。

2. UI 库：Material-UI

Material-UI 作为 UI 库 提供了一套完整的 Material Design 风格的 React 组件，可以帮助快速构建出美观且易于使用的用户界面。此外，Material-UI 还提供了一些实用的工具和功能，如主题定制、响应式布局等。

3. 状态管理：React Hook 和 RTK Query

React Hooks 是 React 16.8 版本引入的新特性，它可以让我们在不编写 class 的情况下使用 state 以及其他的 React 特性。使用 Hooks，可以更简洁、更直观地管理组件状态，同时避免了 class 组件中 this 的困扰。

使用 RTK Query 数据获取。RTK Query 是 Redux Toolkit 的一部分，它提供了一种有效的方式来管理 API 请求的状态，包括缓存管理、数据同步等。使用 RTK Query，可以简化 API 调用的代码，同时也可以提高应用的性能。

三、UI 设计理念

我的 UI 设计理念是“简洁易用”，力求让用户界面尽可能简洁明了，避免不必要的复杂性。使用了 Material Design 的设计语言，这是一种现代、直观且响应式的设计语言，可以提供良好的用户体验。

四、数据可视化方式

在显示查询结果时，使用了卡片式的布局。每一条查询结果都显示在一个卡片中，卡片上显示了文献的标题、作者、摘要等信息。这种布局方式可以清晰地展示每条结果的信息，同时也方便用户浏览和比较不同的结果。

五、性能优化策略

主要采取了以下几种性能优化策略：

1. 使用虚拟 DOM

React 的虚拟 DOM 技术可以有效地减少不必要的 DOM 操作，提高页面渲染性能。

2. 使用 RTK Query 进行状态管理和数据获取

使用 RTK Query，可以简化 API 调用的代码，同时也可以提高应用的性能。RTK Query 的缓存管理和数据同步功能可以避免不必要的 API 请求，从而提高应用的响应速度。

3. 使用分页或滚动加载

当查询结果较多时，可以使用分页或滚动加载的方式，避免一次性加载过多的数据，提高页面加载速度和响应性能。

六、单元测试

使用 Jest 和 React Testing Library 进行单元测试。这两个库可以帮助编写高质量的测试用例，确保应用在各种情况下都能正常工作。对重要功能编写了测试用例，包括 DOI 查询、查找组件、结果显示。

七、主要功能：查询和显示

查询过程开始于用户在搜索框中输入 DOI 并提交表单。提交表单后，应用会调用 RTK Query 的 API 调用函数，向后端发送查询请求。RTK Query 会自动处理请求的状态，包括加载状态、错误状态等。

当后端返回查询结果后，RTK Query 会更新应用的状态，将查询结果保存在状态中。然后，应用会根据新的状态渲染查询结果。 查询结果如果返回不成功，渲染 Not Found；如果成功将以卡片的形式显示，每一条结果都显示在一个卡片中，卡片上显示了文献的标题、作者、摘要等信息。

八、补充说明

文件 State/index.js 目前是空的，它的主要目的是为了在之后可能使用 createSlice 函数来管理 Redux 状态，以实现优化和更好的状态管理。

以上就是 Scientific Citation App 项目的主要决策和考虑因素。希望通过这个项目，为用户提供一个简洁、高效、易于使用的科学引文查询工具。