Kotlin 实现 DSL

何梁伟

极客时间

重拾极客精神·提升技术认知

每天10分钟，邀请顶级技术专家，为你传道授业解惑。

扫一扫，试读专栏

主办方 Geekbang 极客邦科技 InfoQ

ArchSummit

全球架构师峰会 2017

12月8-9日 北京 · 国际会议中心

APSEC 2017

南京大学 NANJING UNIVERSITY × Geekbang 极客邦科技 | InfoQ

AiCon

全球人工智能技术大会 2018

助力人工智能落地

2018.1.13 - 1.14　北京国际会议中心

扫描关注大会官网

何梁伟

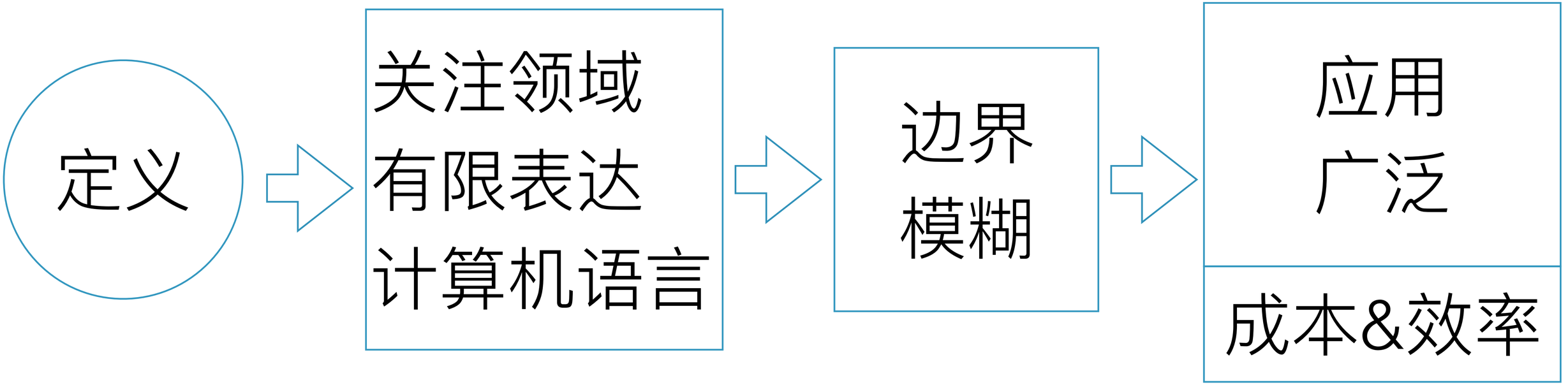
沪江

Android

Kotlin

KotlinThree

DSL 定义



两大分支

外部DSL

内部DSL

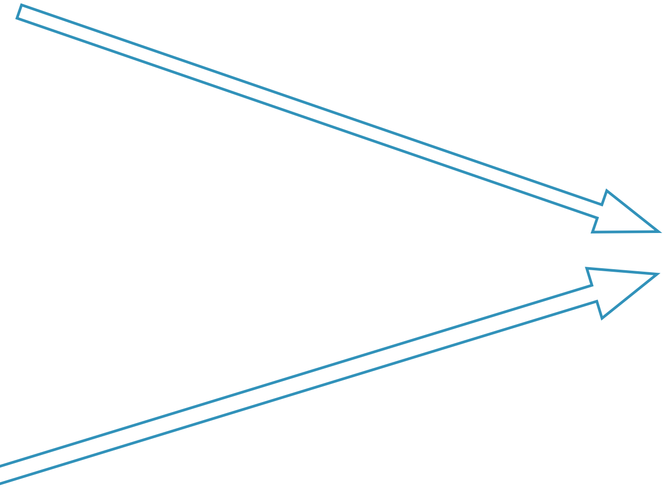
Kotlin 特性支持

```
html {
     head {
          title { +"XML encoding with Kotlin" }
      }
     body {
          h1 { +"XML encoding with Kotlin" }
     }
}
```

```
<html>
  <head>
    <title>
      XML encoding…
    </title>
  </head>
  <body>
    <h1>
      XML encoding…
    </h1>
  </body>
</html>
```

