

ArchData

技术峰会成都站

主办方： 

2018年1月27日成都菁蓉国际广场3W COFFICE 蓉漂茶馆

实战

数据权限

数据扩展

数据展现

数据融合



罗果

<http://www.tinygroup.org/>

个人简介

热爱软件领域，初心不改

主要技术领域为**J2EE及应用开发平台**领域，涉猎广泛，在模块化、元数据、模板引擎、脚本语言、数据库分库分表、数据权限、SOA等诸多领域等都有较深入实践，著有《**企业级Java EE架构设计精深实践**》一书。
拥有极强的抽象能力，可以层层抽丝剥茧分析业务本真，并在业务抽象和技术实现方面做到良好均衡。



恒生电子

业委会首席架构师
技术部门经理

TINY开源组织

发起人



架构、设计能力

90%

语言能力

90 %

管理能力

80 %

开源情况



悠悠然然 🏠

共码了 360463 个字

访问统计

今日访问: 2

昨日访问: 79

本周访问: 412

本月访问: 412

所有访问: 256247

从2013年起，业余时间开展了开源框架Tiny的研究并得到了公司的认同，Tiny框架也在恒生中有相当的使用，基于Tiny的产品一年销售额2亿+。

开源期间，书写博客180余篇，码字36万+，UV25万+，Pv百万+。

所写博客被云栖社区主动收录。

数据权限 TinyDAC

www.tinygroup.org

TinyDAC

What is it?

www.tinygroup.org

什么是数据权限

数据权限有两种表现形式：一种是行权限，另外一种是列权限。

所谓行权限，就是限制用户对某些行的访问权限，比如：只能对本人、本部门、本组织的数据进行访问；也可以是根据数据的范围进行限制，比如：合同额大小来限制用户对数据的访问。

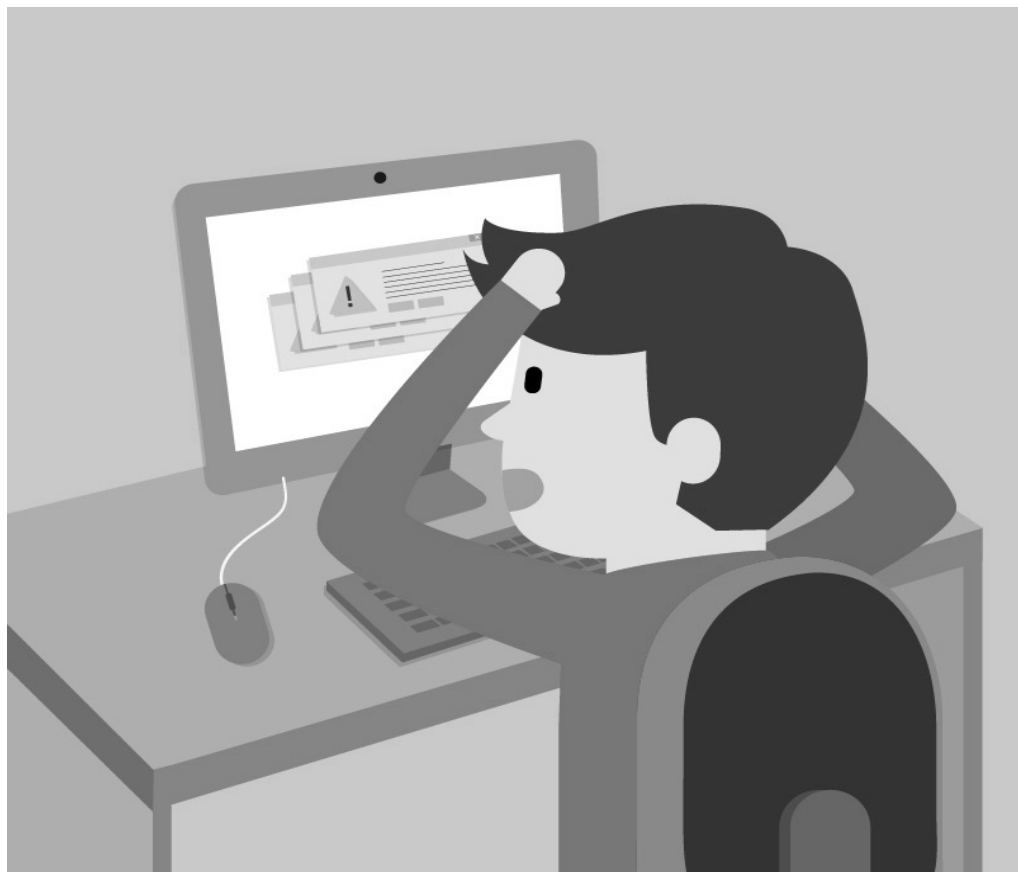
所谓列权限，就是限制用户对某些列的访问权限，比如：某些内容的摘要可以被查阅，但是详细内容就只有VIP用户才能查看。



存在的问题

传统的数据权限一般在DAO层通过SQL拼接实现或在获取了数据之后根据权限进行过滤两种方案，这些方案存在明显的不足：

- 性能较低
- 程序不透明，数据权限相关代码硬编码
- 依赖具体的数据库访问层
- 不能达到一处控制，到处控制的效果
- 不能起到即插即用效果
- 代码改变，需要重新部署不能进行热修改
- 代码编写工作量较大
- 代码重构工作量较大



