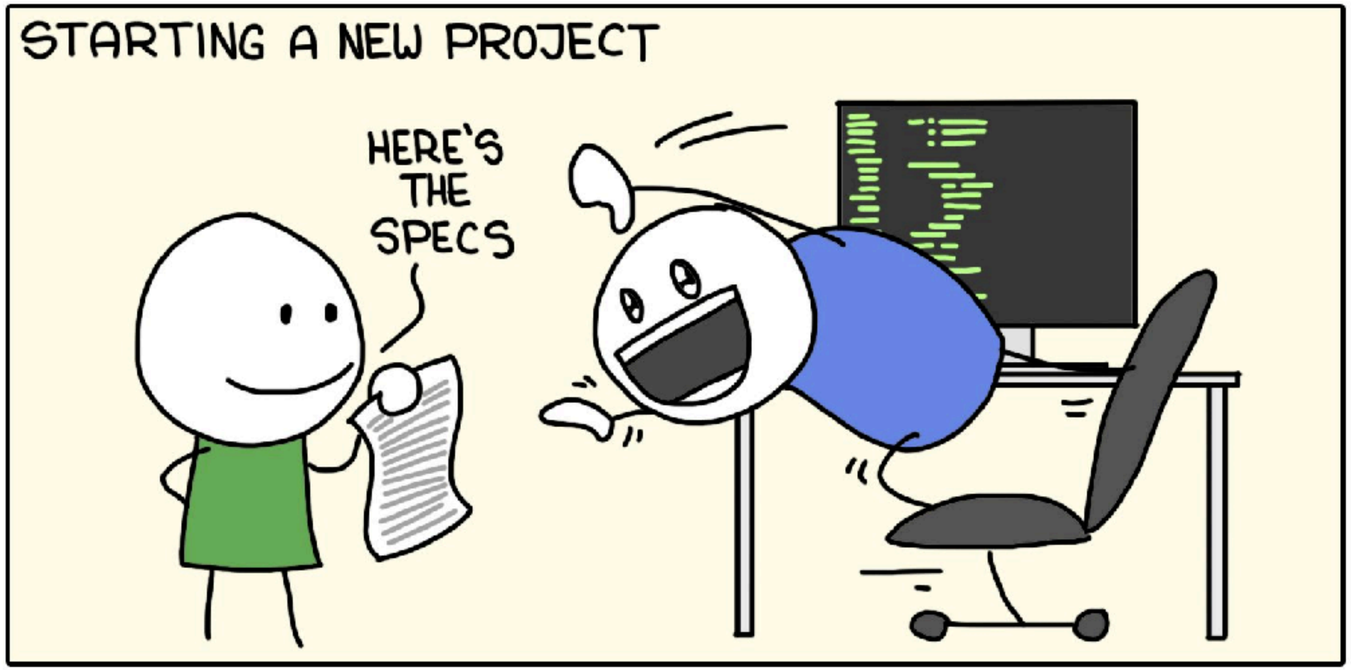
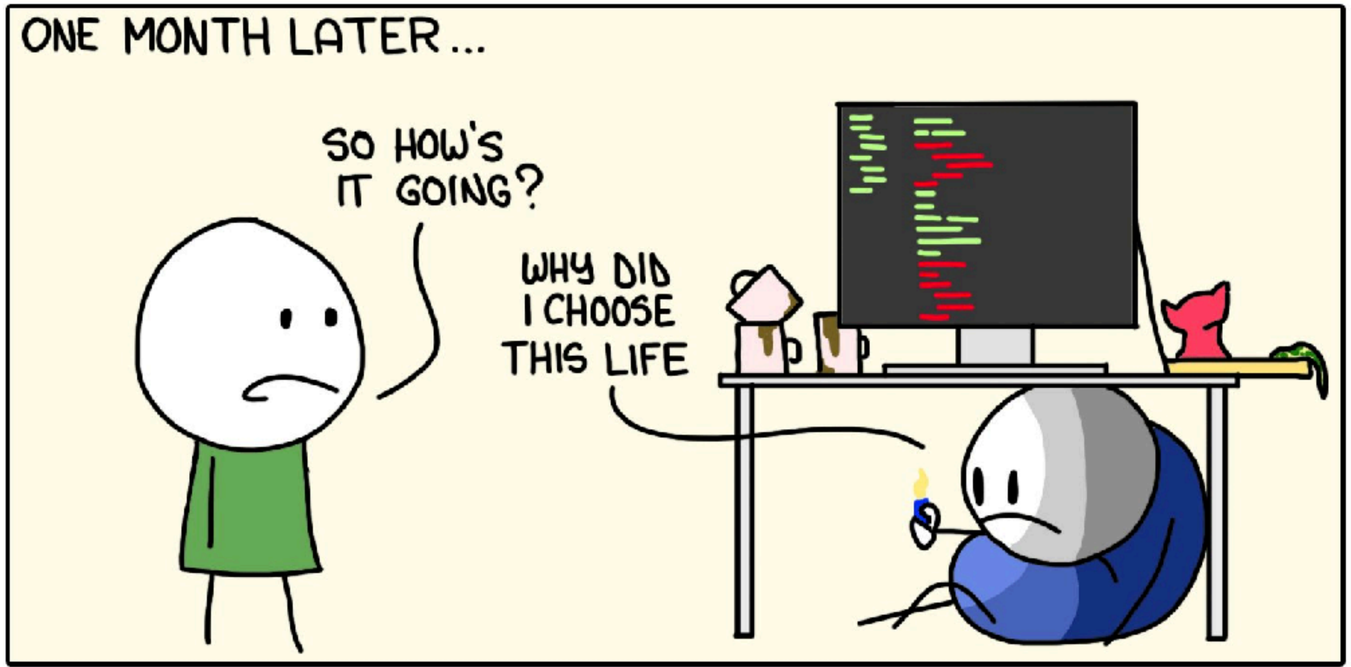
