

项目环境

中间件	版本
JDK	1.8
NodeJs	laset

源码使用说明

- 项目结构

```
|-- src
|   |-- main
|       |-- java
|           |-- com
|               |-- example
|                   |-- tokenuserdemo
|                       |-- TokenUserDemoApplication.java    SpringBoot启动类
|                       |-- demos
|                           |-- domain    实例层
|                               |-- ApiResult.java    请求响应
|                               |-- UserInfo.java    请求响应用户信息
|                           |-- service  服务层
|                               |-- UserService.java    token换用户服务类
|                           |-- web    展示层
|                               |-- TokenController.java    外部入口
|                   |-- resources    资源文件
|-- HELP.md 说明文件
|-- pom.xml maven依赖
```

- 修改配置文件application.yml 配置
 - server.port指定运行端口

- iam.token.path指向IAM服务器IP与端口

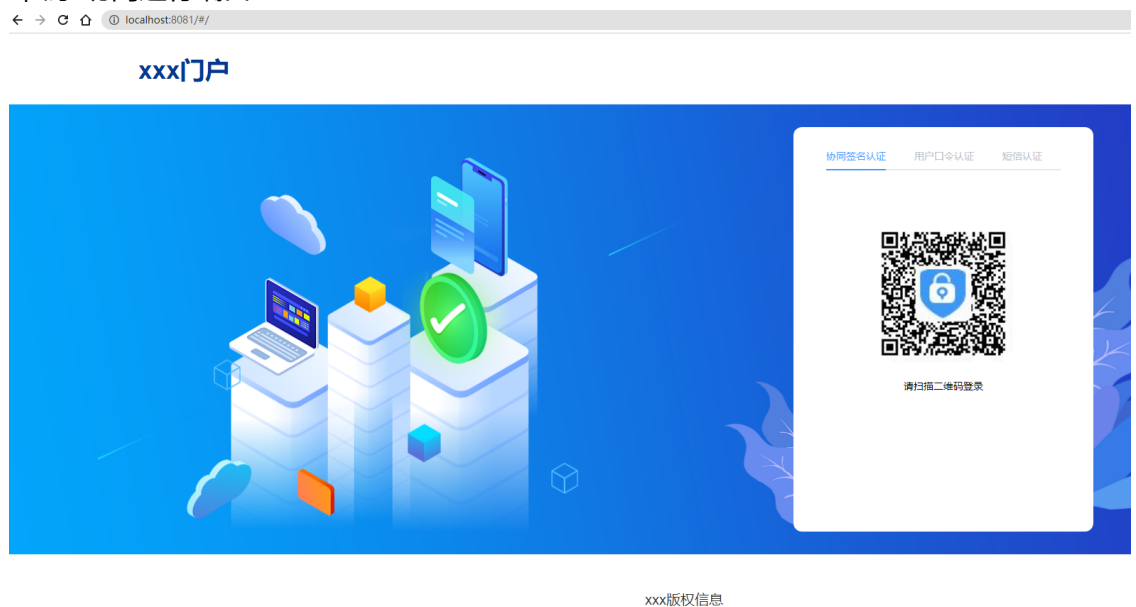
```
# 应用服务 WEB 访问端口
server:
  port: 8081
resources:
  static-locations: classpath:/resources/,classpath:/static/

# IAM服务地址 需要替换ip+port为已知IAM服务地址
iam:
  token:
    path: http://ip:port

# 访问静态资源
spring:
  thymeleaf:
    prefix: classpath:/static/
    suffix: .html
```

- 访问示例:

- 运行项目
- 展示二维码: 访问运行端口



- 换取token: 访问运行端口+请求路径(/token/XXX)
 - 示例: `http://localhost:8081/token/XXX` (xxx为获取的token)
 - 展示结果如图:

← → ↺ 🏠 ⓘ localhost:8081/token/0297a07fbb3446af8ea67c5901899813

```
{
  id: "af9e5bfa867d4d31bdad44e6cf1a7829",
  serialNumber: null,
  loginName: null,
  userName: null,
  allPinYinLoginName: null,
  firstPinYinLoginName: null,
  allPinYinUserName: null,
  firstPinYinUserName: null,
  cryptedPassword: null,
  gender: null,
  email: "null@eetrust.com",
  mobile: null,
  idCardNumber: null,
  secLevel: null,
  showNumber: null,
  status: null,
  isDelete: null,
  lastPawdTime: null,
  parentName: null,
  keyword: null,
  parentType: null,
  parentId: null,
  roleId: null,
  roleName: null,
  appId: null,
  appNames: null,
  manageAuthorize: null,
  role: null,
  telephoneNumber: null,
  lastPinTime: null,
  createTime: null,
  lastPwdTime: null
}
```

