

微信小程序 -- Unit06

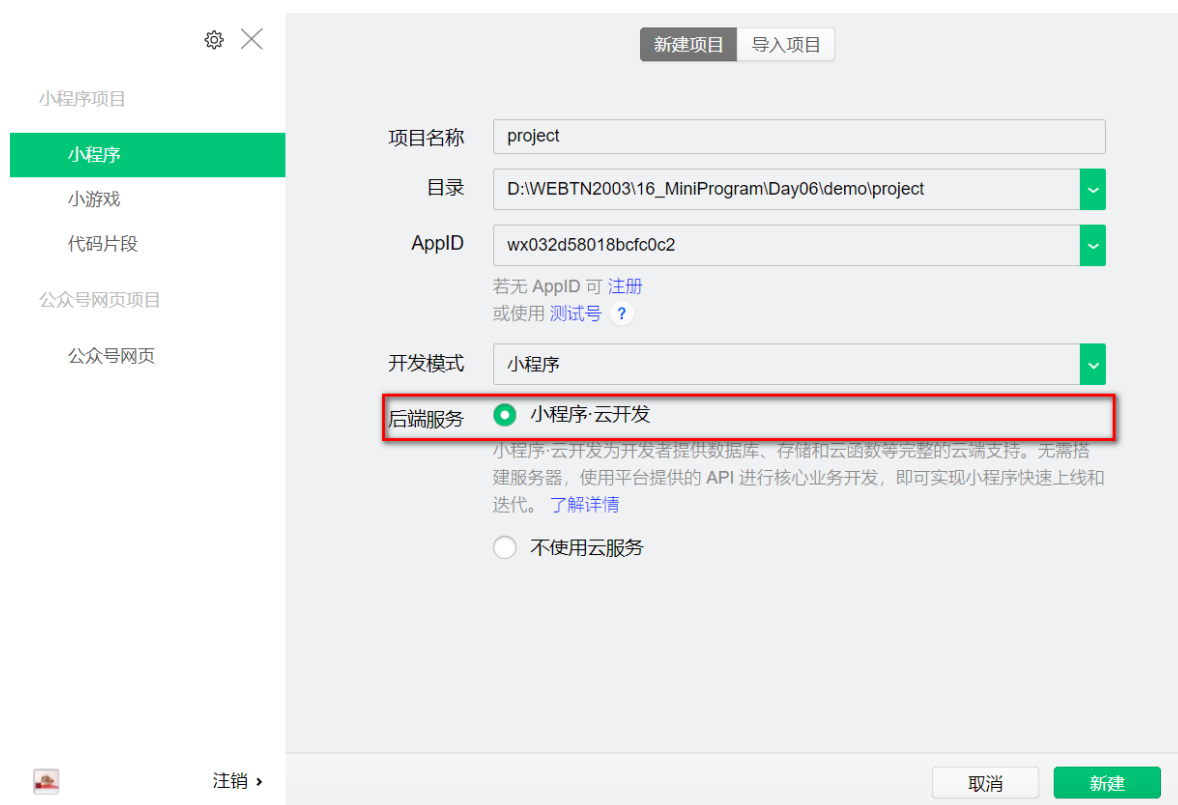
1.微信小程序云开发

1.1 概述

微信小程序云开发的类型有：

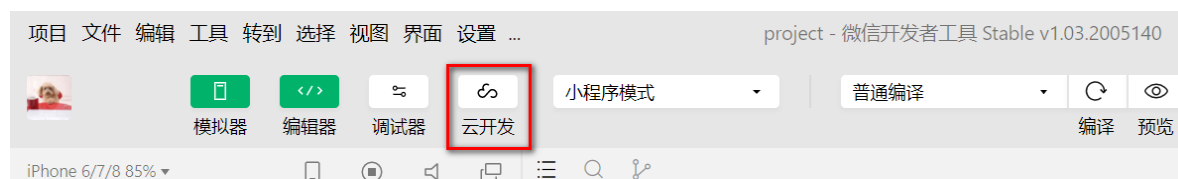
- 云存储，类似于网盘，可以存储任意的图像、音像、视频等文件。
- 云数据库，是一个专门为微信小程序提供的 JSON 类型的数据库。
- 云函数，实现更为复杂的服务器扩展功能。

1.2 创建云开发项目



1.3 开通云服务

A.单击工具栏中的"云开发"按钮



B.

