阿里云 云栖社区

ARouter 介绍与最佳实践

阿里云事业群 刘志龙(正纬)

自我介绍

- 刘志龙(花名 正纬)
- 2015.07 ~ 至今 阿里云 Android开发
- 1.0 ~ 3.16版本

1. 原生的路由方案存在的问题

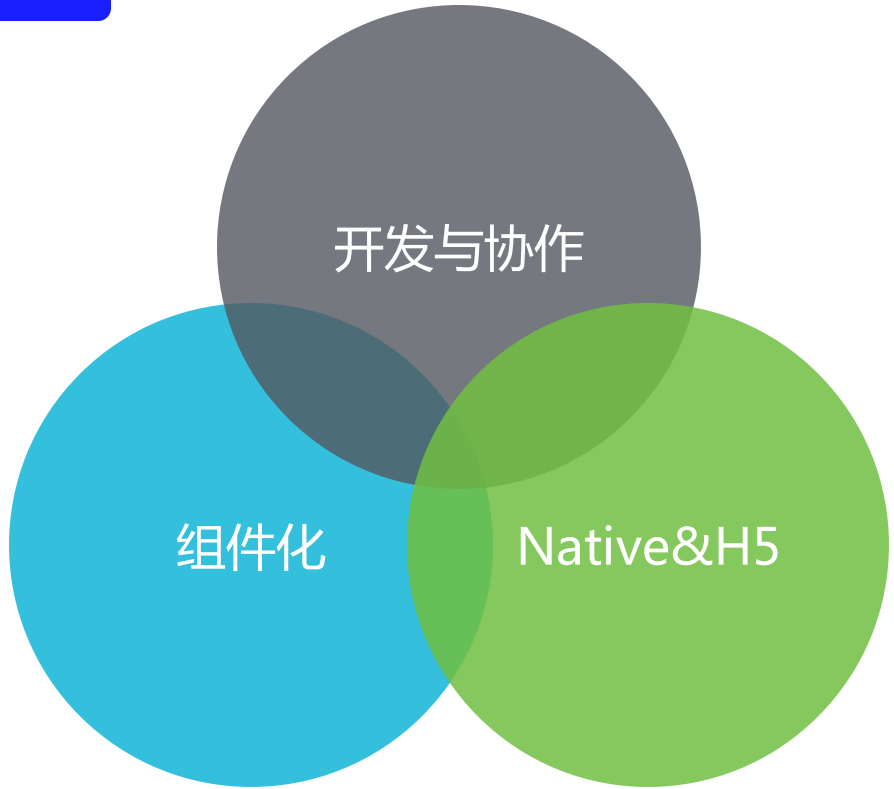
原生的路由方案

- 显式：直接的类依赖，耦合严重
- 隐式：规则集中式管理，协作困难
- Manifest 扩展性较差
- 跳转过程无法控制
- 失败无法降级

自定义的路由组件

- 通过URL索引，无类依赖
- 分布式管理页面配置
- 良好的可扩展性
- AOP支持
- 灵活的降级方式

2. 为什么要用路由组件



3. 路由框架的特点

分发

把一个URL或者请求按照一定的规则分配给一个服务或者页面来处理，这个流程就是分发，分发是路由框架最基本的功能

管理

将组件和页面按照一定的规则管理起来，在分发的时候提供搜索、加载、修改等操作，这部分就是管理，也是路由框架的基础，上层功能都是建立在管理之上

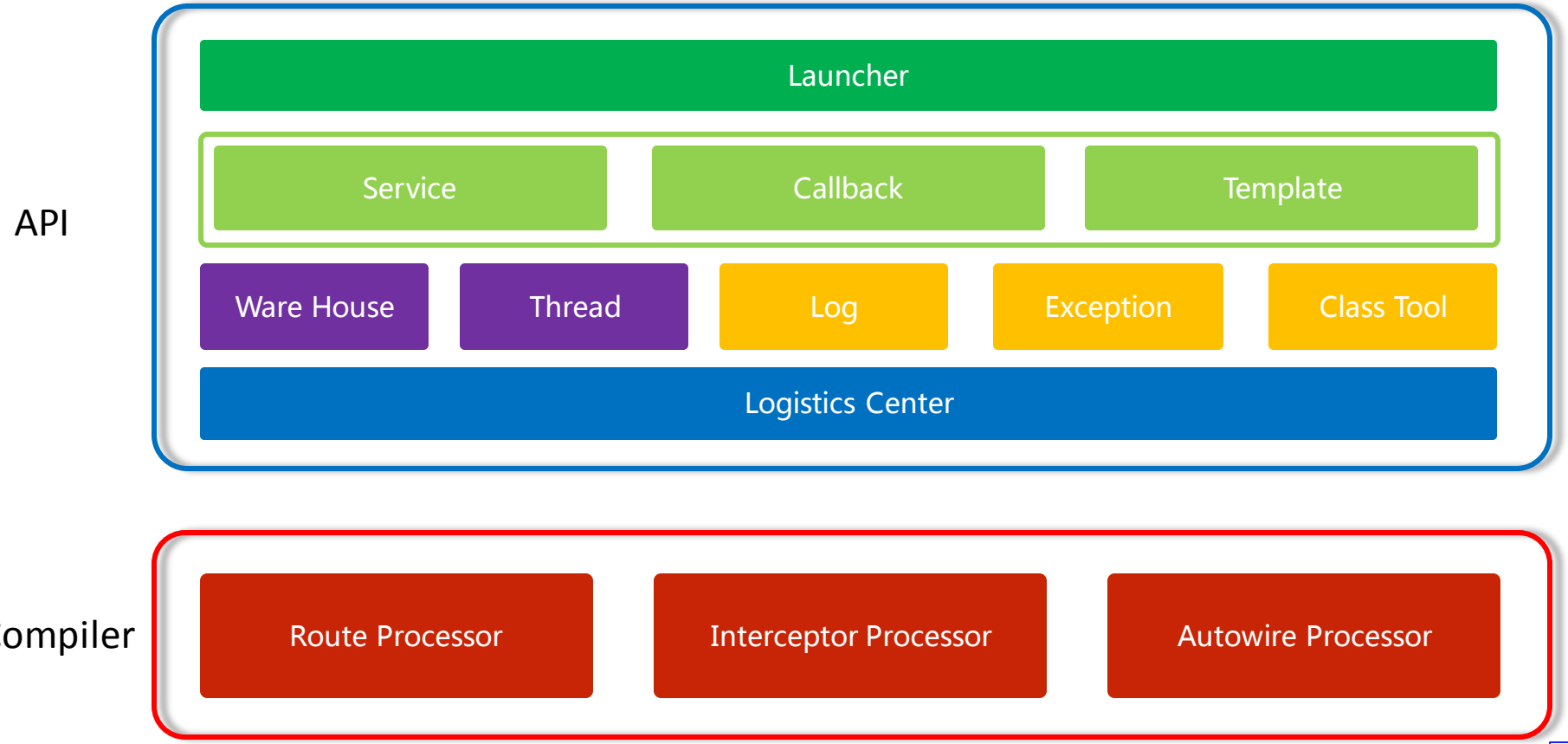
控制