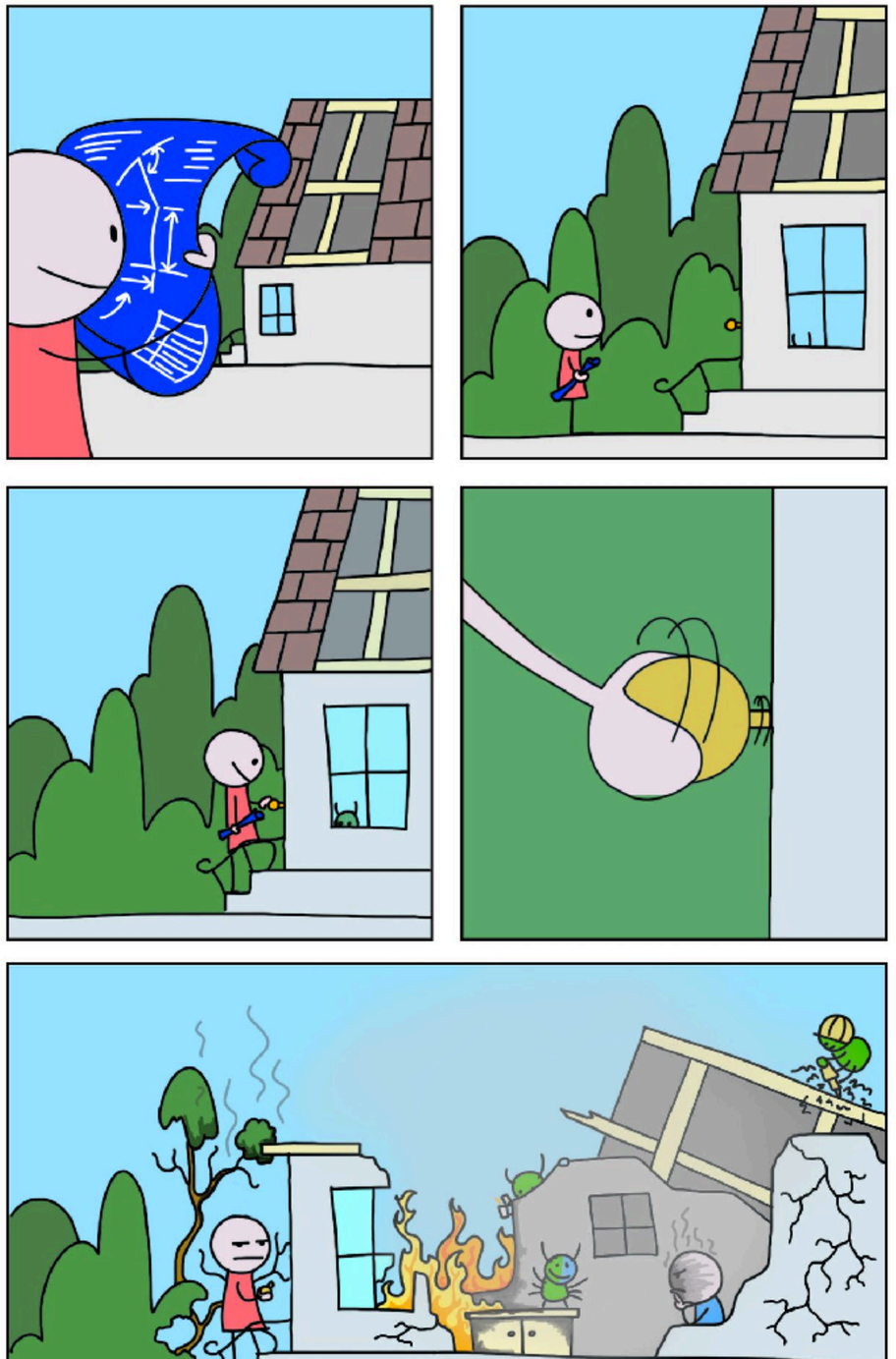
JAVA 达摩班

