链滴

Solo 架构理念和约定

作者：88250

原文链接：https://ld246.com/article/1537695161321

来源网站：链滴

本文是《Solo 从设计到实现》的一个章节，该系列文章将介绍 Solo 这款 Java 博客系统是如何从无有的，希望大家能通过它对 Solo 从设计到实现有个直观地了解、能为想参与贡献的人介绍清楚项目也希望能为给~~重复发明~~重新定义博客系统的人做个参考
eart

Latke 框架

之前写过，复用一下：《为什么又要造一个叫 Latke 的轮子》。

数据模型

在持久层，数据模型主要指的是持久化数据库的数据定义，也就是表结构。所有数据模型都是定义在 repository.json 文件中，初始化时会根据该文件自动生成建表 SQL。

在应用层，基于 JSON 的数据模型理论上是不用定义结构的，因为 JSON 本身是 schemaless 的。但了使用方便，我们还是定义了一些 model 类，但是这些 model 类的作用仅仅是为了归类字段，比如于文章表的字段都放到 article 类中，包括持久化字段和易变字段。

持久化字段对应的是数据库表的列，易变字段是在应用层、控制器层随时可能增减的字段。通过这样定，避免出现模型转换，提高开发效率。

当然，你可能会问，“这还不是一样要写类么，和写一个实体 POJO 有啥区别？”区别是有的，并且是设计理念（Philosophy）上的区别。一言以蔽之，使用 JSON 贯穿整个处理栈（请求-验证-计算-久化）可以简化很多处理，但类型化的数据模型（POJO）会成为快速开发的阻碍。

公共约定

主键都用 oId 表示，类型是字符串 UTC 时间戳。这样设计主要是考虑到：

1. 不使用数据库自增，避免迁移数据
2. 保持有序，有利于数据库性能
3. 实现简单

这也算是一个折中的方案，对于数据量不大的小型应用上已经足够。

关于配置

之前写过，复用一下：Latke 配置剖析。