函数类型

```
fun <T> max(collection: Collection<T>,
    less: (T, T) -> Boolean): T? {
    ...
}
```



函数类型

```
val sum: (Int, Int) -> Int = { x, y -> x + y }
```

闭包

```
public inline fun <T, R> Iterable<T>.map(transform: (T) -> R): List<R> {
    return mapTo(ArrayList<R>(collectionSizeOrDefault(10)), transform)
}

...
```

函数类型

```
arrays.map ({ it.length })

arrays.map() { it.length }

arrays.map { it.length }
```

闭包

嵌套闭包

```
html {
    head {
        title { +"XML encoding with Kotlin" }
    }
    body {
        h1 { +"XML encoding with Kotlin" }
    }
}
```

1

2

3

凹造型

```
fun html(block: () -> Unit){
    block()
}

fun head(block: () -> Unit){
    block()
}

fun body(block: () -> Unit){
    block()
}
```

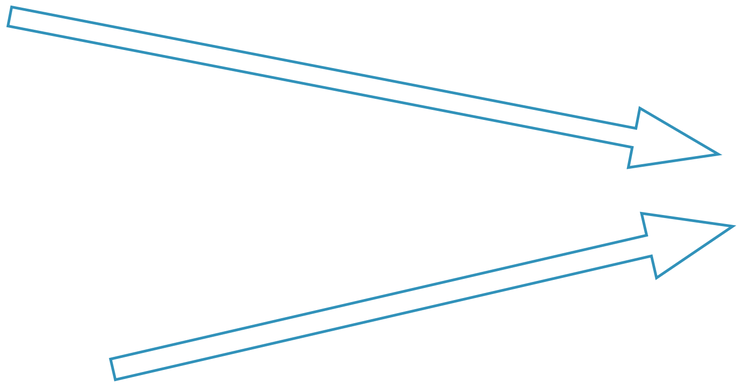
```
html {
    head {

    }
    body {

    }
}
```

扩展

```
fun main(args: Array(String)) {
    val a = 2
    println("Is a positive: ${a.isPositive()}")
}

fun Int.isPositive() = this > 0
```

函数接受者

```
val sum: Int.(Int) -> Int = { other ->
    this + other
}
1.sum(2)
```



带接受者的函数类型

```
fun html(init: Html.() -> Unit): Html{
    val h = Html()
    h.init()
    return h
}
```

填充内容

```
fun html(init: Html.() -> Unit): Html{
    val h = Html()
    h.init()
    return h
}

fun Html.head(init: Head.() -> Unit): Head{
    val h = Head()
    h.init()
    childs.add(h)
    return h
}
```

填充内容

```
html {
     head {   }
}
```

```
html
----head
```

```
<Html>
<Head>


</Head>
</Html>
```

操作符重载

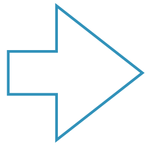
```
html {
    head {
        title { +"XML encoding with Kotlin" }
    }
}
```

```
class Title() : Tag("title"){
    operator fun String.unaryPlus() {
        ...
    }
}

fun title(init: Title.() -> Unit){
    ...
}
```

```
<html>
  <head>
    <title>
      XML encoding…
    </title>
  </head>
</html>
```

DSLMaker

```
html {
    head {
        // should be forbidden
        head {}
    }
}
```



```
@DslMarker
annotation class HtmlTagMarker

@HtmlTagMarker
abstract class Tag(val name: String)

class HTML() : Tag("html") { ... }
class Head() : Tag("head") { ... }
```

```
html {
    head {
        // compile error
        head { }
    }
}
```

小结

- 函数类型
- 闭包
- 扩展
- 函数接受者
- 操作符重载
- DSLMaker

DSL 应用

- Anko
- Gradle
- kotlinx.html
- Kolley

Anko

```
val act = this
val layout = LinearLayout(act)
layout.orientation = LinearLayout.VERTICAL
val name = EditText(act)
val button = Button(act)
button.text = "Say Hello"
button.setOnClickListener {
    Toast.makeText(act, "Hello, ${name.text}!", Toast.LENGTH_SHORT).show()
}
layout.addView(name)
layout.addView(button)
```

```
verticalLayout {
    val name = editText()
    button("Say Hello") {
        onClick { toast("Hello, ${name.text}!") }
    }
}
```