每个微信小程序最多只能开通两个云服务

微信小程序只能使用自己关联的云服务，协议前缀为 `c1oud://`

云服务一旦指定，无法直接进行修改（只能通过填写工单实现）

2.云存储

云存储可想像成网盘，用于存储微信小程序中使用的图像、音频、视频等文件。云存储中的文件管理可以通过两种方式实现管理：

A.通过云开发控制台实现

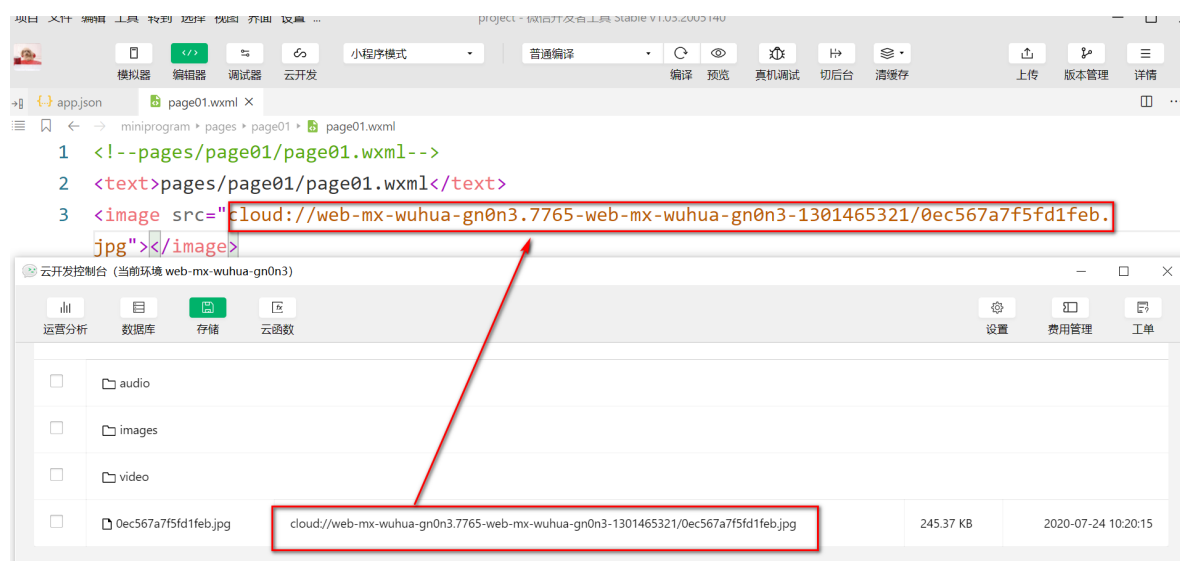
B.通过脚本实现

2.1 通过云开发控制台实现

A. 通过云开发控制台上传文件 -- 如图片文件



B.在 `page.xml` 中书写相应的组件 -- 如 `<image>` 组件，并且修改 `src` 属性值为上传文件的 `file ID`，示例代码如下：



```
<image src="cloud://web-mx-wuhua-gn0n3.7765-web-mx-wuhua-gn0n3-1301465321/0ec567a7f5fd1feb.jpg"></image>
```

2.2 通过脚本实现

• `wx.chooseImage()`

`wx.chooseImage()` 方法用于实现从本地相册选择图片或使用相机拍照，其语法结构是：

```
wx.chooseImage({
  count: 最多选取的图片数量(整数类型, 默认为9),
  //Album, 相册
  //Camera, 相机
  sourceType: 图片源的类型(Array类型, ['album', 'camera']),
  success: (res) => {
    //...
  }
})
```

`success` 回调函数的返回值包括:

`tempFilePaths`, 返回由上传文件临时名称组成的字符串数组, 形如:

```
tempFilePaths: [
  "http://tmp/wx032d58018bcfc0c2.o6zAJs1wk6dVHlKP1MH05VbQ_vUw.VkXLh4NRd8Mv8baad233fb0fc075596bc3ffcaedbe26.jpg",

  "http://tmp/wx032d58018bcfc0c2.o6zAJs1wk6dVHlKP1MH05VbQ_vUw.GwLaZnjRGt8M518c26c29502ac9c279ab2fa9d2b2193.jpg",

  "http://tmp/wx032d58018bcfc0c2.o6zAJs1wk6dVHlKP1MH05VbQ_vUw.4TbnY4FIpkxPcb313f391f36f2de7af69f2.jpg"
]
```

`tempFiles`, 返回由上传文件临时名称组成的对象数组, 形如:

```
tempFiles: [
  {
    path:
      "http://tmp/wx032d58018bcfc0c2.o6zAJs1wk6dVHlKP1MH05VbQ_vUw.VkXLh4NRd8Mv8baad233fb0fc075596bc3ffcaedbe26.jpg",
    size: 99396
  }, {
    path:
      "http://tmp/wx032d58018bcfc0c2.o6zAJs1wk6dVHlKP1MH05VbQ_vUw.GwLaZnjRGt8M518c26c29502ac9c279ab2fa9d2b2193.jpg",
    size: 175410
  }, {
    path:
      "http://tmp/wx032d58018bcfc0c2.o6zAJs1wk6dVHlKP1MH05VbQ_vUw.4TbnY4FIpkxPcb313f391f36f2de7af69f2d64c26828.jpg",
    size: 175410
  }
]
```

