

实验五报告

学号：<3225706084>

姓名：<陈芯悦>

指导老师：<张凯斌老师>

实验日期：<2025-04-13>

一、实验目的

- 完成移动APP的原始框架，将作为后续系统的承载容器；
- 掌握教科书中网络编程相关部分练习；

二、实验内容

- 阅读教科书的第7章“网络编程”；
- 结合理论课要求完成本次APP框架设计；

三、实验要求

- Loading Page：完成起始页展示
 - 展示个性化设计
 - 可采用点击或定时方式完成跳转
- Login/Registration Page：实现登录/注册功能
 - 本次实验只要求完成页面设计
 - 具体功能在后续实验中完成
 - 主要涉及关系数据库操作
- Home Page：主界面，承担APP主要功能展示（以按钮形式）和跳转
 - 本次实验实现三个功能（按钮）
 - 请求图像
 - 要求使用Image组件的ArrayBuffer形式，不能直接使用URL返回
 - Image组件可直接放在Home Page
 - 请求JSON
 - 采用RichText组件展示请求后的数据
 - RichText在新页面，需跳转
 - 理解异步/同步概念，实现页面跳转后的数据正确展示
 - 请求Web
 - 采用Web组件，直接请求本地HTML，展示2019年世界各国农作物产量预测系统
 - HTML文件在实验五Git课程仓库
 - Web组件在新页面，需跳转
 - 初步理解Web应用，为后续WebGIS二次开发打基础
- 撰写并提交实验报告；

- 实验步骤部分需严格按模板要求撰写

四、实验步骤

1. Loading Page

1.1 截图展示

03:23

100%



实验五（猫和老鼠版）

准备就绪
即将进入汤姆与杰瑞的冒险世界...

冒险世界：
1、每日清单
2、JSON
3、Web网页
4、请求图片

进入冒险

实验五：网络编程

1.2 代码实现

```
import router from '@ohos.router';
import { prompt } from '@kit.ArkUI';

@Entry
@Component
struct Index {
  @State private isLoading: boolean = true;

  onPageShow() {
    // 模拟资源加载
    setTimeout(() => {
      this.isLoading = false;
    }, 2000);
  }

  build() {
    column() {
      Flex({ direction: FlexDirection.Column, justifyContent: FlexAlign.Center,
        alignItems: ItemAlign.Center }) {
        if (this.isLoading) {
          // 加载状态
          LoadingProgress()
            .color('#F4B41B')
        }
      }
    }
  }
}
```

```

        .height(40)
        .width(300)
        .margin({ bottom: 30 })

Text('正在加载资源...')
    .fontSize(16)
    .fontColor('#4A5759')
} else {
    // 加载完成后显示内容
    Image($r('app.media.tom_and_jerry'))
        .width(250)
        .height(300)
        .margin({ right: 15 })
        .objectFit(ImageFit.Contain)
        .onError(() => {
        })

    // 标题文字
    Text('实验五（猫和老鼠版）')
        .fontSize(35)
        .fontWeight(FontWeight.Bold)
        .fontColor('#2E4053')
        .margin({ top: 15 })

    // 副标题
    Text('准备就绪\n即将进入汤姆与杰瑞的冒险世界...')
        .fontSize(18)
        .fontColor('#4A5759')
        .margin({ top: 10, bottom: 10 })

    Text('冒险世界：\n1、每日清单\n2、JSON\n3、web网页\n4、请求图片')
        .fontSize(18)
        .fontColor('#4A5759')
        .margin({ top: 5, bottom: 10 })

    // 进入按钮
    Button('进入冒险')
        .width(250)
        .height(80)
        .margin({ bottom: 40 })
        .fontSize(25)
        .fontColor(Color.White)
        .backgroundColor('#4285F4')
        .borderRadius(25)
        .onClick(() => {
            try {
                router.pushUrl({
                    url: 'pages/LoginPage'
                });
            } catch (error) {
                console.error('Navigation failed:', error);
                prompt.showToast({
                    message: '页面跳转失败，请重试',
                    duration: 2000
                });
            }
        })
}