Anko

```
val values = ContentValues()
values.put("id", 5)
values.put("name", "John Smith")
values.put("email", "user@domain.org")
db.insert("User", null, values)
```

```
db.insert("User",
    "id" to 42,
    "name" to "John",
    "email" to "user@domain.org"
)
```

Gradle

Grade 3.0

V1.0m

```
buildscript {
    dependencies {
        classpath("com.android.tools.build:gradle:2.3.1
        classpath(kotlinModule("gradle-plugin"))
    }
    ...
}
apply {
    plugin("com.android.application")
}
android {
    ...
    buildTypes {
        getByName("release") {
            isMinifyEnabled = false
        }
    }
}
```

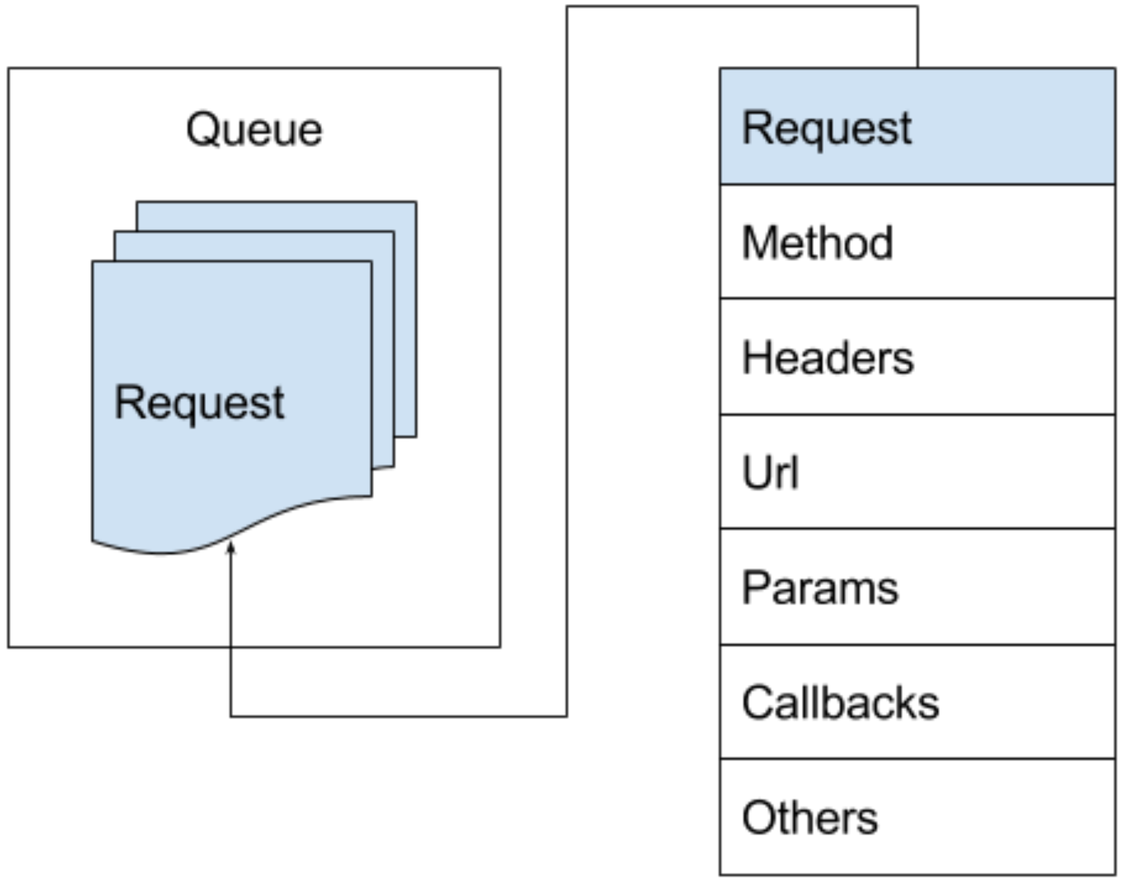
kotlinx.html

```
window.setInterval({
    val myDiv = document.create.div("panel") {
        p {
            +"Here is "
            a("http://kotlinlang.org") { +"official Kotlin site" }
        }
    }

    document.getElementById("container")!!.appendChild(myDiv)

    document.getElementById("container")!!.append {
        div {
            +"added it"
        }
    }
}, 1000L)
```

Kotlin 实现 DSL

Kolley 去哪了?

