链滴

React Visual - 文件的拖拽

作者：Vanessa
原文链接：https://ld246.com/article/1570603898265
来源网站：链滴

2019-10-08

描述

渲染一个单个文件的拖拽组件。

- 为组件创建一个名为 dropRef 的ref
- 使用 React.useState() hook 创建 drag 和 filename 变量，分别初始化为 false 和 ''。dragCounter 和 drag 变量用于检测一个文件是否正在被拖拽，而 filename 用于存储被拖拽后的文件的名称
- 创建 handleDrag，handleDragIn，handleDragOut 和 handleDrop 方法来处理拖拽功能，并他们绑定到组件的上下文中
- 每一个方法都会将处理一个特定的事件，在 React.useEffect() hook 中对每一个监听进行创建和除，移除置于 cleanup() 方法中
- handleDrag 阻止浏览器打开拖拽的文件，handleDragIn 和 handleDragOut 处理被拖拽的文件入和移除组件，而 handleDrop 处理正在被拖拽的文件，并将所拖拽的文件作为参数传递给 props.handleDrop
- 对 返回一个适当的样式，使用 drag 和 filename 来确定他的内容和样式
- 最后，将 dropRef 绑定到创建好的 的 ref 上

实现

```
.filedrop {
  min-height: 120px;
  border: 3px solid #d3d3d3;
  text-align: center;
  font-size: 24px;
  padding: 32px;
  border-radius: 4px;
}

.filedrop.drag {
  border: 3px dashed #1e90ff;
}

.filedrop.ready {
  border: 3px solid #32cd32;
}

function FileDrop(props) {
  const [drag, setDrag] = React.useState(false);
  const [filename, setFilename] = React.useState('');
  let dropRef = React.createRef();
  let dragCounter = 0;

  const handleDrag = e => {
    e.preventDefault();
    e.stopPropagation();
  };

  const handleDragIn = e => {
```

```
    e.preventDefault();
    e.stopPropagation();
    dragCounter++;
    if (e.dataTransfer.items && e.dataTransfer.items.length > 0) setDrag(true);
  };

  const handleDragOut = e => {
    e.preventDefault();
    e.stopPropagation();
    dragCounter--;
    if (dragCounter === 0) setDrag(false);
  };

  const handleDrop = e => {
    e.preventDefault();
    e.stopPropagation();
    setDrag(false);
    if (e.dataTransfer.files && e.dataTransfer.files.length > 0) {
      props.handleDrop(e.dataTransfer.files[0]);
      setFilename(e.dataTransfer.files[0].name);
      e.dataTransfer.clearData();
      dragCounter = 0;
    }
  };

  React.useEffect(() => {
    let div = dropRef.current;
    div.addEventListener('dragenter', handleDragIn);
    div.addEventListener('dragleave', handleDragOut);
    div.addEventListener('dragover', handleDrag);
    div.addEventListener('drop', handleDrop);
    return function cleanup() {
      div.removeEventListener('dragenter', handleDragIn);
      div.removeEventListener('dragleave', handleDragOut);
      div.removeEventListener('dragover', handleDrag);
      div.removeEventListener('drop', handleDrop);
    };
  });

  return (
    <div
      ref={dropRef}
      className={drag ? 'filedrop drag' : filename ? 'filedrop ready' : 'filedrop'}
    >
      {filename && !drag ? <div>{filename}</div> : <div>Drop files here!</div>}
    </div>
  );
}
```

使用

```
ReactDOM.render(<FileDrop handleDrop={console.log} />, document.getElementById('root'));
```

返回总目录

每天 30 秒系列之 React