

SimHei

Microsoft YaHei Microsoft YaHei

Scala with Wechaty

ScalaWechaty

:

jcai@ganshane.com

<https://github.com/jcai>

June 13, 2020









■ 15+

- 99;
- PascalCScalaGo;
- J2EE;
- ;
- C/C++,Java,Scala,Ruby,Go





- 15+
 - 99;
 - PascalCScalaGo;
 - J2EE;
 - ;
 - C/C++,Java,Scala,Ruby,Go
- - ;
 - ;
 -





Scala

Scala,

```
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

JavaJava!

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 SQLException;  
8 }  
9 public interface RowMapper {  
10     Object mapRow(ResultSet rs, int rowNum)  
11     throws SQLException;
```



Scala

JavaJava!

Scala

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



Scala

Table:

	C	Java	Scala
	int x	int x	var x:Int
	const int x	final int x	val x:Int
	N/A	N/A	lazy val x:Int



Scala

```
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

```
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

```
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
7      puppet: PuppetResolver): EventMessagePayload => Unit = {
8      messagePayload: EventMessagePayload => { messageListener.accept(new Message(
9          messagePayload.messageId)) }
10 }
```



Scala

Java,JavaScala



ScalaSQL/HQL/JPQL

```
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)
```





- wechaty =_ wechaty
- wechaty-puppet =_ wechatyGRPC;
- wechaty-hostie=_ Grpc



- GRPC;
- GRPC
- Schema
- ContactMessage



Grpcmock(<https://github.com/Fadelis/grpcmock>
<https://github.com/tomakehurst/wiremock>)

```
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)
```





-
- token
-
-
-





- GRPC;
- token
- ts
- AtomElectronUI
- Schema(antlr);
- puppethostieschema





Future Work

- wechaty;
- ;
- ;
- ();



Thank you

Thank you for listening!



Questions?

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