TinyDAC Features

www.tinygroup.org

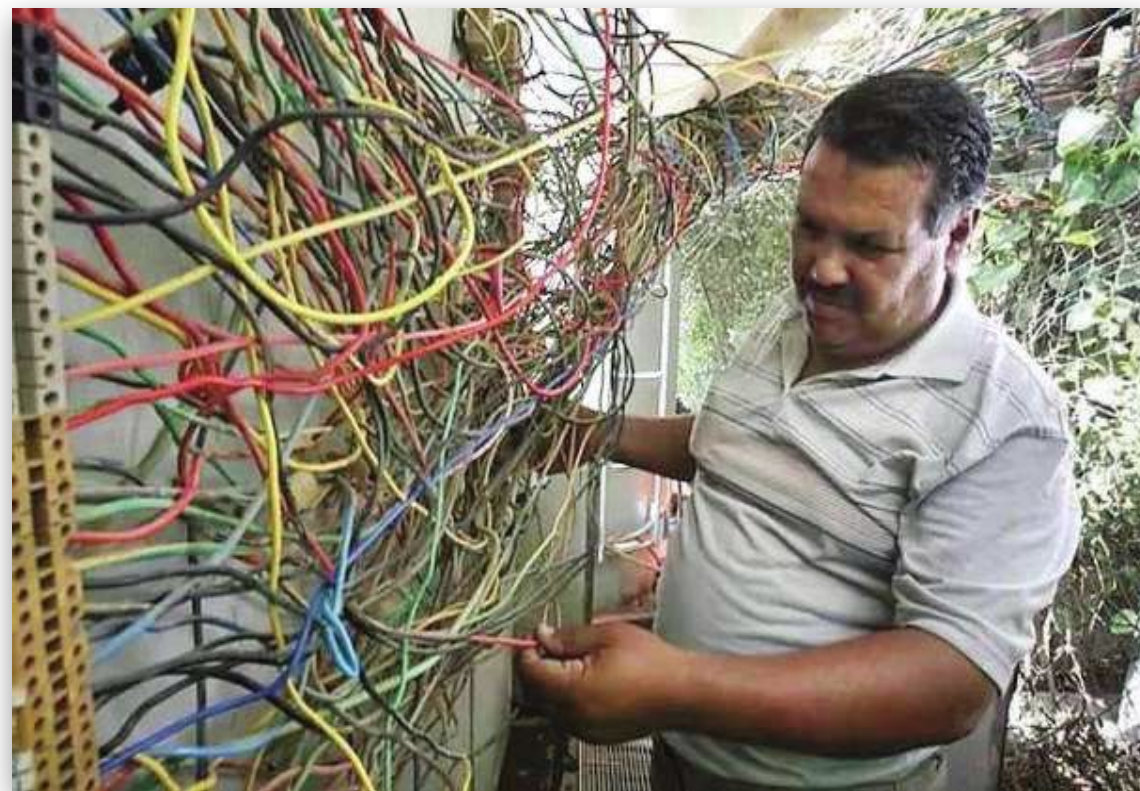
TinyDAC-无侵入性

侵入性的特征就是用户业务人员写代码时必须“知道”框架的代码，表现为用户代码需要引入框架提供的接口或类。

非侵入式则不需要用户代码引入框架代码的信息，从类的编写者角度来看，察觉不到框架的存在。tiny数据权限是基于JDBC层实现的，应用程序编写与以前的做法一模一样。

无侵入性框架优点：

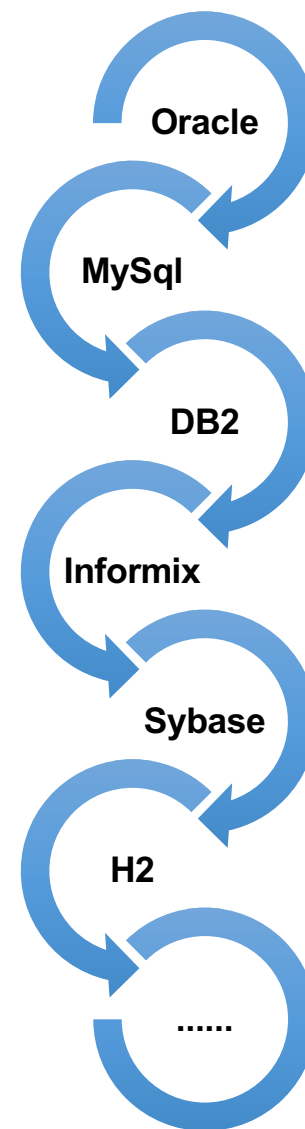
- 业务逻辑编写简单
- 方案易于实施
- 普适性好



TinyDAC-支持各种数据库

跨数据库：框架支持在多种数据库环境下运行，目前支持MySQL、Oracle、DB2、SqlServer、Postgresql、H2数据库。

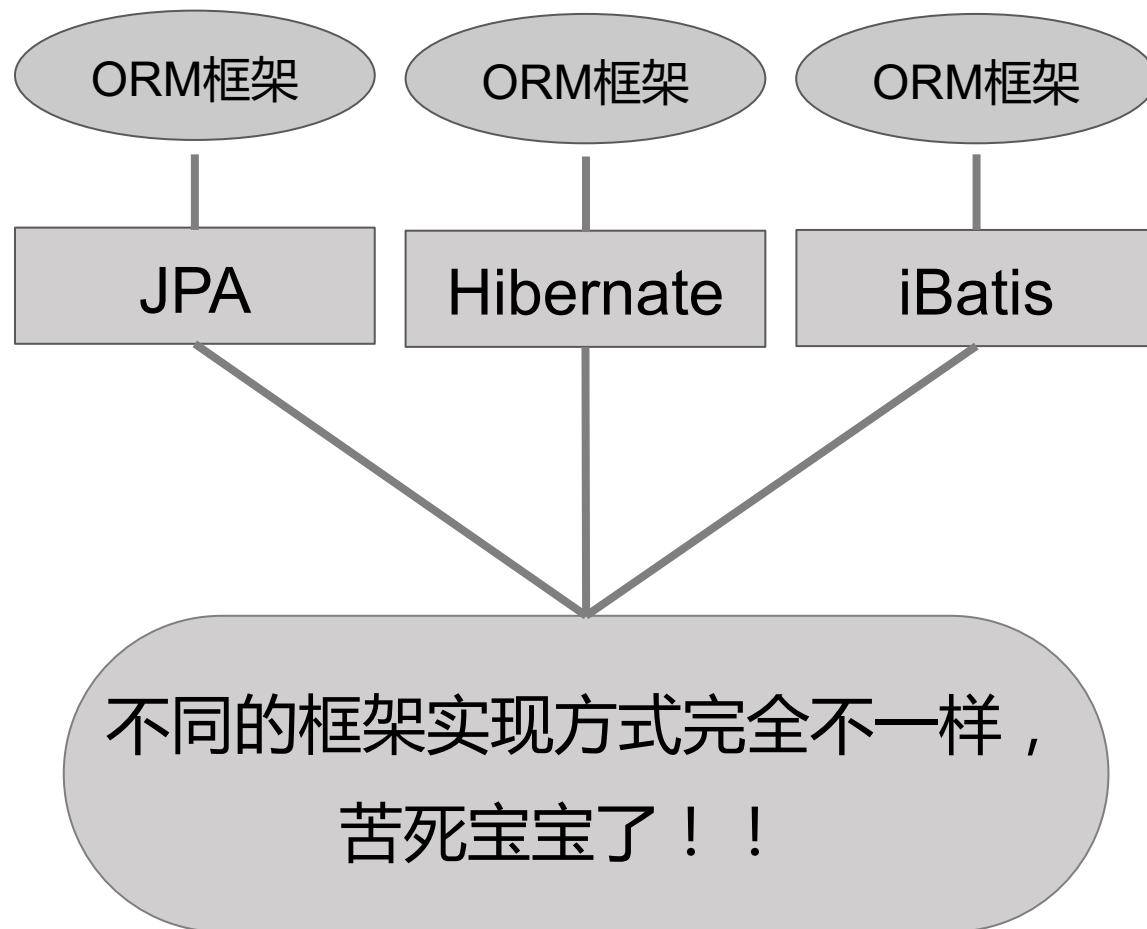
用户代码不需要区分各种数据库，只要绑定实际物理连接即可，框架内部会根据连接选择对应的SQL解析器，进行SQL解析处理。



