

antd中使用表单

单独表单

定义表单：最好把表单单独写，然后在上面export的那个class里用

```
class FilterForm extends Component {
  render() {
    return (
      <Form layout="inline"> //FormItem和Select会显示在同一行
        <FormItem label="城市">
          <Select placeholder="全部" style={{ width: 80 }}>
            <Option value="">全部</Option>
            <Option value="1">北京市</Option>
            <Option value="2">天津市</Option>
            <Option value="3">深圳市</Option>
          </Select>
        </FormItem>
        <FormItem label="用车模式">
          <Select placeholder="全部" style={{ width: 130 }}>
            <Option value="">全部</Option>
            <Option value="1">指定停车点模式</Option>
            <Option value="2">禁停区模式</Option>
          </Select>
        </FormItem>
        <FormItem label="营运模式">
          <Select placeholder="全部" style={{ width: 80 }}>
            <Option value="">全部</Option>
            <Option value="1">自营</Option>
            <Option value="2">加盟</Option>
          </Select>
        </FormItem>
        <FormItem label="加盟商授权状态">
          <Select placeholder="全部" style={{ width: 100 }}>
            <Option value="">全部</Option>
            <Option value="1">已授权</Option>
            <Option value="2">未授权</Option>
          </Select>
        </FormItem>
        <FormItem>
          <Button type="primary" style={{ margin: '0 20px' }}>
            查询
          </Button>
        </FormItem>
      </Form>
    );
  }
}
```

```

        <Button>重置</Button>
      </FormItem>
    </Form>
  );
}
}

FilterForm = Form.create({})(FilterForm);

```

然后在export那个class里使用写好的form

```

<Card>
  <FilterForm />
</Card>

```

Modal中弹出表单

定义表单，同上面

```

class OpenCityForm extends Component {
  render() {
    const formItemLayout = { //给表单设置栅格，占几列
      labelCol: {
        span: 5
      },
      wrapperCol: {
        span: 19
      }
    };

    return (
      <Form layout="horizontal">
        <FormItem label="选择城市" {...formItemLayout}>
          <Select style={{ width: 80 }}>
            <Option value="">全部</Option>
            <Option value="1">北京市</Option>
            <Option value="2">天津市</Option>
          </Select>
        </FormItem>
      </Form>
    );
  }
}

```

```

    <FormItem label="营运模式" {...formItemLayout}>
      <Select>
        <Option value="1">自营</Option>
        <Option value="2">加盟</Option>
      </Select>
    </FormItem>
    <FormItem label="用车模式" {...formItemLayout}>
      <Select>
        <Option value="1">指定停车点</Option>
        <Option value="2">禁停区</Option>
      </Select>
    </FormItem>
  </Form>
);
}
}

OpenCityForm = Form.create({})(OpenCityForm);

```

控制Modal弹出

```

// 开通城市
handleOpenCity = () => {      //点击按钮改变isShowOpenCity值，来控制modal的弹出
  this.setState({
    isShowOpenCity: true
  });
};

<Modal
  title="开通城市"
  visible={this.state.isShowOpenCity}    //false不显示，true显示
  onCancel={() => {
    this.setState({
      isShowOpenCity: false
    });
  }}
  onOk={this.handleSubmit}
>
</Modal>

```

我们想点击确定后获取到用户输入的表单值怎么办呢？

通过antd框架自带的方法 getFieldDecorator

```
//1.定义表单, 使用Form.create
//2.在表单中使用getFieldDecorator 实现双向数据绑定
//3.在使用表单的地方使用wrappedComponentRef, 获取表单数据值

// 开通城市
handleOpenCity = () => {
  this.setState({
    isShowOpenCity: true
  });
};

// 城市开通提交
handleSubmit = () => {
  let cityInfo = this.cityForm.props.form.getFieldsValue(); //获取到表单值
  axios.ajax({
    url: '/city_open',
    data: {
      params: cityInfo
    }
  }).then((res) => {
    if(res.code == '0') {
      message.success('开通成功');
      this.setState({
        isShowOpenCity: false
      })
      this.requestList();
    }
  })
};

-----
<Modal
  title="开通城市"
  visible={this.state.isShowOpenCity}
  onCancel={() => {
    this.setState({
      isShowOpenCity: false
    });
  }}
  onOk={this.handleSubmit}
>
  <OpenCityForm
    wrappedComponentRef={inst => { //2.第二步, 绑定ref
      this.cityForm = inst;
    }}
  >
```

```

    />
  </Modal>

  -----

  class OpenCityForm extends Component {
    render() {
      const formItemLayout = {
        labelCol: {
          span: 5
        },
        wrapperCol: {
          span: 19
        }
      };

      // 辅助我们做双向数据绑定
      const { getFieldDecorator } = this.props.form;      //1.第一步

      return (
        <Form layout="horizontal">
          <FormItem label="选择城市" {...formItemLayout}>
            {
              getFieldDecorator('city_id', {
                initialValue: ''      //初始值
              })
              (
                <Select style={{ width: 100 }}>
                  <Option value="">全部</Option>
                  <Option value="1">北京市</Option>
                  <Option value="2">天津市</Option>
                </Select>
              )
            }
          </FormItem>
          <FormItem label="营运模式" {...formItemLayout}>
            {
              getFieldDecorator('op_mode', {
                initialValue: '1'
              })
              (
                <Select style={{ width: 100 }}>
                  <Option value="1">自营</Option>
                  <Option value="2">加盟</Option>
                </Select>
              )
            }
          </FormItem>
        </Form>
      );
    }
  }

```

```
        </FormItem>
        <FormItem label="用车模式" {...formItemLayout}>
          {
            getFieldDecorator('use_mode', {
              initialValue: '1'
            })
            (
              <Select style={{ width: 120 }}>
                <Option value="1">指定停车点</Option>
                <Option value="2">禁停区</Option>
              </Select>
            )
          }
        </FormItem>
      </Form>
    );
  }
}

OpenCityForm = Form.create({})(OpenCityForm);
```