```

```
        })
    }
}
.width('100%')
.height('100%')

// 底部信息
Text('实验五：网络编程')
    .fontSize(20)
    .fontColor('#6C757D')
    .margin({ bottom: 10 })
}
}
}
```

2. Login/Registration Page

2.1 截图展示

03:24

100%

猫和老鼠冒险之旅



请输入冒险家账号

请输入密码



开始冒险

还不是冒险家？[点击注册](#)



新玩家注册



请输入冒险家账号

请输入密码



请确认密码



加入冒险

已是冒险家? 点击登录

2.2 代码实现

```
import router from '@ohos.router';

@Entry
@Component
struct LoginPage {
  @State isLoginMode: boolean = true;
  @State account: string = '';
  @State password: string = '';
  @State confirmPassword: string = '';

  build() {
    column() {
      Blank().flexGrow(0.3)

      // 页面标题
      Text(this.isLoginMode ? '猫和老鼠冒险之旅' : '新玩家注册')
        .fontSize(32)
        .fontWeight(FontWeight.Bold)
        .fontColor('#2E4053')
        .margin({ bottom: 30 })

      // 登录/注册表单
      column() {
        // 角色头像
        Image($r('app.media.tom'))
          .width(150)
          .height(150)
          .borderRadius(75)
          .margin({ bottom: 40 })
          .objectFit(ImageFit.Cover)
```

// 账号输入框

```
TextInput({ placeholder: '请输入冒险家账号', text: this.account })
  .type(InputType.Normal)
  .placeholderColor('#4A5759')
  .fontColor('#2E4053')
  .padding({ left: 20, right: 20 })
  .height(55)
  .width('85%')
  .margin({ bottom: 25 })
  .backgroundColor('#F0F3F4')
  .borderRadius(28)
  .onChange((value: string) => { this.account = value; })
```

// 密码输入框

```
TextInput({ placeholder: '请输入密码', text: this.password })
  .type(InputType.Password)
  .placeholderColor('#4A5759')
  .fontColor('#2E4053')
  .padding({ left: 20, right: 20 })
  .height(55)
  .width('85%')
  .margin({ bottom: 25 })
  .backgroundColor('#F0F3F4')
  .borderRadius(28)
  .onChange((value: string) => { this.password = value; })
```

// 注册模式下的确认密码输入框

```
if (!this.isLoginMode) {
  TextInput({ placeholder: '请确认密码', text: this.confirmPassword })
    .type(InputType.Password)
    .placeholderColor('#4A5759')
    .fontColor('#2E4053')
    .padding({ left: 20, right: 20 })
    .height(55)
    .width('85%')
    .margin({ bottom: 25 })
    .backgroundColor('#F0F3F4')
    .borderRadius(28)
    .onChange((value: string) => { this.confirmPassword = value; })
}
```

// 登录/注册按钮

```
Button(this.isLoginMode ? '开始冒险' : '加入冒险')
  .width('85%')
  .height(55)
  .fontSize(20)
  .fontWeight(FontWeight.Medium)
  .fontColor(Color.White)
  .backgroundColor('#4285F4')
  .borderRadius(28)
  .margin({ top: 20, bottom: 30 })
  .onClick(() => {
    if (this.isLoginMode) {
      router.pushUrl({ url: 'pages/HomePage' });
    } else {
      this.isLoginMode = true;
    }
  })
```

```

    }
  })

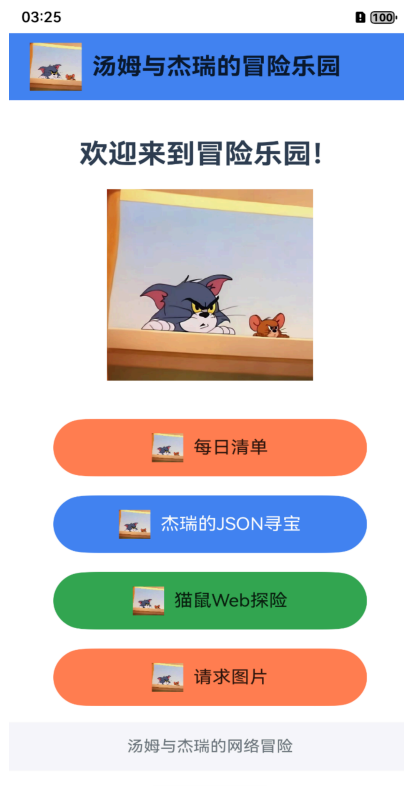
  // 切换登录/注册模式的文本
  Text(this.isLoginMode ? '还不是冒险家? 点击注册' : '已是冒险家? 点击登录')
    .fontSize(16)
    .fontColor('#F4B41B')
    .fontWeight(FontWeight.Medium)
    .margin({ top: 10 })
    .onClick(() => {
      this.isLoginMode = !this.isLoginMode;
    })
  }
  .width('100%')
  .alignItems(HorizontalAlign.Center)

  Blank().flexGrow(1)
}
.width('100%')
.height('100%')
}
}

