## 前置：

Api：6

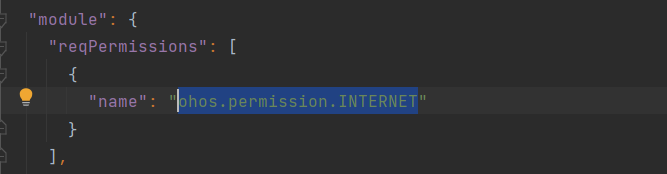
语言：js开发

添加编译依赖参考地址：

<https://developer.huawei.com/>consumer/cn/doc/development/HMSCore-Guides/harmonyos-js-integration-hms-core-sdk-0000001149139136

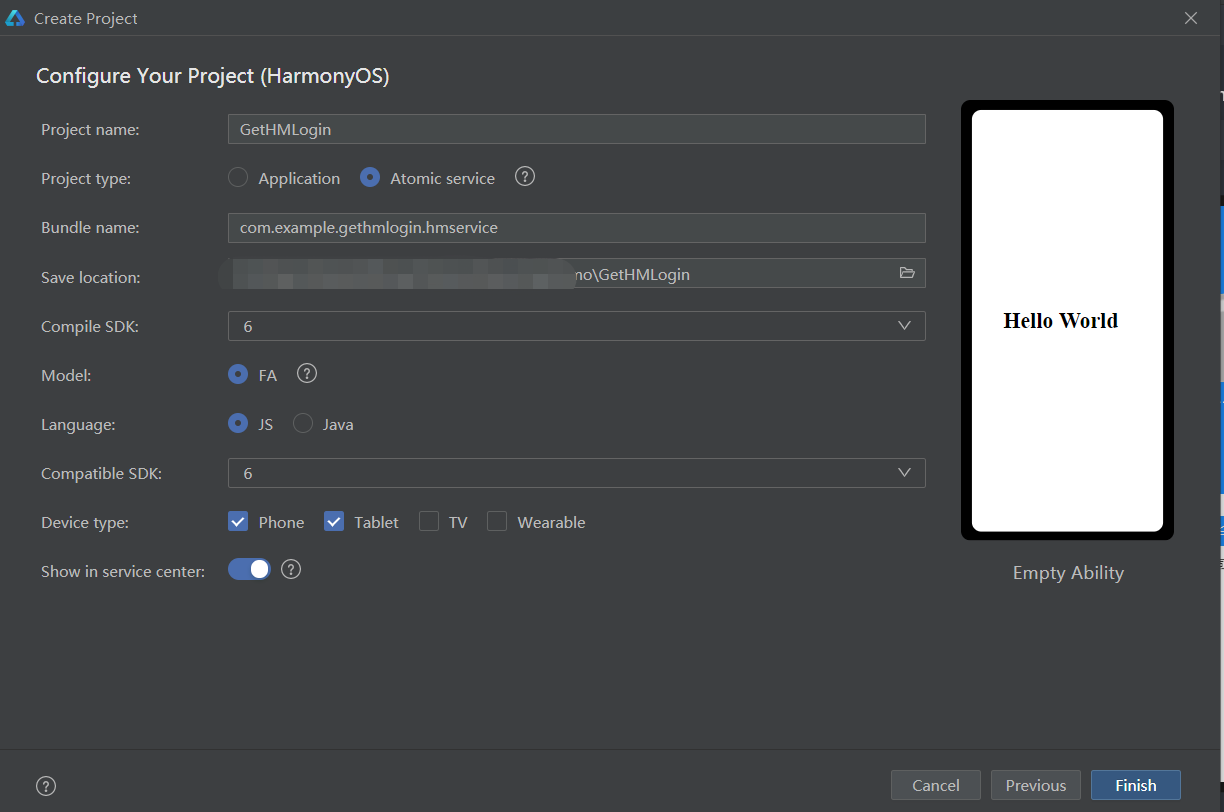
需要权限：

ohos.permission.INTERNET



## 开始：

1.创建项目：



2.示例代码

test.hml

|  |
| --- |
| <div class="container"> <!-- 顶部导航-标题-->  <div class="header-router">  <div class="header-box">  <text class="header-box-back" onclick="back">  <span>＜返回</span>  </text>  </div>  </div>  <div class="login-box">  <image class="user-img" src="{{ userImg }}"></image>  <text class="user-name" onclick="goGetUserNumber">  <span>{{ userName }}</span>  </text>  </div>  <text class="btnText" onclick="getNumber">  <span>一键授权华为账号登录</span>  </text>  <div class="changeDiv">  <input type="checkbox" checked="{{ isCheck }}" onchange="changeBtn"></input>  <text class="privacy">  <span>我已阅读并同意</span>  </text>  <text class="privacy" onclick="goPrivacyContent">  <span style="color: dodgerblue;">《用户隐私协议》</span>  </text>  </div> </div> |