TinyDAC-和ORMapping框架集成

框架基于JDBC底层实现的，可以理解为JDBC驱动力的增强版实现。它是轻量级框架，以jar包形式提供服务，无proxy代理层，无需额外部署，只需依赖底层数据库的驱动：

- 适用于任何第三方数据库连接池如：DBCP, C3P0, Druid等
- 适用于任何基于java的ORM框架，如：IBATIS, Hibernate, Mybatis, Spring JDBC Template或直接使用JDBC
- 使用方式与传统的数据库的使用方式相同，使用者无需花费太多精力重新学习，有基于数据库驱动程序访问数据库基础即可。



TinyDAC-易于使用、管理严密

一处控制，到处生效

传统数据权限方案对某张表的数据访问控制逻辑可能会四处散落，不能起到一处控制到处生效的作用。

Tiny数据权限方案可以控制到表，只要对表的数据库操作，都能生效，对表的控制处理逻辑都是接口化的，可以根据需求实现对应的处理逻辑。

每种权限控制都存在两种方式：

- 1、自定义实现接口方式。然后在配置文件中设置实现类全路径
- 2、基于groovy表达式方式，可以在配置文件中定义groovy表达式语言



TinyDAC-集成方式

框架支持2种使用方式：

- JDBC Driver方式

与传统的JDBC程序一模一样，通过
DriverManager.getConnection方法来获取连接

- 数据源方式

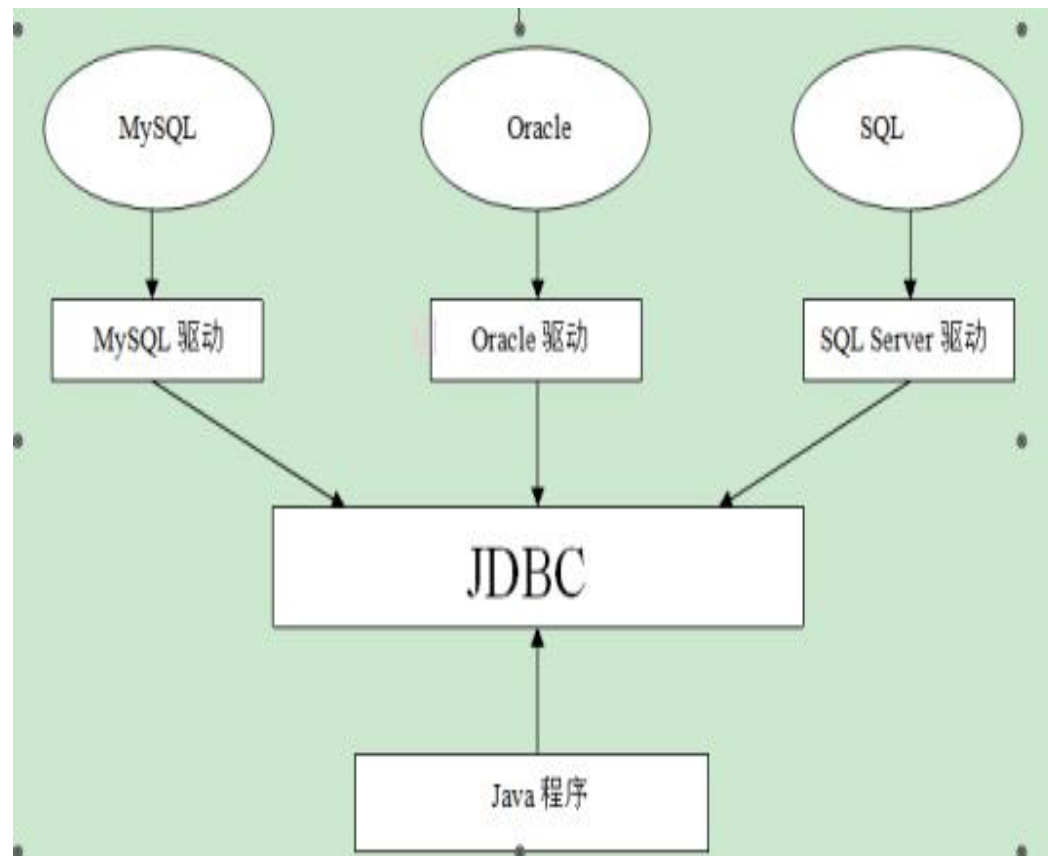
框架提供数据源java.sql.DataSource接口的实现

- 1、单数据源方式：

SingleDataAccessControlDataSource

- 2、多数据源方式：

MultiDataAccessControlDataSource,可以根据绑定在线程变量的数据源名称来选择数据源。



TinyDAC-配置方式

框架支持3种配置方式：

- xml配置方式
- spring自定义命名框架配置方式
- yaml配置方式



TinyDAC Examples

www.tinygroup.org

TinyDAC-行权限示例

场景:协同办公系统，hr能查询已经离职的员工，普通员工不能查询已经离职员工信息。

规则匹配策略：

```
<dac:rule-match id="selectRuleMatch"
  match-expression="select_user_type==null || !'hr'.equalsIgnoreCase(select_user_type)" />
```

附加条件处理：

```
<dac:select-rule id="selectRule">
  <dac:append-conditions>
    <dac:append-condition
      append-condition-class-name="" append-condition-expression="${ALIAS_NAME}.status=1"
      match-strategy="selectRuleMatch" />
    </dac:append-conditions>
  </dac:select-rule>
```

TinyDAC-行权限示例

场景:只有超级管理员才有权限删除员工信息,非超级管理员只能删除部分数据

规则匹配策略:

```
<dac:rule-match id="selectRuleMatch"
  match-expression="select_user_type==null || !'manager'.equalsIgnoreCase(select_user_type)" />
```

附加条件处理:

```
<dac:select-rule id="deleteRule">
  <dac:append-conditions>
    <dac:append-condition
      append-condition-class-name="" append-condition-expression="${ALIAS_NAME}.status=1 and ${ALIAS_NAME}.type=1"
      match-strategy="selectRuleMatch" />
    </dac:append-conditions>
  </dac:select-rule>
```

TinyDAC-行权限示例

场景:每个租户只能查到自己的表，表中有个租户id的字段，根据传入的租户id，添加条件规则匹配策略：

```
<dac:rule-match id="selectRuleMatch" match-expression="tenant_id!=null />
```

附加条件处理：

```
<dac:select-rule id="selectRule">
  <dac:append-conditions>
    <dac:append-condition
      append-condition-class-name="" append-condition-expression="${ALIAS_NAME}.tenant_id=tenant_id"
      match-strategy="selectRuleMatch" />
    </dac:append-conditions>
  </dac:select-rule>
```