- 运行代码结构如下:

```
/**
 * 换取用户信息
 * @param token
 * @return
 */
public UserInfo token2UserInfo(String token) {

    if (null == token || token.length() <= 0) {
        throw new RuntimeException("无效token");
    }
    // 设置请求头信息
    HttpHeaders headers = new HttpHeaders();
    headers.add("Content-Type", "application/json;charset=utf8");
    headers.add("AuthToken-Authorization", token);
    RestTemplate restTemplate = new RestTemplate();
    HttpEntity<MultiValueMap<String, Object>> requestEntity = new
    HttpEntity<>( new LinkedMultiValueMap<>(3), headers);

    ResponseEntity<ApiResult> responseEntity =
    restTemplate.postForEntity(iamTokenPath,
    requestEntity, ApiResult.class);

    if (responseEntity.hasBody() &&
    !HttpStatus.OK.equals(responseEntity.getStatusCode())) {
        throw new RuntimeException("请求IAM失败");
    }
}
```

```
}  
    return responseEntity.getBody().getData();  
}
```

JAR包使用说明

- 在jar目录打开cmd命令行
- java -jar ./token-user-demo.jar --server.port=8081 --iam.token.path=http://ip:port
- 访问参考源码运行访问示例

前端代码说明

- 请求地址配置



- 功能代码说明(路径\src\components\index.vue)

```
// 获取url
getUrl() {
  let _this = this;
  let _data = {
    authType: "QRCodeAuth"
  };
  _this.API.post("/auth/api/v1/function/genLogin", _data)
    .then(res => {
      if (res.code === 20000) {
        // 获取二维码图片信息
        _this.qrLoginUrl = "data:image/png;base64," + res.data.qrCode;
        _this.qrLoginStateUrl = res.data.qrLoginStateUrl;
        // this.qrcode();
        this.getStatus();
      }
    })
    .catch(err => {
      this.$message({
        message: "获取地址失败!",
        type: "warning"
      });
    });
},

// 轮询二维码登录状态
getStatus() {
  let _this = this;
  let _authRandom = _this.qrLoginStateUrl.split("=")[1];
  let _data = {
    userContent: {
      authRandom: _authRandom
    }
  };
  // 轮询二维码状态
  _this.timer = setInterval(() => {
    _this.API.post("/auth/api/v1/function/qr_code", _data)
      .then(res => {
        if (res.code === 20000) {
          if (res.data.qrStatus === "-1") {
            this.$message({
              message: "操作超时!",
              type: "warning"
            });
          } else if (res.data.qrStatus === "0") {
            // 扫码成功 获取用户信息
            _this.token = res.data.token;
            clearInterval(_this.timer);
            _this.getUser();
          }
        }
      })
  }, 1000);
}
```

```
// 获取登录用户信息
```

```
getUser() {  
  let _this = this;  
  let _data = {};  
  let _header = {  
    "AuthToken-Authorization": _this.token  
  };  
  _this.API.post("/auth/api/v1/function/getUserInfo", _data, _header)  
    .then(res => {  
      if (res.code === 20000) {  
        _this.$message({  
          message: res.data.userName + "登录成功!",  
          type: "success"  
        });  
        _this.username = res.data.userName;  
        _this.qrcodeShow = false;  
        _this.$router.push({  
          path: "/main",  
          query: {  
            userName: res.data.userName  
          }  
        });  
      }  
    })  
    .catch(() => {});  
}
```

获取到用户信息 登录成功 跳转页面

```
<div class="panleCls">  
  <!-- 切换 -->  
  <el-tabs v-model="selectName" @tab-click="handleloginClick">  
    <el-tab-pane label="协同签名认证" name="third">  
      <div v-if="qrcodeShow" class="fd-code">  
        <!-- <div id="qrcode" ref="qrcode"></div> -->  
          
        <p>请扫描二维码登录</p>  
      </div>  
      <p v-if="!qrcodeShow">{{ username }}已经登录</p>  
    </el-tab-pane>  
    <el-tab-pane label="用户口令认证" disabled> </el-tab-pane>  
    <el-tab-pane label="短信认证" disabled> </el-tab-pane>  
  </el-tabs>  
</div>  
</div>  
<div class="footer">xxx版权信息</div>  
</div>
```

二维码图片展示