Spring Boot

#EVERYTIME


STARTING A NEW PROJECT
HERE'S THE SPECS

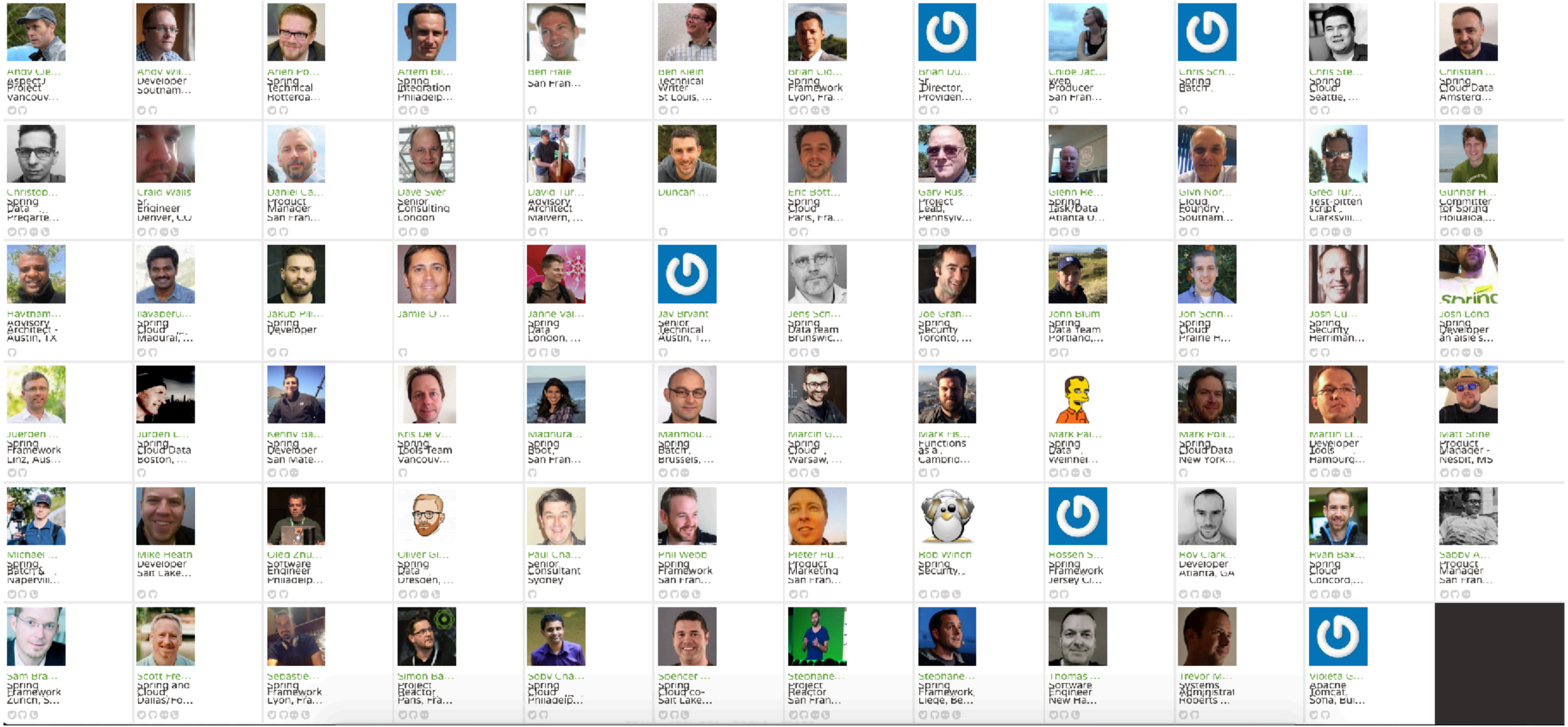

ONE MONTH LATER...
SO HOW'S IT GOING?
WHY DID I CHOOSE THIS LIFE


TESTING DURING DEVELOPMENT

Spring Boot 核心概念和架构 1

The Spring Team

https://spring.io/team



Spring Boot概念

Spring Boot由“The Spring Team (The Pivotal Team)”开发，是一个简化启动，开发和测试Spring应用的框架
Spring Boot将冗长的，常用的开发步骤，模板代码和配置封装为“默认配置”
方法论：Opinionated Defaults Configuration

	从零开发的新框架		基于Spring框架
Spring Boot不是		**而是**	
	用于解决新问题		用于解决和Spring同样的问题

Spring Boot = Spring框架 + 内嵌HTTP服务器 - XML/Annotation配置

Spring Boot概念

选用理由

- 简化java应用开发，单元测试和集成测试过程
- 提供默认配置，快速启动
- 提高生产力

优点：

- 非常容易的开发Spring应用，支持Java或Groovy
- 减少开发时间，提高生产力
- 避免写模版代码，注释和XML配置（每次重复）
- 方便和Spring生态系统集成，包括Spring JDBC，Spring ORM，Spring Data，Spring Security等
- 通过“Opinionated Defaults Configuration”方法降低开发成本
- 提供嵌入式HTTP服务器（Tomcat，Jetty等），便于开发和测试
- 提供CLI工具，通过命令行方式开发和测试Spring Boot应用
- 提供大量插件，更方便使用构建工具（Maven，Gradle）和内嵌的或内存数据库

Spring Boot实现

可以迁移任何类型项目到Spring Boot应用，但**迁移现有的Spring项目会比较困难**
Spring Boot**更适合开发全新项目**

Spring团队提供了三种创建SBA的方式
1. 使用Spring Boot CLI
2. 使用Spring STS IDE