TinyDAC-列权限示例

场景:hr可以看到员工的工资, 普通人员看到的是0

规则匹配策略:

```
<dac:rule-match id="selectRuleMatch"
  match-expression="select_user_type==null || !'hr'.equalsIgnoreCase(select_user_type)" />
```

值过滤处理:

```
<dac:select-rule id="selectRule">
  <dac:value-filters>
    <dac:value-filter match-strategy="selectRuleMatch">
      <value-filter-expression>
        <![CDATA[["salary":"0"]]]>
      </value-filter-expression>
    </dac:value-filter>
  </dac:value-filters>
</dac:select-rule>
```

TinyDAC-列权限示例

场景:禁止不带任何条件的删除, 除非是超级管理员(super)
值检查处理:

```
<dac:delete-rule id="deleteRule">
  <dac:value-checks>
    <dac:value-check value-check-expression="delete_user_type!=null&&'super'.equalsIgnore(delete_user_type)"/>
  </dac:value-checks>
</dac:delete-rule>
```

数据扩展 TinyDBRouter

www.tinygroup.org

为什么需要数据库水平扩展？



- 单数据库出现瓶颈
单服务器CPU、内存、磁盘IO、网络IO都是有限的。
- 单个数据表中的记录条数太多
CRUD都成问题；索引膨胀，查询超时 解决方法：
切分成多个数据集更小的表

常见解决方案

实现层面	产品与框架
数据访问层(DAO层)	项目自行实现
DAO与JDBC间的Spring数据访问封装层	CobarClient
JDBC层	sharding-jdbc、tddl
ORM框架层	Guzz 、 HibernateShards
应用服务器与数据库间的代理层	Routing4DB、Mycat、MysqlProxy、Amoeba

TinyDBCluster有什么特点

支持各种常见数据库

- Mysql、Oracle、Db2、SqlServer、Informix, Postgresql等等

支持各种使用场景

- 适用于任何第三方数据库连接池如：DBCP, C3P0, Druid等
- 适用于任何基于Java的ORM框架，如：iBatis, Hibernate, Mybatis, Spring JDBC Template或直接使用JDBC
- 使用方式与传统的数据库的使用方式相同，使用者无需花费太多精力重新学习，有编写数据库驱动程序访问数据库基础即可。

支持驱动方式和数据源模式

支持多种配置方式

- Xml配置方式
- spring自定义命名框架配置方式
- 基于动态语言的分片策略yaml配置方式

支持配置中心

TinyDBCluster功能特性

Connection接口

不支持存储过程，函数，游标的操作

不支持执行native的SQL

不支持savepoint相关操作

Statement和PreparedStatement接口

不支持返回多结果集的语句（即存储过程，非SELECT多条数据）

不支持国际化字符的操作

对于ResultSet接口

有限支持修改结果集内容

不支持获取国际化字符

支持获取Array

SQL语句限制

有限支持子查询

不支持HAVING

不支持OR，UNION 和 UNION ALL

不支持特殊INSERT

每条INSERT语句只能插入一条数据，不支持VALUES后有多行数据的语句。不支持INSERT...SELECT语句

不支持DISTINCT聚合

不支持dual虚拟表

不支持SELECT LAST_INSERT_ID()

不支持CASE WHEN

对GROUP BY 和ORDER BY语句会在查询项加入新的查询字段，如果整个查询语句作为in条件的子查询，会出现SQL语句执行错误。

例如下面这样的查询片段：col1 in (select count(*) from table group by col2)



TinyDBCluster对工具的支持



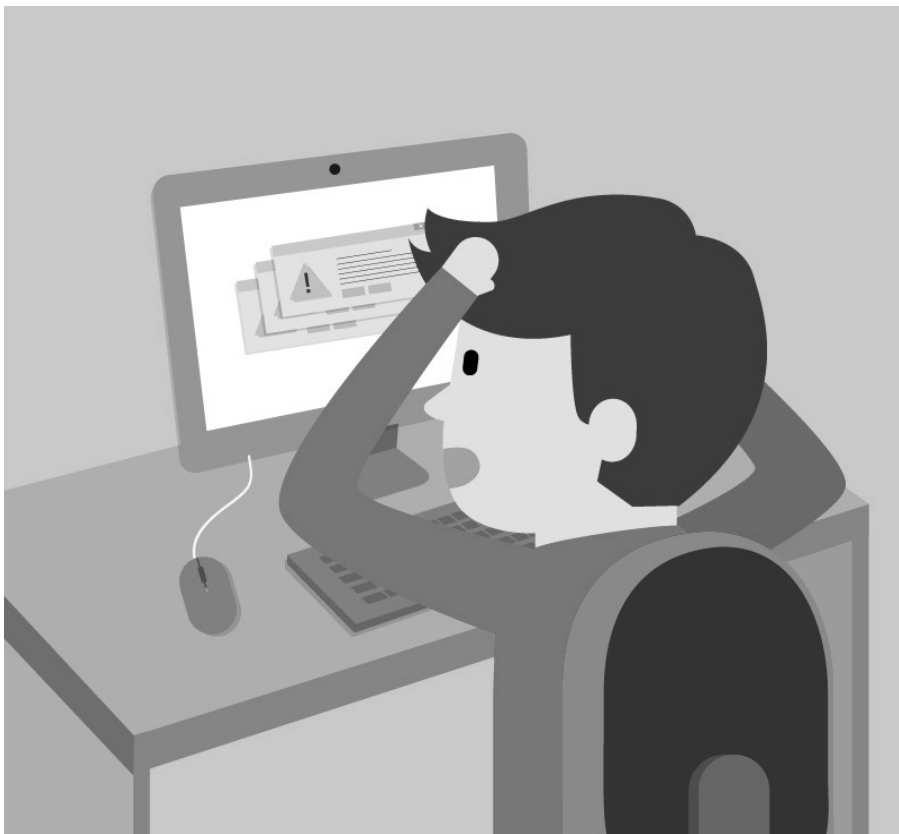
各种支持自定义JDBC Driver的工具都可以使用

数据展现 TinyBI

www.tinygroup.org

背景说明

- 业务人员、管理人员经常需要各种数据
- 找技术写SQL取数查看各个维度的数据
做决策成本高、反应慢，还难看
- 报表产出效率低，维护难
- 后台分析系统的数据报表变更，编码研发周期长，维护困难
- 图表效果设计不佳，人力成本高
- 使用专业Tableau等工具做报表成本高、
安装使用复杂，与现有系统无法无缝集成