网络请求


Queue
Request
Request
Method
Headers
Url
Params
Callbacks
Others

Retrofit

```
public interface GitHubService {
  @GET("users/{user}/repos")
  Call<List<Repo>> listRepos(@Path("user") String user);
}

Retrofit retrofit = new Retrofit.Builder()
    .baseUrl("https://api.github.com/")
    .build();

GitHubService service = retrofit.create(GitHubService.class);
Call<List<Repo>> repos = service.listRepos("octocat");
call.enqueue(new Callback<List<Repo>>() {
    @Override
    public void onResponse(Call<List<Repo>> call, Response<List<Repo>> response) {
    }

    @Override
    public void onFailure(Call<MovieEntity> call, Throwable t) {
    }
});
```

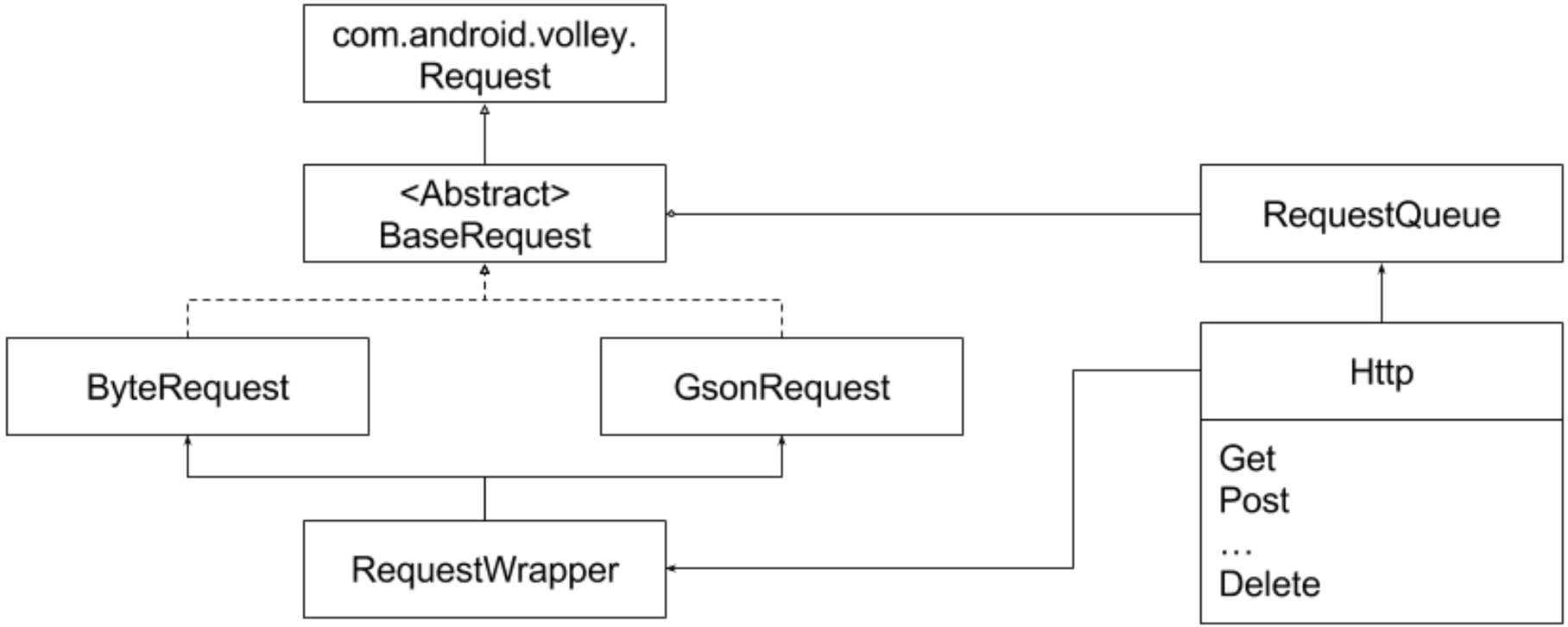
Volley

```
String url ="http://www.google.com";

StringRequest stringRequest = new StringRequest(Request.Method.GET, url
            new Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        mTextView.setText("Response is: "+ response.substring(0,500));
    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        mTextView.setText("That didn't work!");
    }
});

queue.add(stringRequest);
```

