



## 02-07 flutter mixins

Posted on 2020-11-23 23:37 肖无情 阅读(77) 评论(0) 编辑 收藏 举报

mixins的中文意思是混入，就是在类中混入其他功能。

Mixins是一种在多个类层次结构中复用类代码的方法

可以看出Mixins最重要的功能是复用代码，我们先看下JAVA，复用代码的方式有哪些：

1. 继承  
子类可以复用父类的方法和属性，但是JAVA里的继承只能单继承。
2. 组合  
将要复用的代码，封装成类A，让其他类持有A的实例，看上去貌似解决了复用代码的问题，但是一方面，每个类持有的A的实例是不同的，有多少个类，就总共有多少个A的实例，而且另一方面，即使A使用单例，使用起来也很不方便。
3. 接口  
定义一个接口interface，类实现interface，这样虽然接口是同一个，但是实现却是分散的，能复用的代码是有限的。  
在维基百科中有对mixins最准确的定义：

在面向对象的语言中,mixins类是一个可以把自己的方法提供给其他类使用,但却不需要成为其他类的父类。  
复制代码

mixins是要通过非继承的方式来复用类中的代码。

这里举个例子，有一个类A，A中有一个方法a()，还有一个方法B，也想使用a()方法，而且不能用继承，那么这时候就需要用到mixins,类A就是mixins类(混入类)，类B就是要被mixins的类，对应的Dart代码如下：

类A mixins 到 B

```
class A {
  String content = 'A Class';

  void a(){
    print("a");
  }
}

class B with A{

}

B b = new B();
print(b.content);
b.a();
复制代码
```

输出是：

```
A Class
a
复制代码
```

将类A mixins 到 B，B可以使用A的属性和方法，B就具备了A的功能，但是需要强调的是：

1. mixins的对象是类
2. mixins绝不是继承，也不是接口，而是一种全新的特性
3. 可以mixins多个类
4. mixins的使用需要满足一定条件

## 0x02 with

mixins要用到的关键字 with

怎么来理解with呢？很简单：

继承 -> extends

mixins -> with

继承和mixins是一样的，是语言的特性，with和extends是关键字。

## 0x03 使用mixins的条件

因为mixins使用的条件，随着Dart版本一直在变，这里讲的是Dart2.1中使用mixins的条件：

1. mixins类只能继承自object
2. mixins类不能有构造函数
3. 一个类可以mixins多个mixins类
4. 可以mixins多个类，不破坏Flutter的单继承

## 0x04 一个类可以mixins多个mixins类

看下面代码：

```
class A {
  void a(){
    print("a");
  }
}

class A1 {
  void a1(){
    print("a1");
  }
}
```

```
}  
}  
  
class B with A,A1{  
  
}  
  
B b = new B();  
b.a();  
b.al();  
复制代码
```

输出是:

```
a  
al  
复制代码
```

但是，如果A和A1的方法相同，而且调换A和A1的顺序，在被mixins的类中实现同一个方法呢，看下面的代码：

```
class A {  
    void a(){  
        print("a");  
    }  
}  
  
class A1 {  
    void a(){  
        print("a1");  
    }  
}  
  
class B with A,A1{  
  
}  
  
class B1 with A1,A{  
  
}  
  
class B2 with A,A1{  
    void a(){  
        print("b2");  
    }  
}  
  
class C {  
    void a(){  
        print("a1");  
    }  
}  
  
class B3 extends C with A,A1{  
  
}  
  
class B4 extends C with A1,A{  
  
}  
  
class B5 extends C with A,A1{  
    void a(){  
        print("b5");  
    }  
}  
  
B b = new B();  
B1 b1 = new B1();  
B2 b2 = new B2();  
B3 b3 = new B3();  
B4 b4 = new B4();  
B5 b5 = new B5();  
  
b.a();  
b1.a();  
b2.a();  
b3.a();  
b4.a();  
b5.a();  
复制代码
```

会是什么样的结果呢？

## 0x05 mixins的实现原理

Mixins in Dart work by creating a new class that layers the implementation of the mixin on top of a superclass—it is not “on the side” but “on top” of the superclass, so there is no ambiguity in how to resolve lookups.

Dart中的mixin通过创建一个新类来工作，该类将mixin的实现通过创建一个新类并分层在超类之上-；它不是“在边上”而是“在上面”超类，因此在如何解析查找方面没有歧义。

mixins不是多继承

Mixins is not a way to get multiple inheritance in the classical sense. Mixins is a way to abstract and reuse a family of operations and state. It is

similar to the reuse you get from extending a class, but it is compatible with single-inheritance because it is linear.

所以输出结果是:

```
a1
a
b2
a1
a
b5
复制代码
```

## 0x06 mixins类型

mixins的实例类型是什么?

很简单, mixins的类型就是其超类的子类型, 所以:

```
B3 b3 = B3();
print(b3 is C);
print(b3 is A);
print(b3 is A1);
复制代码
```

都会为true

## 0x07 on

on关键字, 之前是用于try catch, 用于指定异常的类型。

这次, on只能用于被mixins标记的类, 例如mixins X on A, 意思是要mixins X的话, 得先接口实现或者继承A。这里A可以是类, 也可以是接口, 但是在mixins的时候用法有区别。

想看接口的请看我的这篇文章Flutter Dart语法(1) :extends、implements、with的用法与区别

### 1. on 一个类

用继承:

```
class A {
  void a(){
    print("a");
  }
}

mixin X on A{
  void x(){
    print("x");
  }
}

class mixinsX extends A with X{

}
复制代码
```

### 1. on 的是一个接口:

得首先实现这个接口, 然后再用mix

```
class A {
  void a(){
    print("a");
  }
}

mixin X on A{
  void x(){
    print("x");
  }
}

class implA implements A{
  @override
  void a() {
    // TODO: implement a
  }
}

class mixinsX2 extends implA with X{

}
复制代码
```

分类: flutter

标签: flutter

好文要顶

关注我

收藏该文



肖无情

粉丝 - 2 关注 - 5


+加关注

0

推荐

0

反对

 登录后才能查看或发表评论，立即 [登录](#) 或者 [逛逛](#) [博客园首页](#)

【推荐】阿里云新人特惠，爆款云服务器2核4G低至0.46元/天

编辑推荐:

- [gRPC 入门与实操 \(.NET 篇\)](#)
- [dotnet 代码优化 聊聊逻辑圈复杂度](#)
- [一个棘手的生产问题，但是我写出来之后，就是你的了](#)
- [你可能不知道的容器镜像安全实践](#)
- [.Net 6 使用 Consul 实现服务注册与发现](#)

阅读排行:

- [Redux与前端表格施展“组合拳”，实现大屏展示应用的交互增强](#)
- [gRPC入门与实操\(.NET篇\)](#)
- [博客园主题修改分享 - 过年篇](#)
- [如何优雅地校验后端接口数据，不做前端背锅侠](#)
- [产品与研发相处之道](#)