就像路由器一样，路由的过程中，会有限速、屏蔽等一些控制操作，路由框架也需要在路由的过程中，对路由操作做一些定制性的扩展，后期的功能更新，是围绕这个部分来做的

A android router middleware that help app navigating to activities and custom services.

ARouter 是 Android 平台中对页面、服务提供路由功能的中间件，提倡简单且够用。

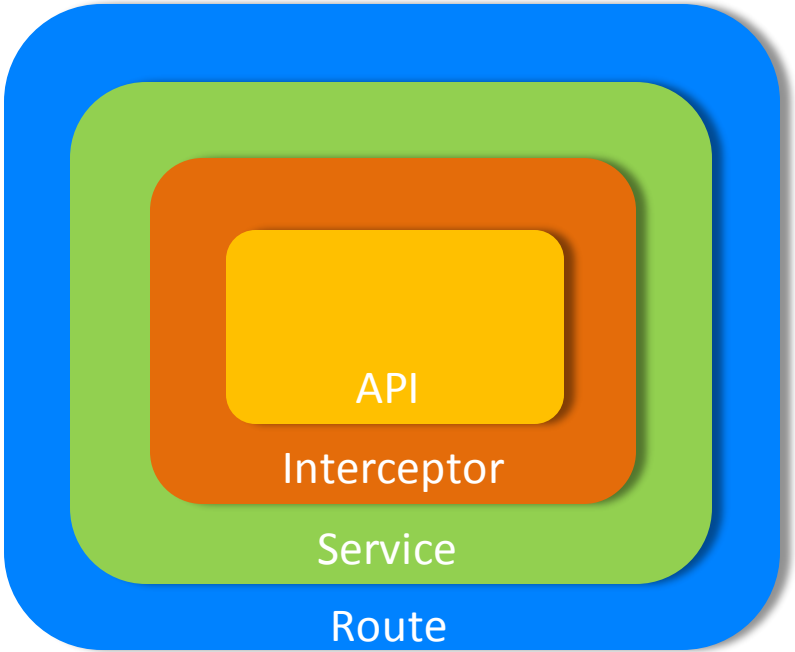

优势一
直接解析URL路由&
解析参数并赋值
优势二
支持多模块项目
优势三
支持InstantRun
优势四
允许自定义拦截器
优势五
提供Ioc容器
优势六
映射关系自动注册
优势七
灵活的降级策略


API
Launcher
Service
Callback
Template
Ware House
Thread
Log
Exception
Class Tool
Logistics Center
Compiler
Route Processor
Interceptor Processor
Autowire Processor

