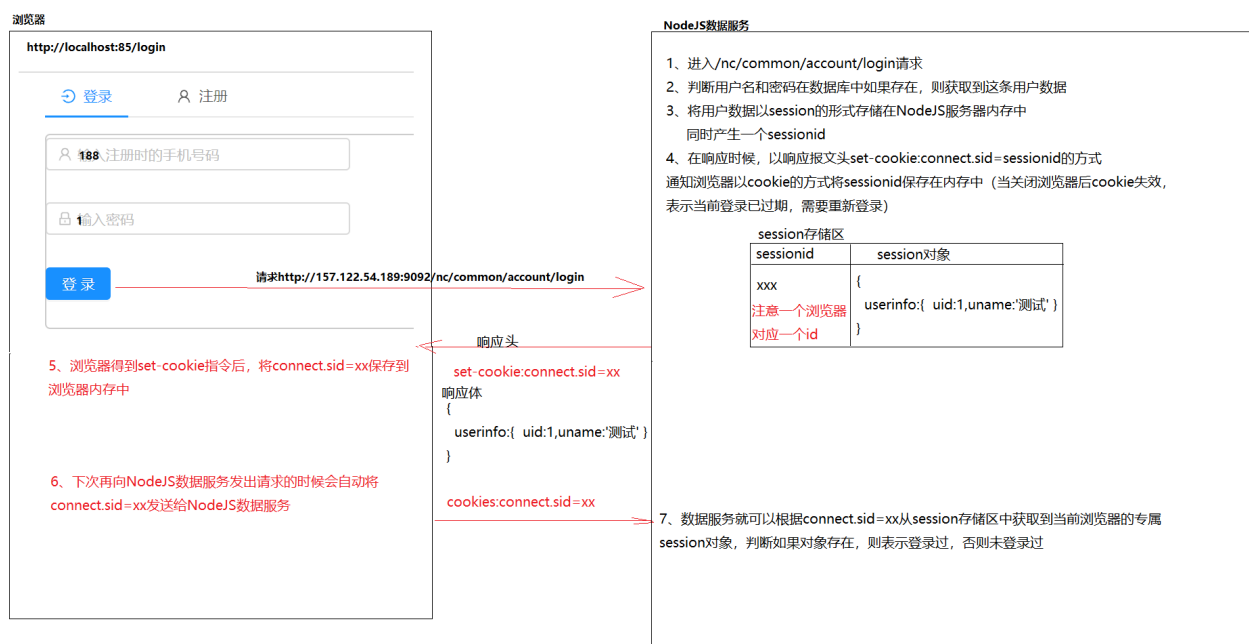


# 1. 登录

- 目标
  - 充分了解后台登录接口的业务登录过程
    - sessionId会通过set-cookie保存到浏览器中
    - 浏览器关闭以后session和sessionId失效
  - 能够利用sessionStorage保存登录以后的用户数据到浏览器内存中
  - 学会antd中的表单组件使用 <https://ant.design/components/form-cn/>

## 1.1. 登录处理过程



## 1.2. 样式文件login.less

```
.login_form{  
  border: 1px solid rgba(0,0,0,0.1);  
  border-radius: 5px;  
}
```

## 1.3. Storage 帮助类

```
// 负责操作sessionStorage的增，查，删三个操作

const KEY = 'USER_INFO'

// 1.0 增加数据到sessionStorage
// userinfo: 就是一个js对象，在此方法中会通过JSON.stringify方法将其转换成json字符串保存
export function setUser(userinfo){
    sessionStorage.setItem(KEY, JSON.stringify(userinfo || {}))
}

// 2.0 从sessionStorage中查询数据
export function getUser(){
    var userinfoJsonString = sessionStorage.getItem(KEY)
    return JSON.parse(userinfoJsonString || '{}')
}

// 3.0 从sessionStorage中删除数据
export function removeUser(){
    sessionStorage.removeItem(KEY)
}
```

## 1.4. 登录&注册基本结构

```
render() {
    return (<div style={{ minHeight: 800 }}>
        <Head><title>学成在线-用户登录</title></Head>

        <Row>
            <Col span="10" offset="7">
                /* 3.0.1 利用Tabs去做一个注册和登录的切换功能 */
                <Tabs defaultActiveKey="1">
                    <TabPane tab={<span><Icon type="vertical-left" />登录</span>} key="1">
                        编写登录界面代码
                    </TabPane>
                    <TabPane tab={<span><Icon type="android" />注册</span>} key="2">
                        编写注册界面代码
                    </TabPane>
                </Tabs>
            </Col>
        </Row>
    </div>)
}
```