test.css

|  |
| --- |
| @import '../../common/css/header-router.css'; .container {  flex-direction: column;  justify-content: flex-start;  align-items: center;  width: 100%; } .login-box{  flex-direction: column;  justify-content: center;  align-items: center;  width: 100%;  height: 50%;  background-color: dodgerblue; } .user-img{  width: 120px;  height: 120px;  border-radius: 60px;  object-fit: cover; /\* border: 1px;\*/  background-color: #eee; }  .user-name{  margin-top: 10px;  width: 150px;  font-size: 20px;  color: #fff;  text-align: center;  max-lines: 1;  text-overflow: ellipsis; } .btnText{  width: 240px;  height:40px;  text-align: center;  color: #fff;  font-size: 20px;  background-color: dodgerblue;  border-radius: 4px;  margin-top: 30px; } .btnText:active{  background-color: #999; } .changeDiv{  flex-direction: row;  align-items: center;  justify-content: center; } .privacy{  font-size: 16px;  color: #333;  text-decoration: underline; } |

test.js

|  |
| --- |
| import prompt from '@system.prompt'; import router from '@system.router'; import { HuaweiIdAuthParamsHelper, HuaweiIdAuthManager } from '@hmscore/hms-jsb-account';   export default {  data: {  isLogin: false,//是否登录  userImg: "common/images/userH.png",  userName: "......",  signInResult: null,  isCheck: false, //是否勾选协议  // timer:null,  },  onShow() {  },   //获取用户华为账号  getNumber() {  let that = this;  let signInOption = new HuaweiIdAuthParamsHelper().setId().setProfile().setAuthorizationCode().build();  let num = 4;  let btnT = this.isCheck;  if (btnT == true) {   // HuaweiIdAuthManager.getAuthApi方法返回huaweiIdAuth对象，在huaweiIdAuth对象中调用huaweiIdAuth.getSignInIntent方法  HuaweiIdAuthManager.getAuthApi().getSignInIntent(signInOption)  .then((result) => {  // 登录成功，获取用户的华为帐号信息  console.log("授权登录成功")  console.info("signIn success");  console.info(JSON.stringify(result));  console.info("昵称: " + result.data);  console.info("头像: " + result.status);  prompt.showToast({  message: "授权登录中...", duration: 2000  });  let setTime = setInterval(() => {  num--;  if (num == 2) {  prompt.showToast({  message: "登陆成功", duration: 2000  });  that.userName = result.data  that.userImg = result.status  that.isLogin = true  }  if (num == 0) {  router.replace({  uri: "pages/index/index",  params: {  userImg: that.userImg,  userName: that.userName,  isLogin: that.isLogin  }  });  clearInterval(setTime);  }  }, 1000);   }).catch((error) => {  // 登录失败  console.error("signIn fail");  console.error(JSON.stringify(error));  prompt.showToast({  message: "授权登录中...", duration: 2000  });  let setTime = setInterval(() => {  num--;  if (num == 2) {  prompt.showToast({  message: "授权登录失败：" + error, duration: 8000,  });  }  if(num == 0){  clearInterval(setTime);  }  }, 1000);  });  } else {  prompt.showToast({  message: "请勾选用户隐私协议"  })  console.log("请先勾选用户隐私协议")  }  },  //监听是否勾选隐私声明  changeBtn(e) {  this.isCheck = !this.isCheck  if (this.isCheck) {  console.log("已勾选阅读")  } else {  console.log("未勾选阅读")  }  },   //跳转用户隐私协议  goPrivacyContent(){   },   //返回上一个page  back() {  router.back();  } } |

3.效果如图：（用的previewer，所以肯定报错哈，要用有账号的真机测试哈）