bootstrapping

Extensibility

Simple & enough



页面注册：注解&注解处理器

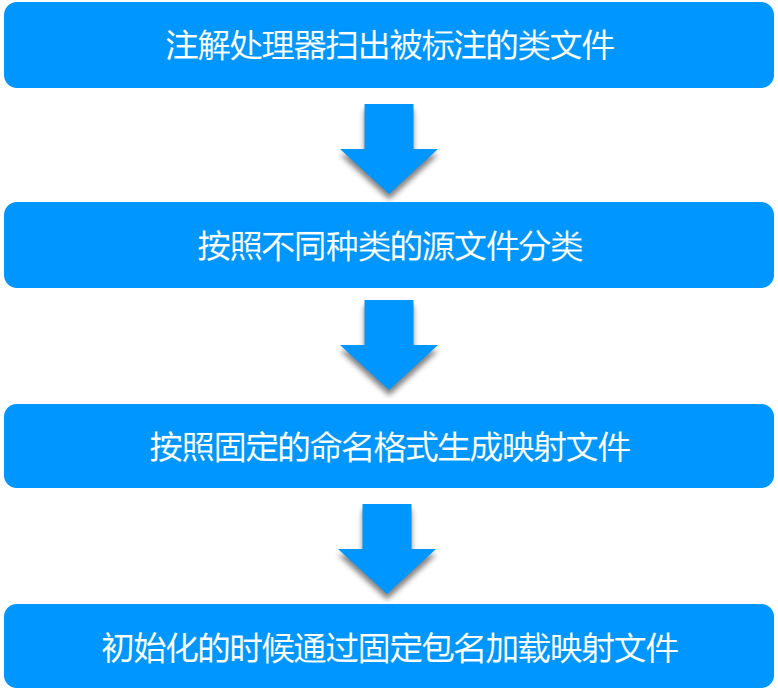
```
@Route(path = "/test/activity1")
public class Test1Activity extends AppCompatActivity {
```

运行期处理注解，大量运用反射

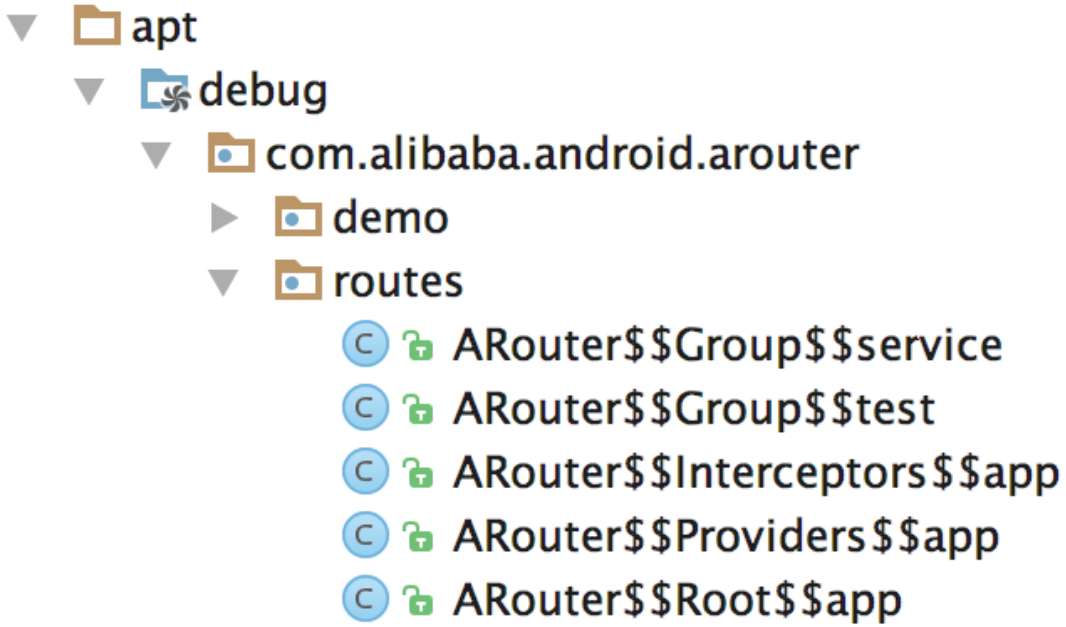
```
/**
 * Mark a page can be route by router.
 *
 * @a
 * @w
 * @s
 */
@Target({ElementType.TYPE})
@Retention(RetentionPolicy.CLASS)
public @interface Route {
```