```

3. Home Page

3.1 截图展示



3.2 代码实现

```

import router from '@ohos.router';

// 定义路由参数接口
interface RouterParams {

```

```

data: string;
}

@Entry
@Component
struct HomePage {
  build() {
    Column() {
      // 顶部标题栏
      Row() {
        Image($r('app.media.tom_and_jerry'))
          .width(50)
          .height(50)
          .margin({ right: 10 })
        Text('汤姆与杰瑞的冒险乐园')
          .fontSize(24)
          .fontWeight(Fontweight.Bold)
      }
      .width('100%')
      .height(70)
      .padding({ left: 20 })
      .backgroundColor('#4285F4')
      .alignItems(VerticalAlign.Center)

      // 内容区域
      Column() {
        // 欢迎标语
        Text('欢迎来到冒险乐园! ')
          .fontSize(28)
          .fontWeight(Fontweight.Bold)
          .fontColor('#2E4053')
          .margin({ top: 40, bottom: 20 })

        // 猫和老鼠图标
        Image($r('app.media.tom_and_jerry'))
          .width(200)
          .height(200)
          .objectFit(ImageFit.Contain)
          .margin({ bottom: 40 })

        // 冒险选项按钮
        Column({ space: 20 }) {
          // 任务清单按钮
          Button() {
            Row() {
              Image($r('app.media.tom_and_jerry')).width(30).height(30)
              Text('每日清单').fontSize(18).margin({ left: 10 })
            }
          }
        }
      }
      .width('100%').justifyContent(FlexAlign.Center).alignItems(VerticalAlign.Center)
      .width('85%')
      .height(60)
      .backgroundColor('#FF7F50') // 使用珊瑚色
      .borderRadius(30)
      .onClick(() => {
        router.pushUrl({ url: 'pages/TaskPage' });
      });
    }
  }
}

```



```

    })

    // JSON冒险按钮
    Button() {
      Row() {
        Image($r('app.media.tom_and_jerry')).width(30).height(30)
        Text('杰瑞的JSON寻宝').fontSize(18).fontColor(Color.white).margin({
left: 10 })
      }
    }.width('100%').justifyContent(FlexAlign.Center).alignItems(VerticalAlign.Center)
  }
  .width('85%')
  .height(60)
  .backgroundColor('#4285F4')
  .borderRadius(30)
  .onClick(() => {
    router.pushUrl({
      url: 'pages/JsonPage',
      params: {
        data: '这是杰瑞藏起来的 JSON 数据~'
      } as RouterParams
    }).catch((err: Error) => { // 使用具体的 Error 类型
      console.error(`跳转失败: ${err.message}`);
    });
  })

  // Web冒险按钮
  Button() {
    Row() {
      Image($r('app.media.tom_and_jerry')).width(30).height(30)
      Text('猫鼠Web探险').fontSize(18).margin({ left: 10 })
    }
  }.width('100%').justifyContent(FlexAlign.Center).alignItems(VerticalAlign.Center)
  }
  .width('85%')
  .height(60)
  .backgroundColor('#34A853')
  .borderRadius(30)
  .onClick(() => {
    router.pushUrl({ url: 'pages/WebPage' });
  })

  // 请求图片按钮
  Button() {
    Row() {
      Image($r('app.media.tom_and_jerry')).width(30).height(30)
      Text('请求图片').fontSize(18).margin({ left: 10 })
    }
  }.width('100%').justifyContent(FlexAlign.Center).alignItems(VerticalAlign.Center)
  }
  .width('85%')
  .height(60)
  .backgroundColor('#FF7F50')
  .borderRadius(30)
  .onClick(() => {
    router.pushUrl({ url: 'pages/ImagePage' });
  })