3. 使用Spring Initializr Website

Spring boot支持两种风格的Spring应用：
A. Java应用
B. Groovy应用

Spring原生支持Groovy语言，可以用如上1，2，3开发Spring Boot Groovy应用；但只能使用方法2和3开发Spring Boot Java应用

扩展阅读

Spring Boot和Groovy

Groovy是一种JVM语言，一个项目可以同时使用Groovy和Java语言
后缀名为.groovy和.java的文件最终都会被编译成为字节码格式完全相同的.class文件

Spring Boot框架受到Groovy编程模型启发，内部使用到一些Groovy技术和工具实现默认导入和配置

Spring Boot架构

Spring Boot框架包含**四个核心**组件和**两个辅助**组成：

1. Spring Boot Starter
2. Spring Boot AutoConfigurator
3. Spring Boot CLI
4. Spring Boot Actuator

5. Spring Initilizr（https://start.spring.io/）
6. Spring Boot IDEs（Eclipse， Intellij IDEA，STS）

Spring Boot Starter

主要职责将一组通用的，相关的依赖组合成一个依赖
一般情况下，Spring应用需要引用大量依赖，而他们通常是固定搭配，不方便构建文件管理

优点：
1. 减少手动定义大量依赖
2. 简化构建依赖

以Spring Web应用为例，spring-boot-starter-web会自动下载所需依赖并添加到classpath

Web应用 == + Spring core Jar file(spring-core-xx.jar)
+ Spring Web Jar file(spring-web-xx.jar)
+ Spring Web MVC Jar file(spring-webmvc-xx.jar)
+ Servlet Jar file(servlet-xx.jar)
+ Spring JDBC Jar file(spring-jdbc-xx.jar)
+ Spring ORM Jar file(spring-orm-xx.jar)
+ Spring Transaction Jar file(spring-transaction-xx.jar)
+ HTTP server（tomcat）