编译期处理被注解标注的类，不使用反射

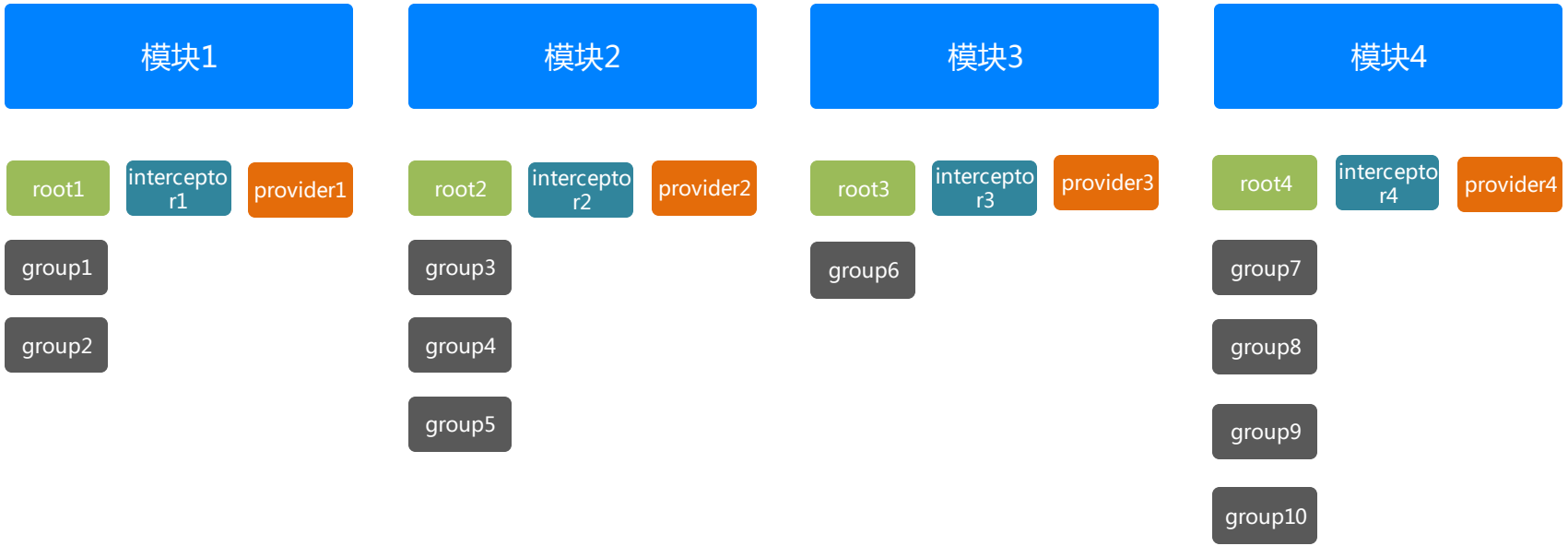
页面注册：注解&注解处理器



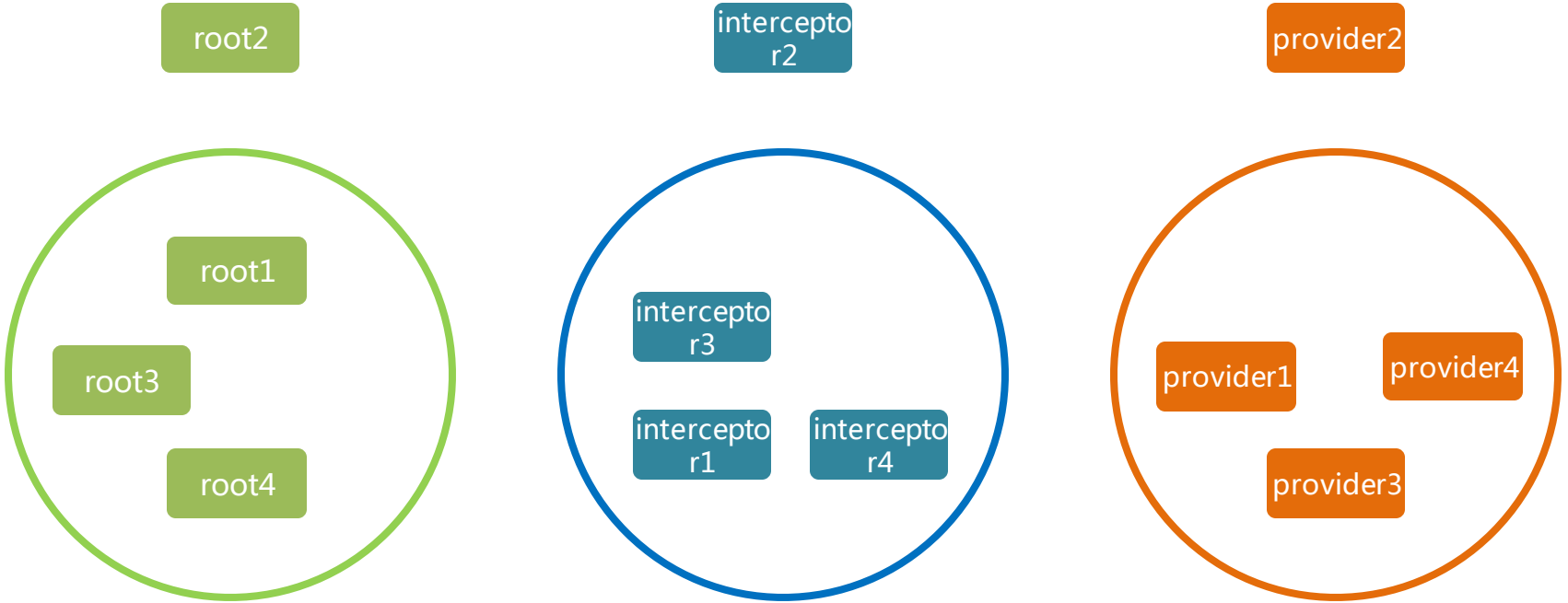
页面注册：注解&注解处理器



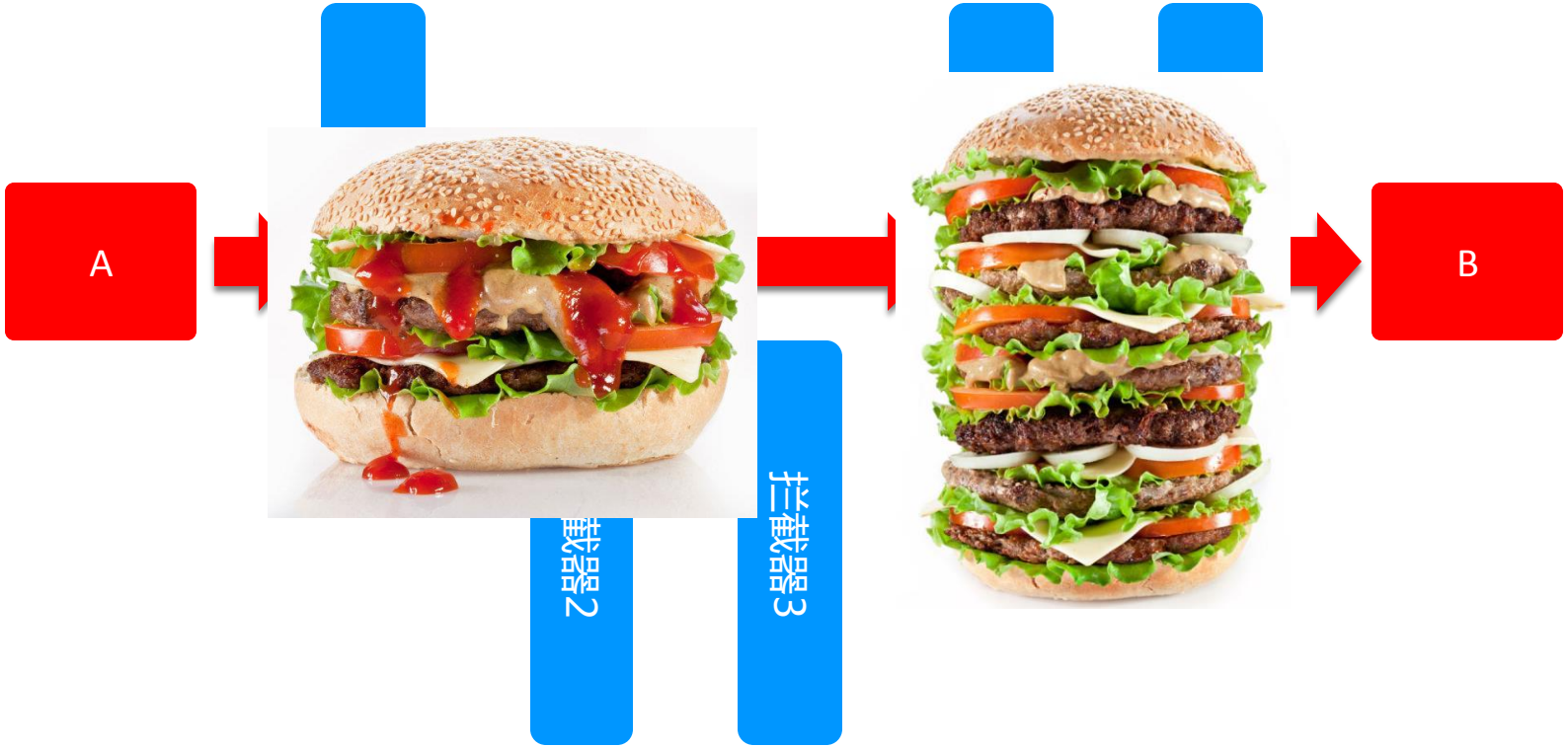
加载：分组管理，按需加载


模块1
root1
interceptor1
provider1
group1
group2
模块2
root2
interceptor2
provider2
group3
group4
group5
模块3
root3
interceptor3
provider3
group6
模块4
root4
interceptor4
provider4
group7
group8
group9
group10

加载：分组管理，按需加载


root2
root1
root3
root4
interceptor2
interceptor3
interceptor1
interceptor4
provider2
provider1
provider4
provider3

拦截器



InstantRun兼容

Gradle Plugin / Android SDK	< 2.3.0	>= 2.3.0
>= 21	getDexFileDirectory	SplitApk
< 21	getDexFileDirectory	getDexFileDirectory

依赖注入的实现

1. 编译期扫出需要自动装配的字段
2. 把自动装配的字段注册在映射文件中
3. 跳转的时候按照预先的配置从URL中提取参数，并按照类型放入Intent中

4. 反射拿到ActivityThread类，调用它的currentActivityTh 方法，拿到当前的ActivityThread实例
5. 反射替换ActivityThread实例中的字段mInstrumentation
6. 覆写Instrumentation的newActivity方法，在Activity实例化的时候，通过反射把Intent预先存好的参数值写入到需要自动装配的字段中

不够稳定

依赖注入的实现

1. 编译期扫出需要自动装配的字段
2. 把自动装配的字段注册在映射文件中
3. 跳转的时候按照预先的配置从URL中提取参数，并按照类型放入Intent中

4. 目标页面在初始化的时候调用ARouter.inject(this)
5. ARouter会查找到编译期为调用方生成的注入辅助类
6. 实例化辅助类之后，调用其中的inject方法完成字段的赋值

依赖注入的实现

```
/**
 * DO NOT EDIT THIS FILE!!! IT WAS GENERATED BY AROUTER. */
public class Test1Activity$$ARouter$$Autowired implements ISyringe {
  @Override
  public void inject(Object target) {
    Test1Activity substitute = (Test1Activity)target;
    substitute.name = substitute.getIntent().getStringExtra("name");
    substitute.age = substitute.getIntent().getIntExtra("age", 0);
    substitute.girl = substitute.getIntent().getBooleanExtra("girl", false);
    substitute.url = substitute.getIntent().getStringExtra("url");
    substitute.helloService = ARouter.getInstance().navigation(HelloService.class);
  }
}
```