```

```

        })
    }
    .width('100%')
    .alignItems(HorizontalAlign.Center)
}
.layoutWeight(1)
.width('100%')
.padding({ left: 16, right: 16 })

// 底部信息
Row() {
    Text('汤姆与杰瑞的网络冒险')
        .fontSize(16)
        .fontColor('#6C757D')
    }
    .width('100%')
    .height(50)
    .backgroundColor('#F8F9FA')
    .justifyContent(FlexAlign.Center)
}
.width('100%')
.height('100%')
}
}

```

4. Function Pages

Function Pages 是各具体功能实现的页面，盛放各自所需组件；

须从Home Page跳转；

本实验有2个Function Pages

- JSON Page
- Web Page

自设2个Function Pages

- TaskPage
- ImagePage（因为前面主页用的是图片文件，而不是请求网页得到的，所以在此设置）

4.1 JSON Page

4.1.1 截图展示

```
[
  {
    "name": "Michael"
  },
  {
    "name": "Andy Huang",
    "age": 25,
    "homePage": "https://waylau.com/books"
  },
  {
    "name": "Justin",
    "age": 19
  },
  {
    "name": "Way Lau",
    "age": 35,
    "homePage": "https://waylau.com"
  }
]
```

4.1.2 代码实现

```
import router from '@ohos.router';
import http from '@ohos.net.http';

// 定义用户信息接口
interface UserInfo {
  name: string;
  age?: number;
  homePage?: string;
}

// 定义请求头部接口
interface RequestHeader {
  'Content-Type': string;
  'User-Agent': string;
}

// 定义请求选项接口
interface RequestOptions {
  method: http.RequestMethod;
  header: RequestHeader;
  connectTimeout: number;
  readTimeout: number;
}

@Entry
@Component
struct JsonPage {
  @State receivedData: string = '加载中...'
  private jsonUrl: string = 'https://raw.githubusercontent.com/kaibin-zhang/MobileAppDevelopment/main/example.json'
```

```

// 添加离线数据
private fallbackData: UserInfo[] = [
  { name: "Michael" },
  { name: "Andy Huang", age: 25, homePage: "https://waylau.com/books" },
  { name: "Justin", age: 19 },
  { name: "Way Lau", age: 35, homePage: "https://waylau.com" }
];

aboutToAppear() {
  this.fetchJsonData()
}

fetchJsonData() {
  // 创建 HTTP 请求对象
  let httpRequest = http.createHttp();

  // 添加请求头部
  const header: RequestHeader = {
    'Content-Type': 'application/json',
    'User-Agent': 'Mozilla/5.0'
  }

  // 添加请求选项
  const options: RequestOptions = {
    method: http.RequestMethod.GET,
    header: header,
    connectTimeout: 60000, // 连接超时时间: 60s
    readTimeout: 60000 // 读取超时时间: 60s
  }

  try {
    // 发送请求
    httpRequest.request(
      this.jsonUrl,
      options,
      (err, data) => {
        if (!err && data.responseCode === 200) {
          // 成功获取数据
          this.receivedData =
            JSON.stringify(JSON.parse(data.result.toString()), null, 2);
          console.info('成功获取数据');
        } else {
          // 获取数据失败
          this.receivedData = '获取数据失败: ' + (err ? err.message : '未知错误');
          console.error('请求失败:', err);
          this.onError();
        }
      }
    )
  } catch (error) {
    this.receivedData = '请求异常: ' + error.message;
    console.error('请求异常:', error);
    this.onError();
  } finally {
    // 释放资源
    httpRequest.destroy();
  }
}