<==ALL-IN-ONE== Spring-boot-starter-web

Spring Boot AutoConfigurator

Spring的槽点经常在“开发spring应用需要繁杂的配置（xml或注释）”
AutoConfigurator的职责就是减少或不需要spring配置，它负责注入配置信息

再次以Spring MVC为例，spring-boot-starter-web会自动启动AutoConfigurator，自动生成XML/注释信息实现views，view resolvers等的配置
@SpringBootApplication = @Configuration + @ComponentScan + @EnableAutoConfiration

Spring Boot CLI

Spring Boot CLI是用于运行和测试Spring Boot应用的命令行软件
使用CLI启动SBA，它内部使用Spring Boot Starter和Spring Boot AutoConfigurator解析依赖并启动应用
例如： `spring run HelloWorld.groovy`

Spring Boot Actuator

主要包含两个功能：
1. 提供SBA的管理终端（Management Endpoint）
2. 提供SBA度量（Metrics）

当使用CLI启动SBWA，它自动提供localhost域名和8080端口的访问终端
管理终端以HTTP请求（GET，POST）的形式提供

扩展阅读

Spring Boot内部机制

Groovy思想：不需要添加imports，不需要添加dependencies，Groovy编译器（groovyc）在编译groovy脚本的时候，会首先添加默认导入语句，然后开始编译

Groovy包含JAR依赖解析器，用于解析和添加所需要的jar包到项目classpath
Spring Boot**内部使用Groovy添加默认配置**，例如Default import statements，main()方法等

Grape是内嵌的Dependency Resolution Engine（DRE），即依赖解析引擎，本质就是内嵌在Groovy中的Jar包管理器
Spring Boot框架的编程模型受到Groovy编程模型启发，其内部天然依赖两大组件：Groovy和Grape

Spring Boot = (Groovy + Grape) + Spring框架

Spring Boot 应用实践 2

Spring Boot实践

@Value注释用于给字段插入值，可以来自Bean，也可以来自properties文件

类文件

```
@Value(value = "what's up")
private String title;

@Value(value = "${dharma.unknown:default}")
private String unknown;

@Value(value = "${dharma.team}")
private String[] team;

@Value(value = "#{admin.name}")
private String admin;
```

Properties文件：

```
dharma.team=tom, jerry, ashton
```

Spring Boot实践

Spring boot提供丰富外化配置的方案，这样同样的代码可以运行在不同的环境下。配置可以存在**properties文件，YAML文件，环境变量和命令行参数**四种方式。

application.properties文件是最常用的方式之一，它的加载优先级从高到低依次为：

1. 项目根目录的/config子目录下
2. 项目根目录
3. classpath（resources）目录下的/config包
4. classpath（resources）目录

YAML是JSON的超集，用于指定结构化的配置数据，spring-boot-starter自动安装Snake YAML，所以它会自动加载classpath下的YAML文件。YAML可以看作properties的替代，如果在同目录下application.properties没有找到，会去寻找application.yaml文件。

Spring Boot实践

```
dharma.randomInt=${random.int}
dharma.randomString: ${random.value}
dharma.project=www.dharma-mall.resources.config

server.port=8080
```

vs

```
dharma:
  randomInt: ${random.int}-yaml
  randomString: ${random.value}-yaml
  project: www.dharma-mall.resources.config.yaml

server:
  port: 8080
```

Spring Boot实践

```
dharma.randomInt=${random.int}
dharma.randomString: ${random.value}
dharma.project=www.dharma-mall.resources.config

server.port=8080
```

vs

```
dharma:
  randomInt: ${random.int}-yaml
  randomString: ${random.value}-yaml
  project: www.dharma-mall.resources.config.yaml

server:
  port: 8080
```

dev，test和prod环境变量定义在同一个配置文件
通过active指定当前环境

application-dev.yaml

```
spring:
  profiles:
    active: dev

server:
  port: 8888

---
spring:
  profiles: dev
server:
  port: 8080

---
spring:
  profiles: prod
server:
  port: 8082

---
spring:
  profiles: test
server:
  port: 8081
```