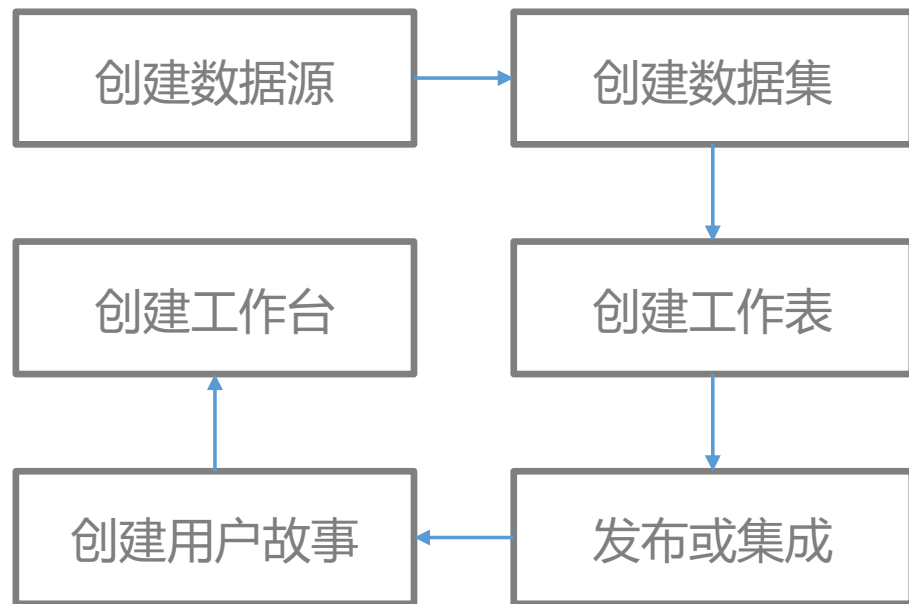


功能特性

- 无缝集成各种数据来源
 - MySQL、Oracle、DB2、...
 - MongoDB
 - Hbase
 - Excel
 - CSV
 - ...
- 方便易用的用户体验
 - 拖拉式操作
 - 强大的数据再加工能力
 - Tableau的样子，tiny的心
- 强大的数据再加工能力
 - 底层采用TinyScript、TinyDAC、TinyDBRouter支撑，由此具备强大的数据清洗、重组、库外计算、异构数据整合等能力
- 展现采用Echarts，并进行功能扩充



使用过程





< 图表项 数据源 图 >

筛选器

行

列

销售报表分析



COMING SOON

维度

公司

地域

城市

度量

销售额

人口数

数据融合 TinyScript

www.tinygroup.org

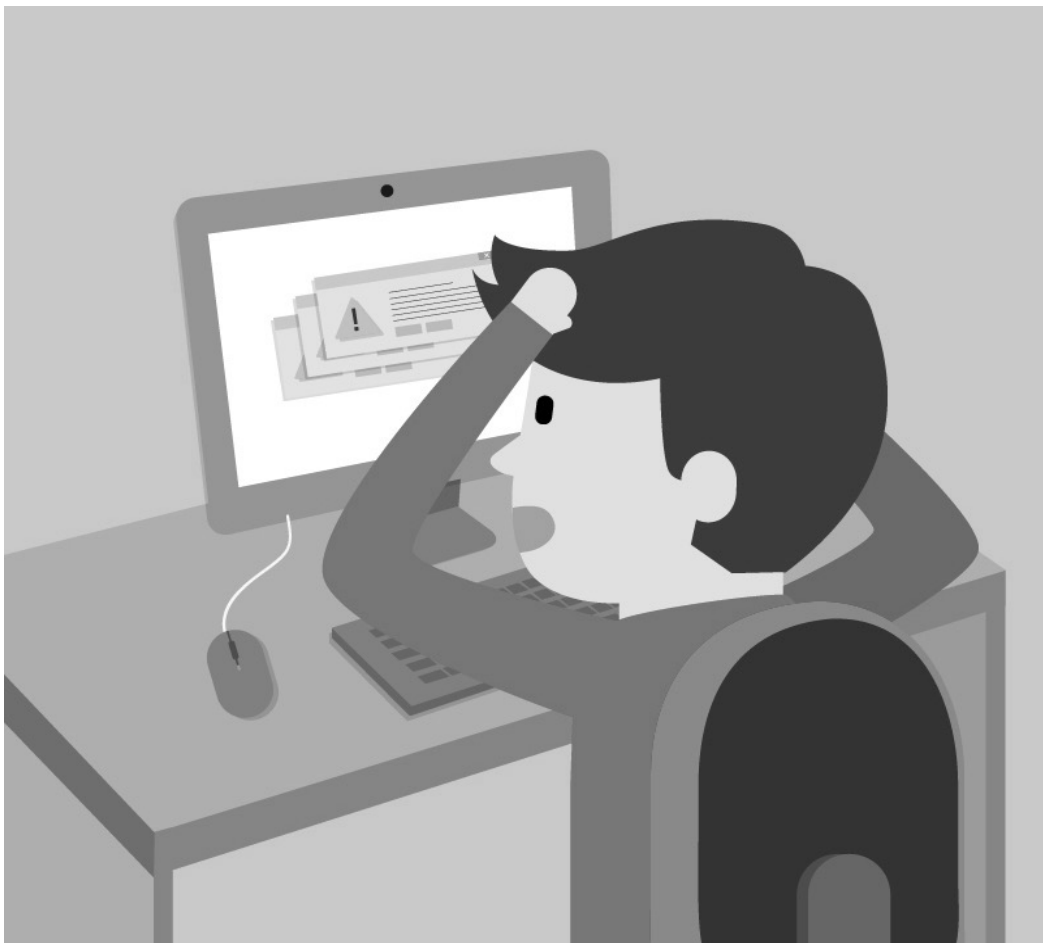
TinyScript

What is it?

www.tinygroup.org

背景介绍

- ✓ 业务逻辑日益复杂，难以维护
- ✓ 代码长度越来越长
- ✓ 第三方语言种类繁多，集成困难
- ✓ SQL标准难以实现传统编程语言的特性
- ✓ 不同数据库差异大，维护迁移不容易
- ✓ 业务逻辑大量使用数据库，难以扩展
- ✓ 语言层面缺少对结构化类库的支持，过滤、分组、排序、关联都非常困难
- ✓ Python 等语法结构与Java偏差较大，用户上手不易
- ✓ 一般脚本语言难以满足多样化业务场景



使用场景

- ✓ **EL表达式**
获取数据，执行计算
引用上下文对象
调用Java方法
- ✓ **数据服务**
支持多种数据源
多源混算，屏蔽底层差异
更容易写的数据服务层
- ✓ **ETL编排**
支持ETL组件化
通过脚本定义组件
可视化的IDE插件
- ✓



TinyScript Features

www.tinygroup.org

功能特性介绍-支持各种集合运算

```
a = [1..5];
b = [3..7];
println(a+b);      [1,2,3,4,5,6,7]
println(a-b);      [1,2]
println(b-a);      [6,7]
println(a^b);      [1,2,6,7]
println(a&b);       [3,4,5]
println(a*3);       [3,6,9,12,15]
println(sum(a));    15
println(max(a));    5
println(min(a));    1
```