```

```

    }
}

// 在获取失败时使用离线数据
onError() {
    this.receiveData = JSON.stringify(this.fallbackData, null, 2);
    console.info('使用离线数据');
}

build() {
    Column() {
        // 顶部标题栏
        Row() {
            Button() {
                Row() {
                    Text('<')
                        .width(20)
                        .height(20)
                        .fontSize(20)
                    Text('返回')
                        .fontSize(16)
                        .fontColor(Color.White)
                        .margin({ left: 5 })
                }
            }
            .width(80)
            .height(35)
            .backgroundColor('#4285F4')
            .borderRadius(18)
            .onClick(() => {
                router.back();
            })

            Text('远程JSON数据')
                .fontSize(24)
                .fontWeight(FontWeight.Bold)
                .fontColor(Color.White)
        }
        .width('100%')
        .height(60)
        .padding({ left: 20, right: 20 })
        .backgroundColor('#4285F4')
        .alignItems(VerticalAlign.Center)

        // 内容区域
        Column() {
            Text(this.receiveData)
                .fontSize(16)
                .fontColor('#2E4053')
                .margin({ top: 20, left: 20, right: 20 })
                .backgroundColor('#F8F9FA')
                .padding(20)
                .borderRadius(12)
        }
        .width('100%')
        .layoutWeight(1)
    }
}

```

```

    }
    .width('100%')
    .height('100%')
  }
}

```

4.2 Web Page

4.2.1 截图展示



4.2.2 代码实现

```

import web_webview from '@ohos.web.webview';
import router from '@ohos.router';

// 定义页面事件接口
interface PageEvent {
  url: string;
}

@Entry
@Component
struct WebPage {
  private webController: web_webview.WebviewController = new
web_webview.WebviewController();

  build() {
    column() {
      // 顶部标题栏
      Row() {
        Button() {
          Row() {
            Text('<')
              .width(20)

```

```

        .height(20)
        .fontSize(20)
        .fontColor(Color.White)
Text('返回')
        .fontSize(16)
        .fontColor(Color.White)
        .margin({ left: 5 })
    }
}
.width(80)
.height(35)
.backgroundColor('#4285F4')
.borderRadius(18)
.onClick(() => {
    router.back();
})

Text('农作物产量数据')
    .fontSize(24)
    .fontWeight(FontWeight.Bold)
    .fontColor('#2E4053')
    .margin({ left: 15 })
}
.width('100%')
.height(60)
.padding({ left: 16, right: 16 })
.backgroundColor('#F8F9FA')
.alignItems(VerticalAlign.Center)

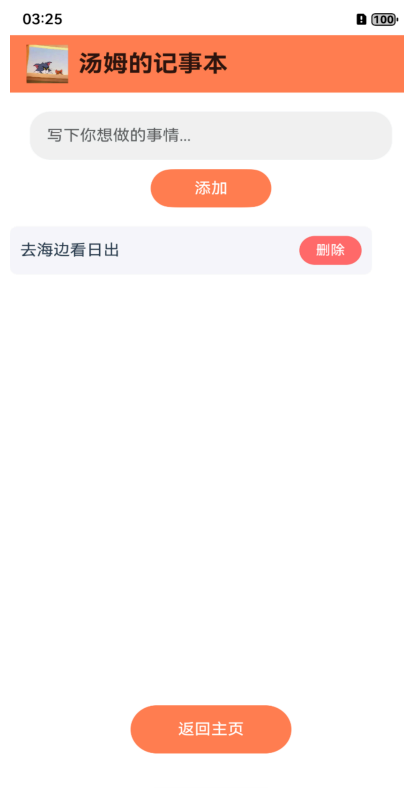
// Web内容区域
web({ src: $rawfile('Agricultural_Crop_Yields_webApp.html'), controller:
this.webController })
    .width('100%')
    .layoutWeight(1)
    .onPageBegin((event: PageEvent) => {
        console.info('开始加载页面:', event.url);
    })
    .onPageEnd((event: PageEvent) => {
        console.info('页面加载完成:', event.url);
    })
    .onErrorReceive((event) => {
        console.error('加载错误:', event);
    })
}
.width('100%')
.height('100%')
}

aboutToDisappear() {
    // 移除控制器的销毁方法, 因为 webViewController 不支持
}
}