dev，test和prod环境变量定义在同不同配置文件
通过active指定当前环境

application.yaml

```
spring:
  profiles:
    active: dev

server:
  port: 8888
```

application-dev.yaml

```
spring:
  profiles: dev
server:
  port: 8080
```

application-prod.yaml

```
spring:
  profiles: prod
server:
  port: 8082
```

application-test.yaml

```
spring:
  profiles: test
server:
  port: 8081
```

MySQL 安装与使用 3

MySQL

MySQL是最流行的**关系数据库管理系统（ORDBMS）**之一，也是应用于Web开发，适用于中小企业的最常用的数据库管理系统。

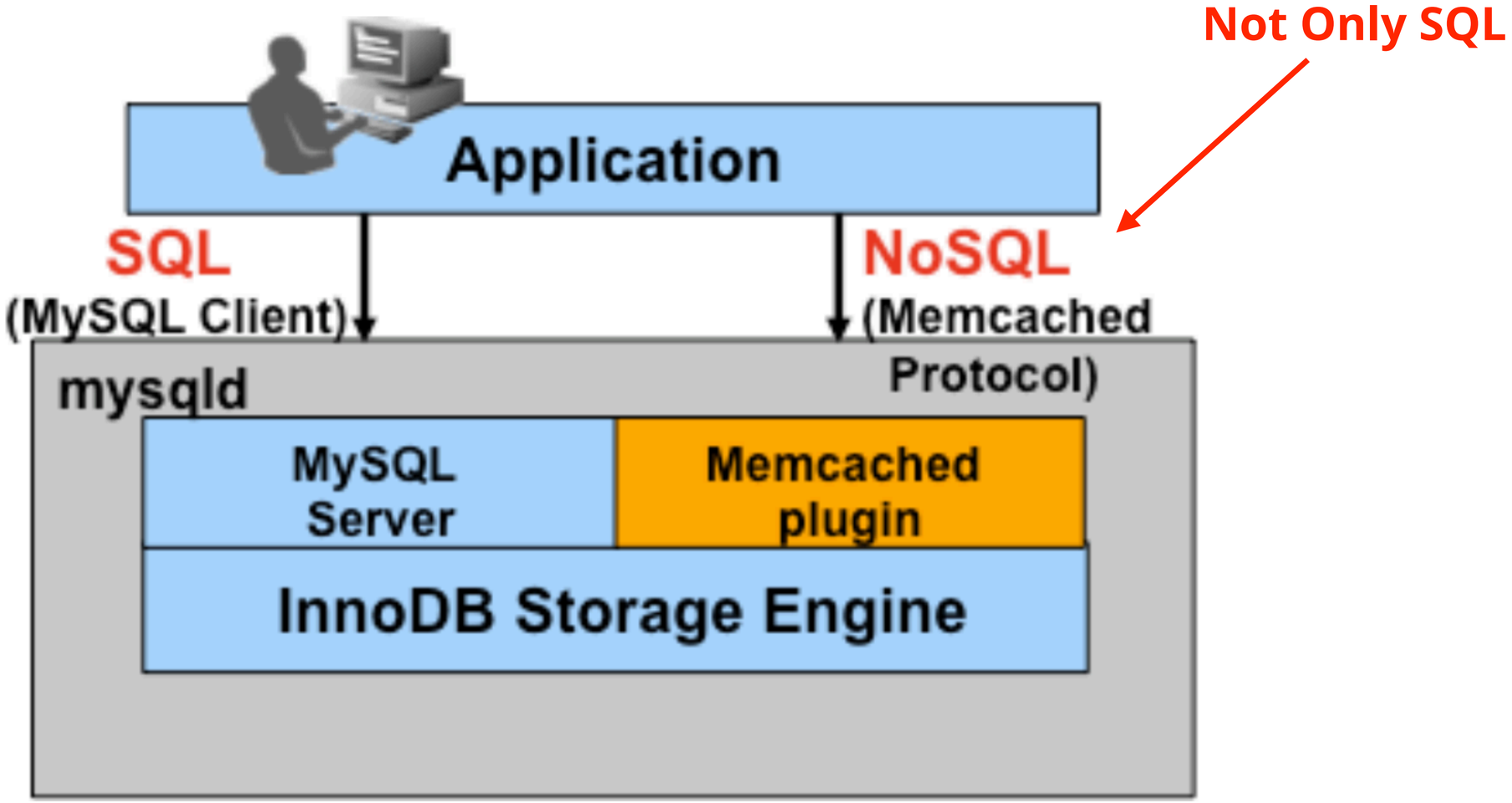
MySQL最早由MySQL AB公司（成立于1995年瑞典）开发，2008年被Sun Microsystem收购，后者又被Oracle于2010年收购。