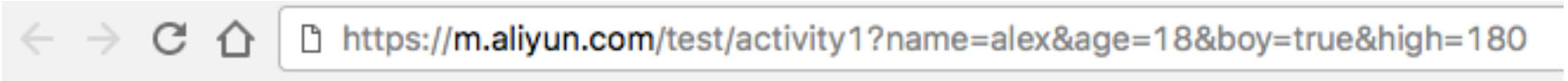
页面跳转

```
@Route(path = "/message/center", name = "消息中心")
public class MessageCenterActivity {

    public static void launch(Activity c) {
        ARouter.getInstance().build("/message/center").navigation(c);
    }
```

```
MessageCenterActivity.launch(this);
```

从外部导航到内部页面


arouter://m.aliyun.com/test/activity1?name=alex&age=18&boy=true&high=180


https://m.aliyun.com/test/activity1?name=alex&age=18&boy=true&high=180

从外部导航到内部页面

```
<!-- Schame -->
<intent-filter>
    <data android:scheme="aliyun"/>

    <action android:name="android.intent.action.VIEW"/>

    <category android:name="android.intent.category.DEFAULT"/>
    <category android:name="android.intent.category.BROWSABLE"/>
</intent-filter>
```

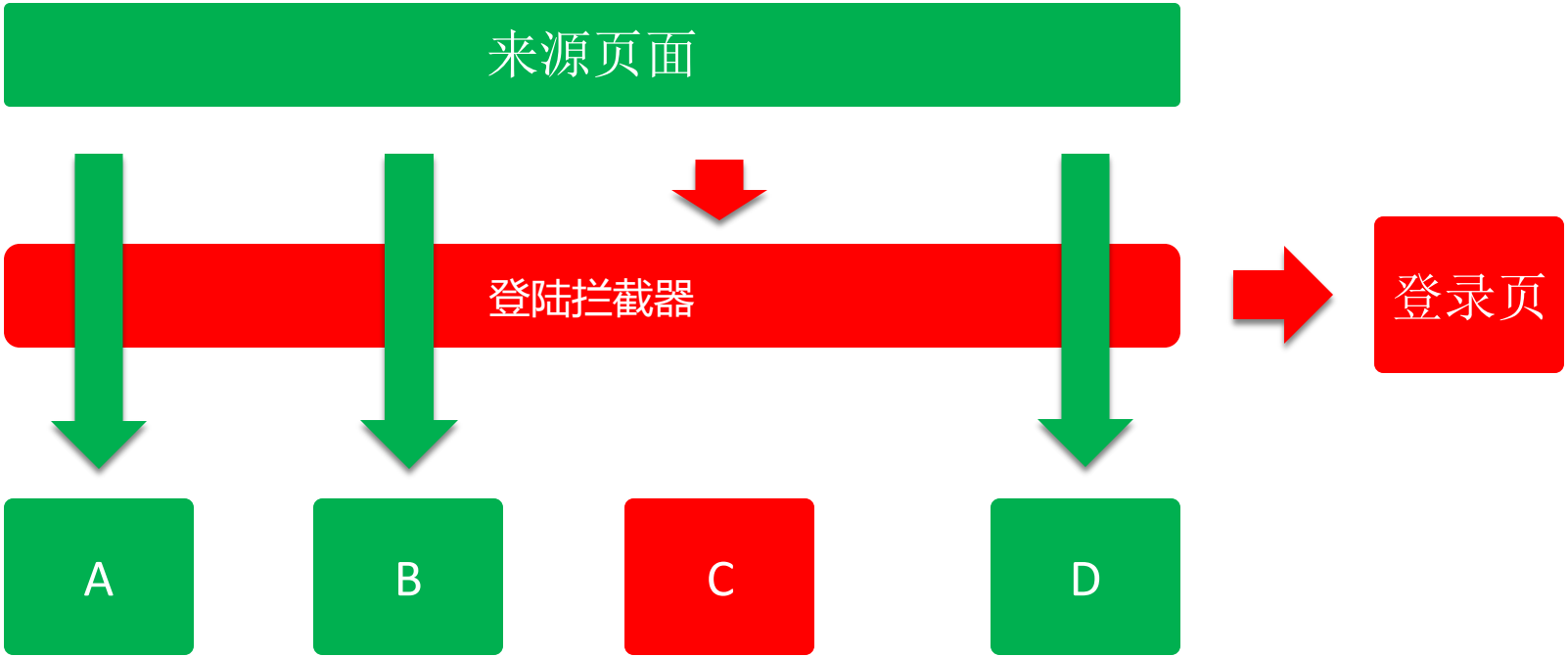
从外部导航到内部页面

```
class BaseActivity extends AppCompatActivity {
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ARouter.getInstance().inject(this);
    }
}
```

从外部导航到内部页面

```
@Route(path = "/test/activity1", extras = -2147483647)
public class Test1Activity extends BaseActivity {

    @Autowired
    String name;

    @Autowired
    int age;

    @Autowired(name = "boy")
    boolean girl;

    private long high;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        Log.d("test", String.format("name=%s, age=%s, girl=%s, high=%s", name, age, girl, high));
    }
}
```