```

4.3 TaskPage

4.3.1 截图展示



4.3.2 代码实现

```
import router from '@ohos.router';
import promptAction from '@ohos.promptAction';

@Entry
@Component
struct TaskPage {
  // 定义任务列表和新任务输入
  @State taskList: Array<string> = [];
  @State newTask: string = '';

  build() {
    column() {
      // 顶部标题栏
      row() {
        Image($r('app.media.tom_and_jerry'))
          .width(40)
          .height(40)
          .margin({ right: 10 })
        Text('汤姆的记事本')
          .fontSize(24)
          .fontWeight(Fontweight.Bold)
      }
      .width('100%')
      .height(60)
      .padding({ left: 16 })
      .backgroundColor('#FF7F50')
      .alignItems(VerticalAlign.Center)
    }
  }
}
```



```

// 输入区域
Column() {
  TextInput({ placeholder: '写下你想做的事情...', text: this.newTask })
    .width('90%')
    .height(50)
    .margin({ top: 20 })
    .backgroundColor('#F0F3F4')
    .onChange((value: string) => {
      this.newTask = value;
    })

  Button('添加')
    .width('30%')
    .height(40)
    .margin({ top: 10 })
    .backgroundColor('#FF7F50')
    .onClick(() => {
      if (this.newTask.trim() !== '') {
        this.taskList.push(this.newTask);
        this.newTask = '';
        promptAction.showToast({ message: '添加成功! ' });
      }
    })
}

.width('100%')
.alignItems(HorizontalAlign.Center)

// 任务列表
List({ space: 10, initialIndex: 0 }) {
  ForEach(this.taskList, (task: string, index: number) => {
    ListItem() {
      Row() {
        Text(task)
          .fontSize(16)
          .fontColor('#2E4053')
          .layoutWeight(1)

        Button('删除')
          .width(60)
          .height(30)
          .fontSize(14)
          .backgroundColor('#FF6B6B')
          .onClick(() => {
            if (typeof index === 'number') {
              this.taskList.splice(index, 1);
              promptAction.showToast({
                message: '删除成功!',
                duration: 2000
              });
            }
          })
      }
    }
  })
}

.width('90%')
.padding(10)
.borderRadius(8)
.backgroundColor('#F8F9FA')

```

```

        }
      })
    }
    .width('100%')
    .height('60%')
    .layoutWeight(1)
    .margin({ top: 20 })

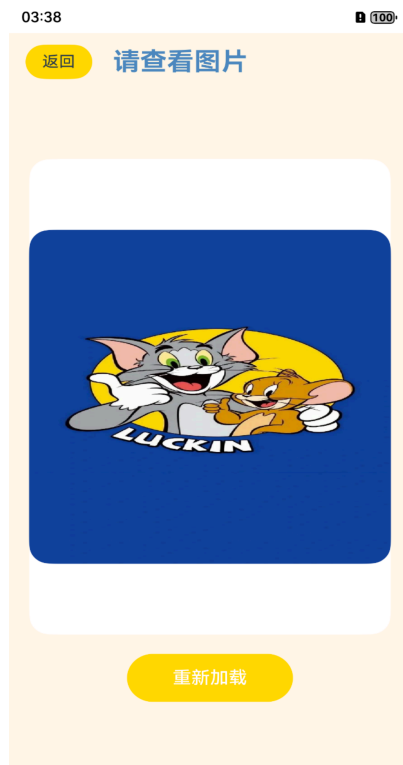
    // 提示文本
    if (this.taskList.length === 0) {
      Text('还没有添加任何事项哦~')
        .fontSize(16)
        .fontColor('#6C757D')
        .margin({ top: 100 })
    }

    // 返回按钮
    Button('返回主页')
      .width('40%')
      .height(50)
      .backgroundColor('#FF7F50')
      .margin({ bottom: 20 })
      .onClick(() => {
        router.back();
      })
    }
    .width('100%')
    .height('100%')
  }
}