- 脚本语言支持集合的差、并、交、异或运算
- 集合运算同时支持函数和操作符两种方式
- 脚本语言还支持集合的聚合运算
- 脚本语言支持通过集合直接调用元素API，无需循环

功能特性介绍-脚本类

```
class User{
    name,age;
    User(name,age){
    }
    User(name){
    }
}
user = new User("张三",22);
println(user);
userList = [new User("张三",22),new User("李四",25),new User("王五",32)];
println(userList.filter((it)->{it.age>24}));
println(userList.filter((it)->{it.age>24 && it.name.startsWith("王")}));
println(avg(userList.age));
```

- 构造函数，参数名与属性名相同，自动赋值
- 属性的set、get免写
- 用户可重写toString等方法
- 统计、过滤函数支持脚本类

功能特性介绍-字符串模板功能

```
var =123;  
println("var=${var}");  
println("var+3=${var+3}");  
println("5+3=${5+3}");
```

```
var=123  
var+3=126  
5+3=8
```

- 在Java中，要么采用字符串拼接，要么采用StringBuffer，但是不管哪一种，都比较拖沓不自然。
- 本脚本语言不仅支持变量占位符，而且支持完整的表达式计算。

功能特性介绍-lambda支持

```
helloWorld(greeting,name)->{  
    println("${greeting},${name}");  
};  
helloWorld("Hello","悠然");
```

```
userList = [new User("张三",22),new User("李四",25),new User("王五",32)];  
userList.sort((right,left)->{return  
right.age>left.age});
```

- 支持命名lambda函数和匿名lambda函数
- 简化代码逻辑，避免冗余代码
- 匿名Lambda可以作为参数，允许用户实现自定义逻辑，提高灵活性

功能特性介绍-排列组合

```
elements = [0,1,2];
elements.permute(2,(record) -> {
    println(record);
});
elements.combine(2,(record) -> {
    println(record);
});
elements.permuteAll(2,(record) -> {
    println(record);
});
```

- 目前支持排列、组合和全排列三种
- 通过排列组合的lambda接口，用户可以在遍历过程中实现自定义逻辑

功能特性介绍-注册函数

`object.method(args...);` //绑定对象类型
`method(args...);` //无绑定类型

- 注册函数是脚本语言非常重要的扩展机制
- 脚本语言内置了近百种注册函数，按调用方式的不同可以分为绑定对象类型和无绑定类型两种函数
- 脚本类型如果按场景分,可以划分三角函数、数学函数、聚合函数、日期函数、字符串函数、转换函数、数据库函数、序表函数、序列函数等，用户可以根据自己的需要进行选择

功能特性介绍-序表操作

```
dataSet = [[ select * from user ]].toDynamic();  
dataSet.next();  
dataSet.previous();  
dataSet.first();  
dataSet.last();  
dataSet.absolute(1);  
dataSet[1][1] = "Boss Tom";  
dataSet["A1"] = 12.5d;  
dataSet.update("price","WEIGHT*150");  
dataSet.sort("id asc,price desc");  
dataSet.filter()->{return price>100;});
```

- 序表支持从数据库、Excel等多种数据源加载
- 序表同时支持遍历和下标两种操作方式
- 序表功能强大，可以进行关联、排序、分组、过滤等各种操作

功能特性介绍-支持Spring Bean

```
println(user); //调用bean  
user = new User("钱九",25);  
println(user); //本地变量优先级高于bean
```

```
dataSet = mysql.query("select * from user");  
mysql.execute("delete from user where  
sex=1");
```

- 脚本语言与Spring的Bean无缝集成，用户可以直接使用
- 相同名称的变量/属性优先级高于同名Bean
- Bean机制减少重复配置，如数据库数据源

功能特性介绍-与Java的完美结合

```
import org.apache.commons.lang.StringUtils;
import java.util.LinkedHashMap;
```

```
System.out.println("调用Java方法");
println(StringUtils.trim("import第三方资源"));
```

```
linkmap = new java.util.LinkedHashMap(); //完整类路径
linkmap = new LinkedHashMap(32); //import资源类, 省略路径
```

```
dataSet.toList(); //转换map的序列
dataSet.toList("org.tinygroup.User"); //转换指定类的序列
dataSet.toList()->{
    user = new org.tinygroup.User();
    user.sex = dict(sex);
    user.userName = name;
    return user;
});
```

- 所有的java资源都可以被脚本语言使用
- 脚本语言的结果可以返回给Java使用
- 脚本语言提供人性化设计, 用户可以更方便操作Java对象

功能特性介绍-集成第三方语言

```
userDs = dataSource [[select * from user where status='1' and  
name=@name]];  
dataSource [[delete from user where name=@name]];
```

```
renderContent = templateRender[[  
  <html><body>  
    #if(user!=null)  
      Hello,{user.name}  
    #else  
      Hello,anonymous  
    #end  
  </body></html>  
]];
```

- 在TinyScript中，可以方便的集成各种DSL/脚本语言，采用最合适的语言来有针对性的解决具体的问题，以便发挥最大的语言效能。
- 语言扩展可以充分利用框架资源，节约开发时间
- 内建已经支持了SQL和模板语言

功能特性介绍-扩展能力

- 可以扩展注册函数
- 可以扩展聚合函数：一次开发多场景使用
- 可以扩展语言包
- 可以扩展操作符的处理逻辑
- 可以扩展内嵌语言类型：默认支持SQL、Tiny模板



功能特性介绍-友好的异常提示

Exception in thread "main" [org.tinygroup.tinyscript.interpret.exception.InterpretFormatException](#):

第三方异常:/ by zero

异常类:[java.lang.ArithmeticException](#)

表达式异常:a=2,b=0进行运算发生异常

错误位置:[8,10]-[8,12]

错误文本:a/b;

函数异常:执行函数[devide]发生异常

错误位置:[14,9]-[14,21]

错误文本:devide(sum,c);

函数异常:执行函数[fun1]发生异常

错误位置:[19,10]-[19,20]

错误文本:fun1(i,0,0));

函数异常:执行函数[println]发生异常

错误位置:[19,2]-[19,21]

错误文本:println(fun1(i,0,0));

```
at org.tinygroup.tinyscript.interpret.ParserRuleContextSegment.execute(ParserRuleContextSegment.java:146)
at org.tinygroup.tinyscript.interpret.ParserRuleContextSegment.execute(ParserRuleContextSegment.java:118)
at org.tinygroup.tinyscript.impl.AbstractScriptEngine.execute(AbstractScriptEngine.java:83)
at org.tinygroup.tinyscript.executor.AbstractTinyScriptOperator.execute(AbstractTinyScriptOperator.java:54)
at org.tinygroup.tinyscript.executor.DefaultTinyScriptExecutor.main(DefaultTinyScriptExecutor.java:31)
```