处理登录逻辑：拦截器的运用

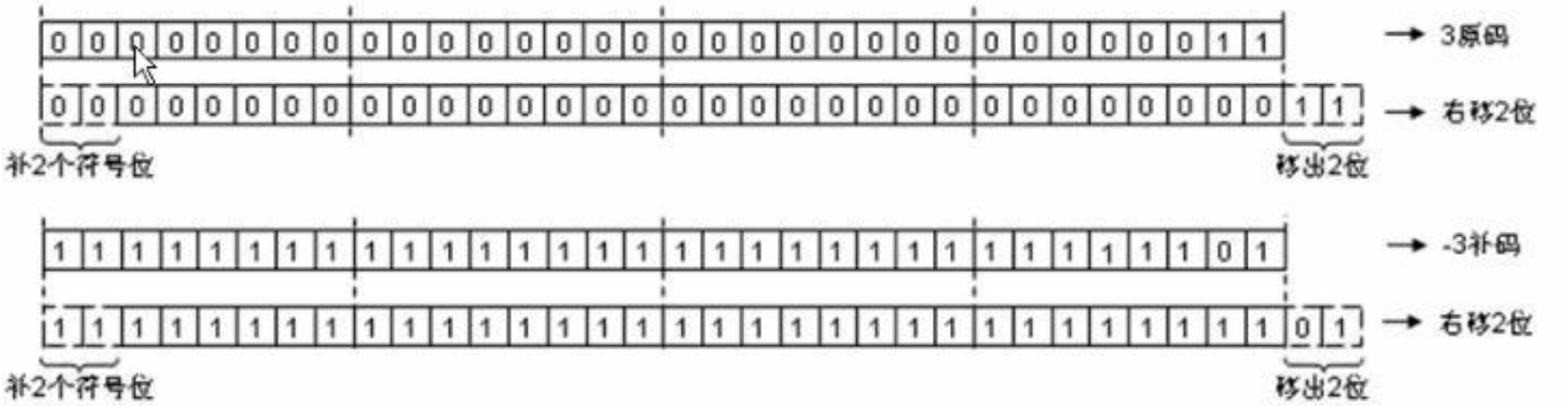


标识目标页面信息：配置extra参数

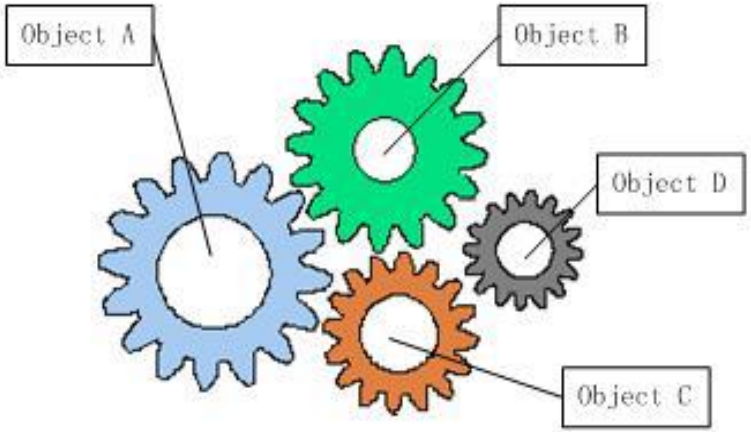
```
@Route(path = "/test/test", name = "测试页面",)
public class TextUtils {

    /**
     * Print thread stack
     */
```

```
priority default -1
extras default -2147483648
group default ""
Did you know that Quick Definition View (⌥Spa
```

标识目标页面信息 : 配置extra参数



模块间通信解耦 ：控制反转



模块间通信解耦 ：控制反转

```java
public interface HelloService extends IProvider {
    void sayHello(String name);
}
```

```java
@Route(path = "/service/hello")
public class HelloServiceImpl implements HelloService {
    Context mContext;

    @Override
    public void sayHello(String name) {
        Toast.makeText(mContext, "Hello " + name, Toast.LENGTH_SHORT).show();
    }

    /**
     * Do your init work in this method, it well be call when processor has been load.
     *
     * @param context ctx
     */
    @Override
    public void init(Context context) {
        mContext = context;
        Log.e("testService", HelloService.class.getName() + " has init.");
    }
}
```

模块间通信解耦 ：控制反转

```
public class Test1Activity extends BaseActivity {

    @Autowired
    HelloService helloService;

    @Autowired(required = true)
    HelloService helloService2;

    @Autowired(name = "/service/hello")
    HelloService helloService3;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        helloService.sayHello("mike");
        helloService2.sayHello("mike");
        helloService3.sayHello("mike");
    }
}
```

模块间通信解耦 ：控制反转

```
case R.id.navByName:
    ((HelloService) ARouter.getInstance().build("/service/hello").navigation()).sayHello("mike");
    break;
case R.id.navByType:
    ARouter.getInstance().navigation(HelloService.class).sayHello("mike");
    break;
```

解决运行期动态修改路由的问题：

```
@Route(path = "/sdk/service/replace")
public class PathReplaceServiceImpl implements PathReplaceService {
    @Override
    public void init(Context context) {

    }

    @Override
    public String forString(String path) {
        return path;
    }

    @Override
    public Uri forUri(Uri uri) {
        return uri;
    }
}
```