• `wx.cloud.uploadFile()`

`wx.cloud.uploadFile()` 方法用于实现将本地资源上传至云存储空间, 其语法结构是:

```
wx.cloud.uploadFile({
  filePath: "要上传文件资源的路径",
  cloudPath: "云存储的路径",
  success: res=>{
    //...
  }
})
```

3. 云数据库

3.1 概述

云数据库是一种既可以在微信小程序的前端，也可以在云函数中读/写数据的 JSON 类型数据库。

微信小程序提供的云数据库底层为 MongoDB -- 非关系型数据库。

3.2 关系型数据库与非关系型数据库的区别

关系型数据库	非关系型数据库												
常用的 DBMS: MySQL、SQL Server、Oracle、DB2	常用的 NoSQL 有: MongoDB、Redis NoSQL: Not Only SQL												
核心概念: 数据库(Database)->数据表(Table)->列 Column (称为字段-Field)->记录-Record (称为行-Row)	数据库->集合(Collection)->字段(Field)->记录(Record)/文档(Document)												
数据类型: 整型、浮点型、字符型、日期时间型等	数据类型: 字符型、数值型、布尔型、数组、对象等												
操作命令: 所有关系型数据库的查询命令都来源于 SQL 语言, 所以大致是相同的。 Microsoft SQL Server 使用 T-SQL Oracle 使用 PL/SQL	操作命令: 不尽相同												
关系的维护: 通过外键约束实现	关系的维护: 不存在关系的概念												
记录的结构: <table><tr><th>id</th><th>username</th><th>password</th><th>c_id</th></tr><tr><td>1</td><td>Tom</td><td>12345</td><td>2</td></tr><tr><td>1</td><td>Rose</td><td>NULL</td><td>2</td></tr></table>	id	username	password	c_id	1	Tom	12345	2	1	Rose	NULL	2	记录的结构: <pre>[{ id:1, username:"Tom", className:"三班" , age:26},{ id:1, username:"Tom", className:"三班" , sex:'男'}, { id:1, username:"Tom" , friends:[AA, 'BB', 'CC']}]</pre>
id	username	password	c_id										
1	Tom	12345	2										
1	Rose	NULL	2										

3.2 通过云开发控制台管理集合及记录



A. 创建集合(创建集合时, 只需要输入合法的名称即可)

B. 根据需要添加记录

C. 如有需要, 还必须设置数据的访问权限

3.3 通过微信小程序客户端管理记录

先行通过云开发控制台创建集合(参见3.2)

• `wx.cloud.database()`

`wx.cloud.database()` 方法用于完成数据库的连接, 语法结构是:

```
Database wx.cloud.database()
```

`wx.cloud.database()` 方法可以理解为 `USE ...`

该方法的返回值为 `Database` 对象(也称为 `Database` 实例)

• 数据库实例

数据库实例是通过 `wx.cloud.database()` 方法获取到的, 示例代码如下:

```
let db = wx.cloud.database();
```

在上述代码中, 变量 `db` 即为数据库实例。

- `collection()` 方法

`collection()` 方法用于获取当前数据库中指定集合的引用, 语法结构是:

```
Collection Database.collection("集合名称")
```

假设当前准备对于 `comments` 集合进行操作, 再结合上述的代码时, 示例代码如下:

```
// 获取数据库对象实例
let db = wx.cloud.database();
// 获取集合对象的实例
let coll = db.collection('comments');
```

•

• 集合对象实例

- `add()` 方法

`add()` 方法用于在集合中插入记录, 其语法结构是:

```
Collection.add({data:{...}})
```

`_id` 是由微信小程序生成的保证记录唯一性的ID

`_openid`，是当前记录的所有者

`add()` 方法的返回值为 `promise` 或 `callback`，包含 `_id` 属性，表示最新插入记录的 ID

- `field()` 方法

`field()` 方法用于指定查询的字段，语法结构是：

```
Collection Collection.field({...})
```

- `where()` 方法

`where()` 方法用于指定集合的查询条件，语法结构是：

```
Collection collection.where()
```

- `get()` 方法

`get()` 方法用于获取集合中的记录，语法结构是：

```
Collection.get()
```

`get()` 方法返回 `Promise` 或 `callback`，其中包含 `data` 的属性(`Array` 类型)，为返回记录所形成的数组，每个数组成员为 `object` 类型

-