Kolley

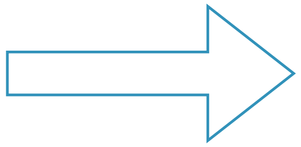
```
Http.get {
    url = "http://api.openweathermap.org/data/2.5/weather"

    params {
        "q" - "shanghai"
        "appid" - "d7a98cf22463b1c0c3df4adfe5abbc77"
    }

    onSuccess { bytes ->
        log("on success ${bytes.toString(Charset.defaultCharset())}")
    }

    onFail { error ->
        log("on fail ${error.toString()}")
    }
}
```

Kolley内部关系图



Kolley

```
val request: (Int, RequestWrapper.() -> Unit) -> Request<ByteArray> =
{ method, init ->
    val baseRequest =RequestWrapper()
    baseRequest.method = method
    baseRequest.init() // 执行闭包
    baseRequest.excute() // 添加到队列执行
    baseRequest._request
}
```

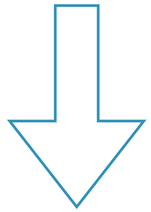


```
Http.get {

}
```

```
val get = request.partially1(Request.Method.GET)
val post = request.partially1(Request.Method.POST)
...
val patch = request.partially1(Request.Method.PATCH)
```

Kolley

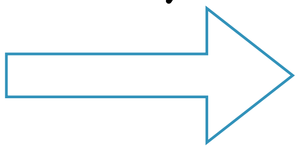
```
var method: Int = Request.Method.GET
var url: String = ""
var raw: String? = null // used for a POST or PUT request.
var tag: Any? = null
```



```
Http.get {
    url = "http://api.openweathermap.org/data/2.5/weather"
    tag = this@MainActivity
}
```

Kolley

```
val pairs = fun (map: MutableMap<String,String>,
                 makePairs: RequestPairs.() -> Unit){
    val requestPair = RequestPairs()
    requestPair.makePairs()
    map.putAll(requestPair.pairs)
}
val params = pairs.partially1(_params)
val headers = pairs.partially1(_headers)
```

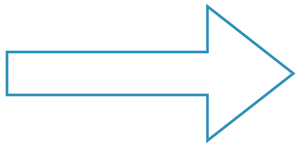


```
params {
    "q" - "shanghai"
    "appid" - "d7a98cXXX"
}
```

```
class RequestPairs {
    var pairs: MutableMap<String, String> = HashMap()
    operator fun String.minus(value: String) {
        pairs.put(this, value)
    }
}
```

Kolley

```
private var _success: (ByteArray) -> Unit = {}
private var _fail: (VolleyError) -> Unit = {}

...

fun onFail(onError: (VolleyError) -> Unit) {
    _fail = onError
}

fun onSuccess(onSuccess: (ByteArray) -> Unit) {
    _success = onSuccess
}
```



```
onSuccess { bytes ->
    log("on success")
}

onFail { error ->
    log("on fail")
}
```

Kolley

```
Http.get {
    url = "http://api.openweathermap.org/data/2.5/weather"

    params {
        "q" - "shanghai"
        "appid" - "d7a98cf22463b1c0c3df4adfe5abbc77"
    }

    onSuccess { bytes ->
        log("on success ${bytes.toString(Charset.defaultCharset())}")
    }

    onFail { error ->
        log("on fail ${error.toString()}")
    }
}
```

关注QCon微信公众号
获得更多干货!

Thanks!

INTERNATIONAL SOFTWARE DEVELOPMENT CONFERENCE

主办方：Geekbang. 极客邦科技 InfoQ