解决降级问题：

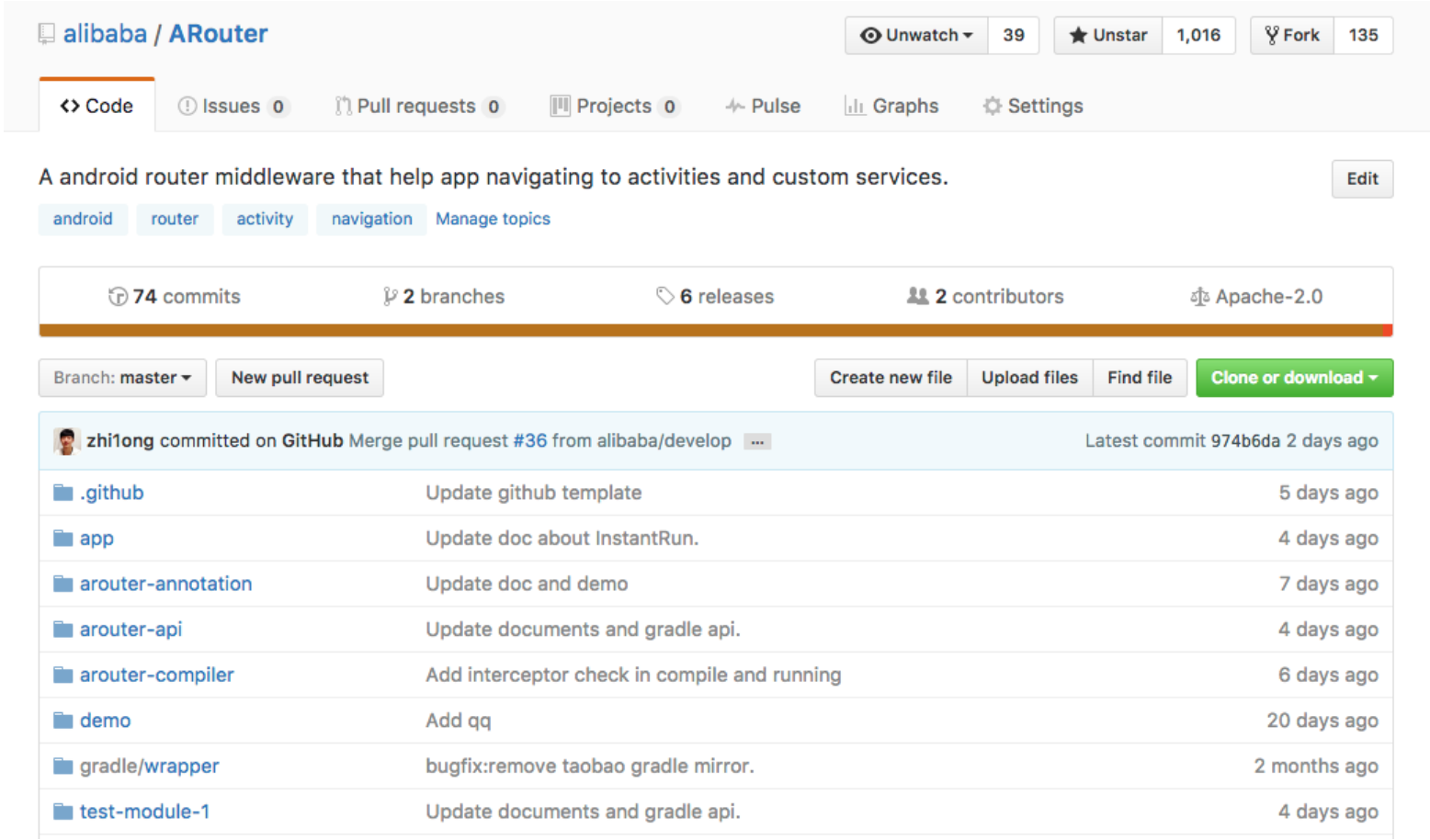
```
@Route(path = "/sdk/service/degrade")
public class DegradeServiceImpl implements DegradeService {
    @Override
    public void onLost(Context context, Postcard postcard) {
        ARouter.getInstance()
                .build("/h5/webview")
                .withString("url", "https://m.aliyun.com/error?path=" + postcard.getPath())
                .navigation();
    }

    @Override
    public void init(Context context) {

    }
}
```

插件化的支持

现在使用的方案需要去dex文件中拿到指定包名下的类，但是插件化之后，很多插件化方案都会对dex进行操作，导致dex位置变动，常规方案无法找到dex文件，不能实现映射文件的初始化

生成映射关系文档

目前映射关系的保存比较复杂，不同类型的映射文件很多，后续的版本中会对这部分进行简化，并添加版本控制，解决后续多版本的兼容性问题



alibaba / ARouter

Unwatch 39 | Unstar 1,016 | Fork 135

Code | Issues 0 | Pull requests 0 | Projects 0 | Pulse | Graphs | Settings

A android router middleware that help app navigating to activities and custom services.

Edit

android router activity navigation Manage topics

74 commits | 2 branches | 6 releases | 2 contributors | Apache-2.0

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

zhi1ong committed on GitHub Merge pull request #36 from alibaba/develop ... Latest commit 974b6da 2 days ago

.github	Update github template	5 days ago
app	Update doc about InstantRun.	4 days ago
arouter-annotation	Update doc and demo	7 days ago
arouter-api	Update documents and gradle api.	4 days ago
arouter-compiler	Add interceptor check in compile and running	6 days ago
demo	Add qq	20 days ago
gradle/wrapper	bugfix:remove taobao gradle mirror.	2 months ago
test-module-1	Update documents and gradle api.	4 days ago

Github : https://github.com/alibaba/ARouter

欢迎大家一起贡献代码~~

感谢各位观众