Atitit 提升扩展性 自由化理念 参数

# 前瞻性设计

# 自由化理念

自由化参数 json map等半结构化参数，dsl等全功能参数

自由化功能接口 dsl

自由化返回数据结构json

# 热部署等实现模式

Xml+sql mybatis

Script php js python dsl

Cmd hot load模式。。。

Atitit 架构之提升扩展性问题

目录

1. 扩展性大原则 2

1.1. 适当向上抽象 更加通用 2

1.2. 产品化 为“普罗大众”去设计，而不是为“个例”设计 2

2. 扩展性指标 3

2.1. 前瞻性设计，接口不只是为了已知的业务，还为未来的未知业务而服务 3

2.2. 方便修改，容易添加某个业务功能尽可能无需改动程序代码 或少量改动 3

2.3. 修改某个部分避免跨越太多多文件 3

2.4. 动态化 实现少编译免编译 热部署 或免部署 3

2.5. 松耦合(ioc) 业务无关性 3

2.6. 库表结构自动化扩展，随时添加字段 修改字段名 而无需调整程序代码 3

2.7. 通用化接口，少量接口实现大量的业务功能，接口业务无关性 3

2.8. DSL声明化（不具体指明细节），结果导向模型 4

3. Top3 是三件套dsl业务脚本 通用接口 半结构化数据 4

3.1. 无中心节点组件设计 避免庞大组件出现， 4

3.2. 分段建设，避免大节点 4

3.3. 单一用途&模块化，小粒度化 4

3.4. 组合（Composition），而不是继承（inheritance） 4

3.5. 适当使用写 设计模式 4

3.6. 利用db和java语言 os操作系统等提供的扩展机制api 5

4. dsl业务脚本 工作流BPM 规则引擎 5

4.1. 提升语言级别与抽象DSL 5

4.2. 各种QL查询语言 各种EL表达式语言 5

4.3. 内嵌解释器 表达式解释器等 5

4.4. 脚本。脚本是扩展复杂功能的利器php python js等 5

4.5. 表映射表处理扩展if else 5

4.6. Sql脚本 sp view等 5

4.7. 频繁变动的业务配置化 脚本化 配置业务脚本 5

5. 通用接口参数化（查询和修改） 6

5.1. 类似于sql一样的通用接口 6

5.2. Socket接口 转rest 6

5.3. 类似usb接口那样通用 6

5.4. 接口转换器 adapter模式 6

6. 半结构化数据 与动态类型 6

6.1. 数据扩展 schema free , schema less模式 半结构化数据表示 6

6.2. 对于编译类型语言，多使用map等动态结构 6

6.3. 数据扩展性 数据库适当Json字段提升扩展性 7

7. 数据库系统提供的扩展性机制 7

7.1. 扩展Sql（带流程控制） xxsql Tsql Plsql等 7

7.2. View 于重写sql模式 7

7.3. Sp udf trigger（拦截器机制） 7

7.4. Db timer（mysql 里面叫event事件机制） 7

7.5. Json数据类型方便可扩展 7

7.6. 调用外部代码交互（mysql不直接支持） 7

7.7. Shell api （mysql不支持） 7

7.8. Other 7

8. 编程语言一般提供的扩展性机制 7

8.1. 脚本解释器 api 解释器(或脚本引擎scripting engine 7

8.2. Cli接口 Shell解释器 api 8

8.3. Sql解释器（通过sql驱动） 8

8.4. 正则与其他QL EL解释器 8

8.5. 通过其他dsl类库解释器模式等 8

8.6. 配置与元数据解释器api 8

8.7. 流程解释器 规则解释器 lib 8

8.8. Timer定时机制 事件机制 8

8.9. Vm拦截器机制过滤器机制 插件机制等（例如Java Agent ） 8

9. 利用os操作系统提供的扩展机制 9

9.1. Shell api 9

9.2. Timer机制（linux crontab，win 定时器等） 9

9.3. 拦截器机制 9

9.4. 其他（系统托盘通知等 9

10. 其他 10

10.1. 表格化 excel 10

10.2. 观察者模式 事件驱动 mq 10

11. 功能扩展 vs 性能扩展 10

12. Ref 10