## 1.5. 登录表单结构和实现逻辑

```

// 负责执行登录请求的
login = (e) => {
  e.preventDefault();
  this.props.form.validateFields((err, values) => {
    if (!err) {
      console.log('Received values of form: ', values);
      // 将values数据通过post请求，发送给/nc/common/account/login
      fetchHelper.post('/nc/common/account/login', values)
        .then(json=>{
          if(json.status == 1){
            // 表示服务器处理异常或者用户名和密码错误
            message.error(json.message)
          }else{
            // 登录成功以后
            // 1、将用户信息保存到sessionStorage中
            setUser(json.message.user)
            // 2、跳转到首页 /index
            message.success(json.message.text, 1, ()=>{
              window.location = '/index'
            })
          }
        })
    }
  });
}

const { getFieldDecorator } = this.props.form;
<Form onSubmit={this.login} className={css.login_form}>
  <FormItem>
    {getFieldDecorator('user_name', {
      rules: [
        { required: true, message: '请输入用户名!' },
        { pattern: /^1(3|4|5|7|8)\d{9}$/ , message: '用户名必须符合手机格式!' }
      ],
    })
    <Input prefix={<Icon type="user" style={{ color: 'rgba(0,0,0,.25)' }} />} placeholder="请输入用户名或者手机号" />
  }
</FormItem>
  <FormItem>
    {getFieldDecorator('password', {
      rules: [{ required: true, message: '请输入密码!' }],
    })
    <Input prefix={<Icon type="lock" style={{ color: 'rgba(0,0,0,.25)' }} />} type="password" placeholder="请输入密码" />
  }
</FormItem>
  <FormItem>

  <Button type="primary" htmlType="submit" className="login-form-button">
    登录
  </Button>

```

```
</FormItem>  
</Form>
```

- antd Form表单使用文档: <https://ant.design/components/form-cn/>

## 1.6. 完整代码login.js

---

```

// 1.0 导入样式
import css from './login.less'

import fetchHelper from '../../kits/fetchHelper.js'

// 2.0 导入antd这个ui组件中的 Tabs, Icon,Form,, Input, Button
import { Form, Icon, Input, Button, Tabs, Row, Col,message } from 'antd';
const FormItem = Form.Item;
const TabPane = Tabs.TabPane;

// 将用户对象存储到sessionStroage中
import {setUser} from '../../kits/storageHelper.js'

class login extends React.Component {

  // 负责执行登录请求的
  login = (e) => {
    e.preventDefault();
    this.props.form.validateFields((err, values) => {
      if (!err) {
        console.log('Received values of form: ', values);
        // 将values数据通过post请求，发送给/nc/common/account/login
        fetchHelper.post('/nc/common/account/login',values)
          .then(json=>{
            if(json.status == 1){
              // 表示服务器处理异常或者用户名和密码错误
              message.error(json.message)
            }else{
              // 登录成功以后
              // 1、将用户信息保存到sessionStorage中
              setUser(json.message.user)
              // 2、跳转到首页 /index
              message.success(json.message.text,1,()=>{
                window.location = '/index'
              })
            }
          })
      })
    });
  }

  render() {
    const { getFieldDecorator } = this.props.form;

    //3.0 布局整个登录页面的样式
    return (<div style={{ minHeight: '800px' }}>
      <Row>
        <Col span="10" offset="7">
          { /* 3.0.1 利用Tabs去做一个注册和登录的切换功能 */ }
          <Tabs defaultActiveKey="1">

```

```

        <TabPane tab={<span><Icon type="vertical-left" />登录</span>}} key="1">
          <Form onSubmit={this.login} className={css.login_form}>
            <FormItem>
              {getFieldDecorator('user_name', {
                rules: [
                  { required: true, message: '请输入用户名!' },
                  { pattern: /^1(3|4|5|7|8)\d{9}$/, message: '用户名必
须符合手机格式!' }
                ],
              })(
                <Input prefix={<Icon type="user" style={{ color:
'rgba(0,0,0,.25)' }} />} placeholder="请输入用户名或者手机号" />
              )}
            </FormItem>
            <FormItem>
              {getFieldDecorator('password', {
                rules: [{ required: true, message: '请输入密码!' }],
              })(
                <Input prefix={<Icon type="lock" style={{ color:
'rgba(0,0,0,.25)' }} />} type="password" placeholder="请输入密码" />
              )}
            </FormItem>
            <FormItem>

              <Button type="primary" htmlType="submit" className="login-
form-button">

                登录

              </Button>

            </FormItem>
          </Form>
        </TabPane>
        <TabPane tab={<span><Icon type="android" />注册</span>}} key="2">
        </TabPane>
      </Tabs>
    </Col>
  </Row>
<style>{
  `
    .ant-form-item{
      margin:10px 10px;
    }
  `
}
</style>
</div>
}

```

// 利用Form.create高阶函数将form对象附加到login组件中的props中

```
const WrappedNormalLoginForm = Form.create()(login);
```

```
// export default WrappedNormalLoginForm  
export default WrappedNormalLoginForm
```