- 脚本语言考虑初学者，异常信息包含完整的错误链
- 异常信息包含错误位置和错误原因，方便用户定位
- 脚本语言的异常提示也实现了国际化

功能特性介绍-国际化方案

```
Exception in thread "main" org.tinygroup.tinyscript.interpret.exception.InterpretFormatException:
Other Exception:/ by zero
Exception Class:java.lang.ArithmeticException

Expression Exception:a=2,b=0 performs an arithmetic exception
Exception Place:[8,10]-[8,12]
Exception Text:a/b;

Function Exception:An exception occurred with the execution function [devide]
Exception Place:[14,9]-[14,21]
Exception Text:devide(sum,c);

Function Exception:An exception occurred with the execution function [fun1]
Exception Place:[19,10]-[19,20]
Exception Text:fun1(i,0,0));

Function Exception:An exception occurred with the execution function [println]
Exception Place:[19,2]-[19,21]
Exception Text:println(fun1(i,0,0));

at org.tinygroup.tinyscript.interpret.ParserRuleContextSegment.execute(ParserRuleContextSegment.java:146)
at org.tinygroup.tinyscript.interpret.ParserRuleContextSegment.execute(ParserRuleContextSegment.java:118)
at org.tinygroup.tinyscript.impl.AbstractScriptEngine.execute(AbstractScriptEngine.java:83)
at org.tinygroup.tinyscript.executor.AbstractTinyScriptOperator.execute(AbstractTinyScriptOperator.java:54)
at org.tinygroup.tinyscript.executor.DefaultTinyScriptExecutor.main(DefaultTinyScriptExecutor.java:31)
```

- 脚本语言支持国际化方案，方便语种不是中文的用户
- 国际化方案目前支持中英文，根据Locale调用不同的语言信息
- 脚本语言内部信息全部采用配置，扩展新的语言包无需编码

功能特性介绍-强大的脚本编辑器



```
1 collection.tsf
2
3 //List集合模型
4 list1=[1,2,3]; list2=["dog","cat","bird"].toUpperCase();
5 println(list1+3);
6 println(list2);
7
8 //Map集合模型
9 map1={"key1":100,"key2":200,"key3":300};
10 map2={1:"Hello",2:"Tom",3:"John"}.toLowerCase();
11 println(map1/100);
12 println(map2);
13
14 //Set集合模型
15 set1=map1.keySet().substring(1,4);
16 set2=map2.keySet();
17 println(set1);
18 println(set2*10);
19
20 if(a>1){
21 }else{
22 }
23
```

脚本 预览

- 再好的语言也离不开优秀的工具支持
- 脚本编辑器包含语法高亮、智能提示、标识符定位、语法块折叠、错误提示等特性
- 脚本编辑器可以帮助用户编辑、调试、运行脚本，提升开发效率

功能特性介绍-还有更多

限于篇幅，我们不能穷
举所有特性，下一节我
们通过示例来展示



TinyScript Examples

www.tinygroup.org

示例介绍-计算青歌赛选手成绩

青年歌手参加歌曲大奖赛，有 n ($n > 2$) 个评委打分，试编程求选手的平均得分（去掉一个最高分和一个最低分）

```
getScore(scores)->{  
    return (sum(scores)-max(scores)-min(scores))/(scores.size()-  
2);  
};
```

示例介绍-01背包求解

有编号分别为a,b,c,d,e的五件物品，它们的重量分别是“2,2,6,5,4”，它们的价值分别是“6,3,5,4,6”，现在给一个承重为10的背包，如何让背包里装入的物品具有最大的价值总和？

```
class Obj{  
    name,weight,value;  
    Obj(name,weight,value){  
    }  
}  
list=[new Obj("a",2,6.0),new Obj("b",2,3.0),new Obj("c",6,5.0),new  
Obj("d",5,4.0),new Obj("e",4,6.0)];  
println("01背包问题:\n"+list.dpKnapsack(10,list.weight,1,list.value));
```

示例介绍-关联不同的序表-代码

实现四种关联操作的示例

//测试内联

```
inner = ds1.join(ds2,"userid=userid");  
println(inner);
```

//测试左联

```
left = ds1.joinLeft(ds2,"userid=userid");  
println(left);
```

//测试右联

```
right = ds1.joinRight(ds2,"userid=userid");  
println(right);
```

//测试全联

```
full = ds1.joinFull(ds2,"userid=userid");
```

示例介绍-关联不同的序表-结果

userid username password userid jifen dengji

1 jack jackpwd 1 20 3

=====

userid username password userid jifen dengji

1 jack jackpwd 1 20 3

2 open openpwd null null null

=====

userid username password userid jifen dengji

1 jack jackpwd 1 20 3

null null null 3 50 6

=====

userid username password userid jifen dengji

1 jack jackpwd 1 20 3

2 open openpwd null null null

null null null 3 50 6

=====

示例介绍-数据字典

加载字典表配置，获取字典显示值

```
dict(key1)->{  
    return dict1[key1]; //读取一级数据字典  
};  
dict1 = new java.util.HashMap();  
dictDs = [[ select * from dict1 ]].toDynamic();  
dictDs.forEach((row)->{  
    dict1.put(dict_code,dict_name);  
});  
userDs = [[ select * from user ]].toDynamic();  
userDs.update("sex","dict(sex)"); //替换显示值
```

示例介绍-ETL示例

将excel中历年的上证指数，转换后写入数据库

//验证数据集的写入

```
ds = readCsv("/other/000001.csv","gbk").update("涨跌额",涨跌额=="None"?0:涨跌额).update("涨跌幅",涨跌幅=="None"?0:涨跌幅)
. update("成交金额",成交金额=="None"?0:成交金额).double("收盘价","最高价","最低价","开盘价",
"涨跌额","涨跌幅");
ds.forEach((row)->{
    [[ insert into
stock000001(tradeDT,code,name,openingPrice,preclosingPrice,closingPrice,maxPrice,minPrice,changeAmount,changeRate,hand,myriad)
    values(@日期,@股票代码,@名称,@开盘价,@前收盘,@收盘价,@最高价,@最低价,@涨跌额,@涨跌幅,@成交量,@成交金额);
    ]];
});
```

示例介绍-优化投资组合

小王有一笔钱 m ，在 t 时间内，想做理财投资，有如下几只理财产品可供选择，请给出最佳投资组合方案。

```
class Product{
    name,amountPerServing ,maxCount,rate;//name:基金名字
    amountPerServing:一份的价格 maxCount : 最大份数 rate : 利率
    Product(name,amountPerServing,maxCount,rate){
    }
}

list=[new Product("鹏华国防",100,10,0.00045),new Product("鹏华中证",100,20,0.00035),new Product("国投瑞银",100,20,0.00055),new Product("华商主题精选",100,10,0.0004),new Product("金鹰智慧",100,5,0.0003)];
```