```

4.4 ImagePage

4.4.1 截图展示



4.4.2 代码实现

```
import { http } from '@kit.NetworkKit';
import { image } from '@kit.ImageKit';
import { router } from '@kit.ArkUI';
import prompt from '@ohos.prompt';

@Entry
@Component
struct image_page {
  @State private pixelMapImg: PixelMap | undefined = undefined;
  @State private isLoading: boolean = false;
  @State private hasError: boolean = false;

  private imageUrl: string =
'https://img2.baidu.com/it/u=2141068052,1976678131&fm=253&fmt=auto&app=138&f=JPEG?w=800&h=1732';

  aboutToAppear() {
    // 页面显示时自动加载图片
    this.loadImage();
  }

  async loadImage() {
    if (this.isLoading) return;

    this.isLoading = true;
    this.hasError = false;
    this.pixelMapImg = undefined; // 重置图片

    try {
      const response = await http.createHttp().request(this.imageUrl, {
        method: http.RequestMethod.GET,
```

```

        connectTimeout: 10000,
        readTimeout: 10000,
        header: {
            'Content-Type': 'application/json'
        }
    });

    if (response.responseCode === http.ResponseCode.OK) {
        const imgSource = image.createImageSource(response.result as
        ArrayBuffer);
        if (imgSource) {
            try {
                const pixelMap = await imgSource.createPixelMap({
                    desiredSize: { height: 1000, width: 1000 },
                    editable: false
                });
                if (pixelMap) {
                    this.pixelMapImg = pixelMap;
                } else {
                    throw new Error('Failed to create pixel map');
                }
            } catch (error) {
                console.error('Create pixel map failed:', error);
                this.hasError = true;
                prompt.showToast({
                    message: '图片处理失败, 请重试',
                    duration: 2000
                });
            }
        } else {
            throw new Error('Failed to create image source');
        }
    } else {
        throw new Error(`HTTP request failed with code
        ${response.responseCode}`);
    }
} catch (error) {
    console.error('Load image failed:', error);
    this.hasError = true;
    prompt.showToast({
        message: '图片加载失败, 请检查网络连接',
        duration: 2000
    });
} finally {
    this.isLoading = false;
}

}

@Builder
private Title() {
    Row() {
        Button() {
            Row() {
                Text("返回").fontSize(16).fontColor('#2C3E50')
            }.padding({ left: 4 })
        }
    }
}

```

```

        .height(36)
        .backgroundColor('#FFD700')
        .borderRadius(18)
        .padding({ left: 12, right: 16 })
        .onClick(() => router.back())

        Text('请查看图片')
            .fontSize(26)
            .fontWeight(FontWeight.Bold)
            .fontColor('#4B88C1')
            .margin({ left: 20 })
    }
    .width('100%')
    .height(60)
    .backgroundColor('#FFF8E7')
    .padding({ left: 16, right: 16 })
    .alignItems(VerticalAlign.Center)
}

@Builder
private ImageContent() {
    Column() {
        Image(this.pixelMapImg)
            .objectFit(ImageFit.Contain)
            .width('90%')
            .height('70%')
            .borderRadius(20)
            .backgroundColor(Color.White)

        Button('重新加载')
            .width(160)
            .height(50)
            .fontSize(18)
            .backgroundColor('#FFD700')
            .margin({ top: 20 })
            .onClick(() => this.loadImage())
    }
    .width('100%')
    .alignItems(HorizontalAlign.Center)
}

build() {
    Column() {
        this.Title()

        Stack({ alignContent: Alignment.Center }) {
            Column()
                .width('100%')
                .height('100%')
                .backgroundColor('#FFF8E7')

            if (this.isLoading) {
                LoadingProgress()
                    .width(50)
                    .height(50)
                    .color('#FF6B4A')
            }
        }
    }
}

```

```
    } else if (this.hasError) {
      Column() {
        Text('加载失败，请重试')
          .fontSize(16)
          .fontColor('#566573')
          .margin({ bottom: 20 })
        Button('重试')
          .backgroundColor('#FFD700')
          .fontColor('#2C3E50')
          .onClick(() => this.loadImage())
      }
    } else if (this.pixelMapImg) {
      this.ImageContent()
    } else {
      Button('加载图片')
        .width(160)
        .height(50)
        .backgroundColor('#FFD700')
        .fontColor('#2C3E50')
        .onClick(() => this.loadImage())
    }
  }
  .layoutWeight(1)
  .width('100%')
}
.height('100%')
}
```

