

# Scala with Wechaty

## Scala 实现 Wechaty 小节

报告人: 蔡君

jcai@ganshane.com

<https://github.com/jcai>

June 13, 2020



# 目录

1 自我介绍

2 Scala

3 Scala-Wechaty

4 吐槽

5 期望和建议

6 未来开展工作



# 目录

## 1 自我介绍

## 2 Scala

## 3 Scala-Wechaty

## 4 吐槽

## 5 期望和建议

## 6 未来开展工作



# 程序员老司机



- 15 年 + 全职软件开发经验
  - 99 年接触计算机;
  - 从 Pascal、C 到 Scala、Go;
  - 丰富 J2EE 开发经验;
  - 丰富算法开发经验, 早前致力于警用指纹识别技术;
  - 擅长 C/C++,Java,Scala,Ruby,Go



# 程序员老司机



- 15 年 + 全职软件开发经验
  - 99 年接触计算机;
  - 从 Pascal、C 到 Scala、Go;
  - 丰富 J2EE 开发经验;
  - 丰富算法开发经验, 早前致力于警用指纹识别技术;
  - 擅长 C/C++,Java,Scala,Ruby,Go
- 目前工作
  - 军工企业创始人;
  - 生产制造型企业;
  - 聚焦军民融合战略资源。



# 目录

1 自我介绍

2 Scala

3 Scala-Wechaty

4 吐槽

5 期望和建议

6 未来开展工作



# Scala 简介

Scala 很棒, 真心安利大家!

## 函数式编程和面向对象编程

```
1 trait Cat
2 trait Bird
3 trait Catch
4 trait FullTummy
5 def catch(hunter: Cat, prey: Bird): Cat with Catch
6 def eat(consumer: Cat with Catch): Cat with FullTummy
7 val story = (catch _) andThen (eat _)
8 story(new Cat, new Bird)
```



# Scala 简介

兼容 Java 语法，同时胜过 Java 语法！

摘自 Spring-JDBC

```
1 public interface JdbcTemplate {  
2     List query(PreparedStatementCreator psc,  
3         RowMapper rowMapper)  
4     ...  
5 }  
6 public interface PreparedStatementCreator {  
7     PreparedStatement createPreparedStatement(Connection con) throws  
8         SQLException;  
9 }  
10 public interface RowMapper {  
11     Object mapRow(ResultSet rs, int rowNum)  
12     throws SQLException;
```





# Scala 简介

兼容 Java 语法，同时胜过 Java 语法!

## Scala 的等同语法

```
1 trait JdbcTemplate {
2   def query[ResultItem](psc: Connection => PreparedStatement,
3     rowMapper: (ResultSet, Int) => ResultItem): List[ResultItem]
4 }
```



# Scala 简介

## 静态类型

Table: 变量定义对比

变量类型	C	Java	Scala
可变变量	int x	int x	var x:Int
不可变变量	const int x	final int x	val x:Int
延迟变量	N/A	N/A	lazy val x:Int



# Scala 简介

## 类型推导

```

1  val x: Int = 5
2  val y = 5
3
4  def myMethod(functionLiteral: A => B): Unit
5  myMethod({ arg: A => new B })
6  myMethod({ arg => new B })
    
```



# Scala 简介

## 更加简洁的语法

```
1 def qsort[T <% Ordered[T]](list:List[T]):List[T] = list match {  
2   case Nil => Nil  
3   case x :: xs =>  
4     val (before, after) = xs.partition ( _ < x )  
5     qsort(before) ++ (x :: qsort(after))  
6 }
```



# Scala 简介

## 隐式转换

```

1  def onMessage(listener:Consumer[Message]):Wechaty={
2      EventEmitter.addListener[EventMessagePayload](PuppetEventName.
        MESSAGE,listener)
3      this
4  }
5
6  private[wechaty] implicit def toMessage(messageListener: Consumer[Message
    ])(implicit puppet: PuppetResolver): EventMessagePayload => Unit = {
7      messagePayload: EventMessagePayload => { messageListener.accept(new
        Message(messagePayload.messageId)) }
8  }
    
```



# Scala 简介

完全兼容 Java 库, 所有 Java 类库都能在 Scala 中使用。



# 我如何接触上 scala 的?

通过写 Scala 编译器插件，实现 SQL/HQL/JPQL 编译时刻校验！

```
1  val q3 = select[ModelA] where ModelA.name === "cctv" and (  
2      ModelA.seq === 1 or ModelA.name === "cctv" or ModelA.name[String].  
        isNull  
3      or ModelA.seq[Int] > 1 or ModelA.name === "asdf"  
4  ) limit 3 offset 0 orderBy ModelA.name[String].desc  
5  
6  
7  Assert.assertEquals(1, q3.size)
```



# 目录

1 自我介绍

2 Scala

3 Scala-Wechaty

4 吐槽

5 期望和建议

6 未来开展工作





# 架构

- wechaty => wechaty 的上层调用接口;
- wechaty-puppet => wechaty 和 GRPC 交互中间定义层;
- wechaty-hostie=> 和 Grpc 底层交互。



# 已完成工作

- 与后端 GRPC 稳定交互，多日持续稳定运行;
- 实现所有和后端 GRPC 交互的代码功能;
- 实现所有 Schema 定义;
- 前端实现了 Contact 和 Message 部分功能;



# 已完成工作

使用 Grpcmock(<https://github.com/Fadelis/grpcmock>  
<https://github.com/tomakehurst/wiremock>) 实现了单元测试。

```
1  val contactPayloadResponse1 = ContactPayloadResponse.newBuilder()
2      .setName("jcai")
3      .build()
4  stubFor(unaryMethod(PuppetGrpc.getContactPayloadMethod)
5      .willReturn(contactPayloadResponse1))
6
7  val contact = new Contact("contactId")
8  Assertions.assertEquals("jcai", contact.name)
```



# 目录

1 自我介绍

2 Scala

3 Scala-Wechaty

4 吐槽

5 期望和建议

6 未来开展工作



# 文档和测试太少

- 没有地方说明如何实现？可能我没找到。
- token 满天飞！
- 单元测试太少！
- 单元测试太少！
- 单元测试太少！



# 目录

1 自我介绍

2 Scala

3 Scala-Wechaty

4 吐槽

5 期望和建议

6 未来开展工作



# 期望和建议

- 获得更多后端 GRPC 细节;
- 精简 token 的使用说明;
- 增强 ts 版本的单元测试;
- 实现基于 Atom、Electron 的 UI, 后端能够支持各类语言的实现, 方便展示和测试;
- Schema 使用中性语言描述 (考虑 antlr);
- puppet 和 hostie 大部分代码能够根据 schema 能够自动生成!



# 目录

1 自我介绍

2 Scala

3 Scala-Wechaty

4 吐槽

5 期望和建议

6 未来开展工作





# Future Work

- 完善剩余 wechaty 功能;
- 让更多人参与进来;
- 更加全面的单元测试;
- 实现一个轻量级应用 (优惠券助手);



# Thank you

## Thank you for listening!



# Q&A

## Questions?

<https://github.com/jcai/wechaty-meetup>