示例介绍-优化投资组合~运算结果

```
println(list.dpKnapsack(5000,list.amountPerServing,list.maxCount,()->{  
    //每个基金的总收益由利率*时间*一份价格  
    return list.rate*days*100;  
}));
```

5000元80天优选投资组合：(国投瑞银 2000 鹏华国防 1000 华商主题精选 1000 鹏华中证 1000),预计收益：184.0

3000元30天优选投资组合：(国投瑞银 2000 鹏华国防 1000),预计收益：46.5

3000元60天优选投资组合：(国投瑞银 2000 鹏华国防 1000),预计收益：93.0

2000元20天优选投资组合：(国投瑞银 2000),预计收益：22.0

TinyScript Environment

www.tinygroup.org

运行和使用环境

- ✓ 推荐与Java应用集成在一起使用
- ✓ 以JDK为支撑，可以独立使用



TinyScript

扩充Java语法、优化代码结构、简化程序逻辑。

让程序员们有更多的时间与家人在一起，让公司有更多的利润空间。

其他作品

作品展示~Tiny框架

VFS

元数据

模板语言

DB设计

服务中心

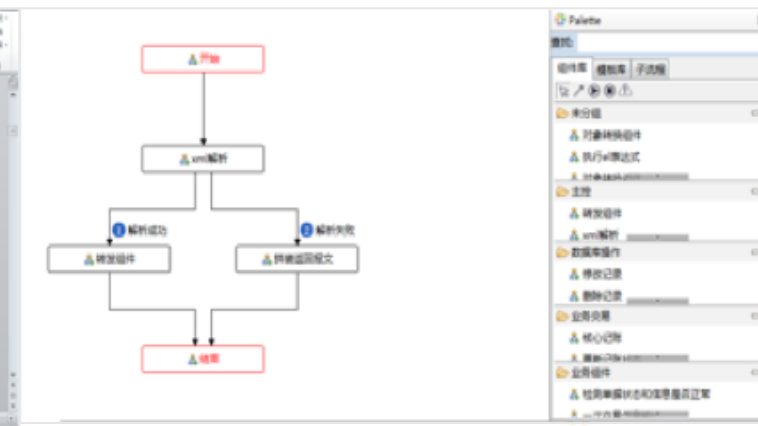
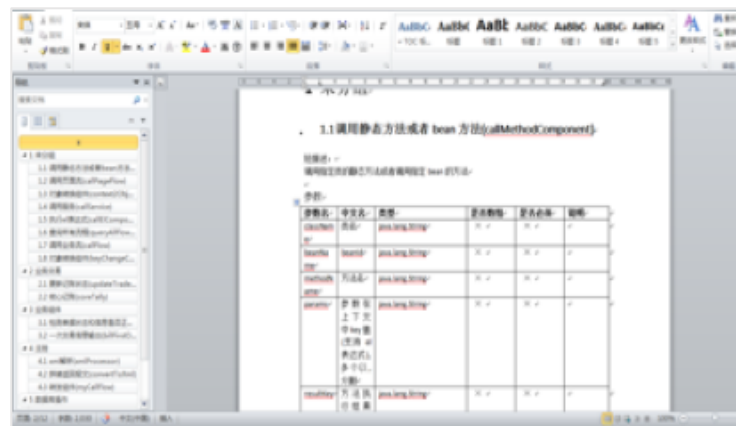
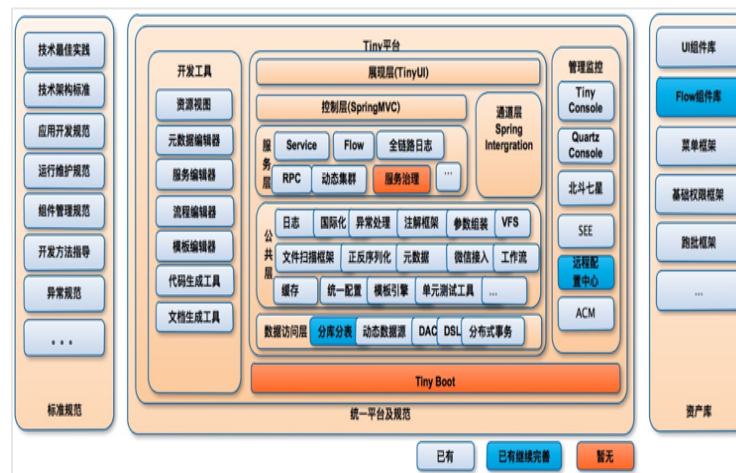
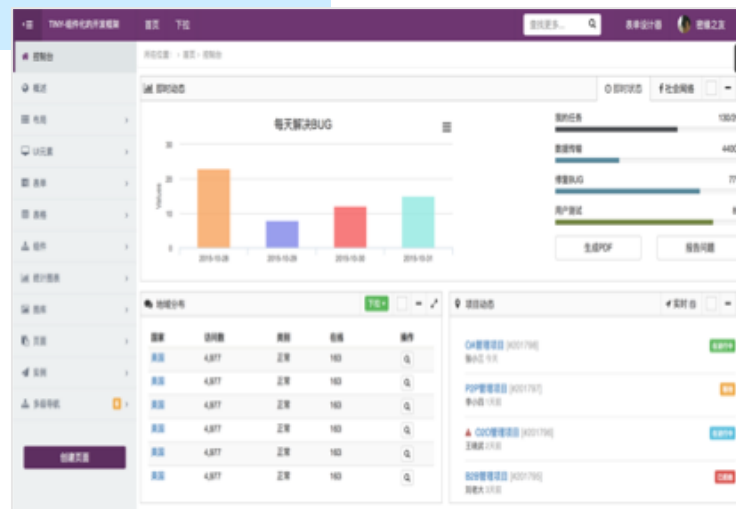
配置中心

UI框架

文档生成

流程编排

相关内容：
<http://www.tinygroup.org/docs/04b2cd7aa22b427fa3dc8fd0fea52f5f>



作品展示~协作平台<http://www.tinyapp.top/>

论坛

博客

简报

看板

知识库

...

技术平台支持
插件话体系
SaaS支持
功能切面化分割
数据互通
版本对比
关系图谱

...



相关内容：<https://my.oschina.net/tinyframework/blog/787546>
演示网站：<http://tinyappdemo.tinyapp.top>，用户密码：demo/123456

THANKS

**感谢您的聆听，下面
是自由提问时间！**





中生代技术
FRESHMAN TECHNOLOGY

ArchData技术峰会
聚焦人工智能、大数据、基础架构、
区块链等前沿课题

中生代技术提供咨询内训服务
技术架构，研发管理，敏捷开发，大
数据微服务，AI，机器学习等

中生代技术提供人才服务对接研发主
管，内推精准人才