MySQL提供多个版本

- MySQL Community Edition：社区版，免费，开源
- MySQL Standard Edition：标准版，商业版，
- MySQL Enterprise Edition：企业版，商业版
- MySQL Cluster CGE：专注分布式，高可用性的企业版，商业版，最贵
- MySQL OEM: 用于OEM厂商

https://www.mysql.com/products/

MySQL架构[实验]


Not Only SQL
Application
SQL
(MySQL Client)
NoSQL
(Memcached
Protocol)
mysqld
MySQL
Server
Memcached
plugin
InnoDB Storage Engine

MySQL

343 systems in ranking, June 2018

Rank			DBMS	Database Model	Score		
Jun 2018	May 2018	Jun 2017			Jun 2018	May 2018	Jun 2017
1.	1.	1.	Oracle	Relational DBMS	1311.25	+20.84	-40.51
2.	2.	2.	MySQL	Relational DBMS	1233.69	+10.35	-111.62
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1087.73	+1.89	-111.23
4.	4.	4.	PostgreSQL	Relational DBMS	410.67	+9.77	+42.13
5.	5.	5.	MongoDB	Document store	343.79	+1.67	+8.79
6.	6.	6.	DB2	Relational DBMS	185.64	+0.03	-1.86
7.	7.	↑ 9.	Redis	Key-value store	136.30	+0.95	+17.42
8.	↑ 9.	↑ 11.	Elasticsearch	Search engine	131.04	+0.60	+19.48
9.	↓ 8.	↓ 7.	Microsoft Access	Relational DBMS	130.99	-2.12	+4.44
10.	10.	↓ 8.	Cassandra	Wide column store	119.21	+1.38	-4.91
11.	11.	↓ 10.	SQLite	Relational DBMS	114.26	-1.19	-2.44
12.	12.	12.	Teradata	Relational DBMS	75.77	+1.36	-1.55
13.	↑ 14.	↑ 18.	MariaDB	Relational DBMS	65.85	+0.85	+12.95
14.	↓ 13.	↑ 16.	Splunk	Search engine	65.78	+0.68	+8.26
15.	15.	↓ 14.	Solr	Search engine	62.06	+0.55	-1.55
16.	16.	↓ 13.	SAP Adaptive Server	Relational DBMS	61.49	-0.02	-6.04
17.	17.	↓ 15.	HBase	Wide column store	59.70	-0.25	-2.17
18.	18.	↑ 20.	Hive	Relational DBMS	57.33	+0.36	+12.95
19.	19.	↓ 17.	FileMaker	Relational DBMS	56.18	+1.51	-0.90
20.	20.	↓ 19.	SAP HANA	Relational DBMS	49.35	+0.97	+1.85

MySQL

MySQL和Oracle是并驾齐驱的

- 因Web而生，诞生于互联网时代（Oracle Mysql团队参与制定Web基础架构）
- 数据库先行者，形成行业标准（尤其被Oracle收购后）
- 用户熟知度高，大到Fackbook，Youtube，Twitter，Wikipedia，小到初学者和学生，固定搭配（LAMP），方便搜索问题（曾经出现过）
- 低成本实现高速度，高扩展性，高可用性，低出错率，低DBA排错时间
- InnoDB成为5.5开始后的默认引擎，数据备份，快速链接（连接池），充分利用内存（schema）和缓存（查询），自动数据切分
- 基于C和C++实现
- 既可以所谓服务单独运行（B/S），也可以嵌入到应用中（运行在无网络环境），也可以用在SaaS和云端（Amazon RDS）
- 容易安装，可以运行在任何平台（系统），零配置
- 多语种支持（错误提示），时区切换
- 多版本满足不同需求，Cluster专注事务数据，Enterprise专注大型应用管理和开发

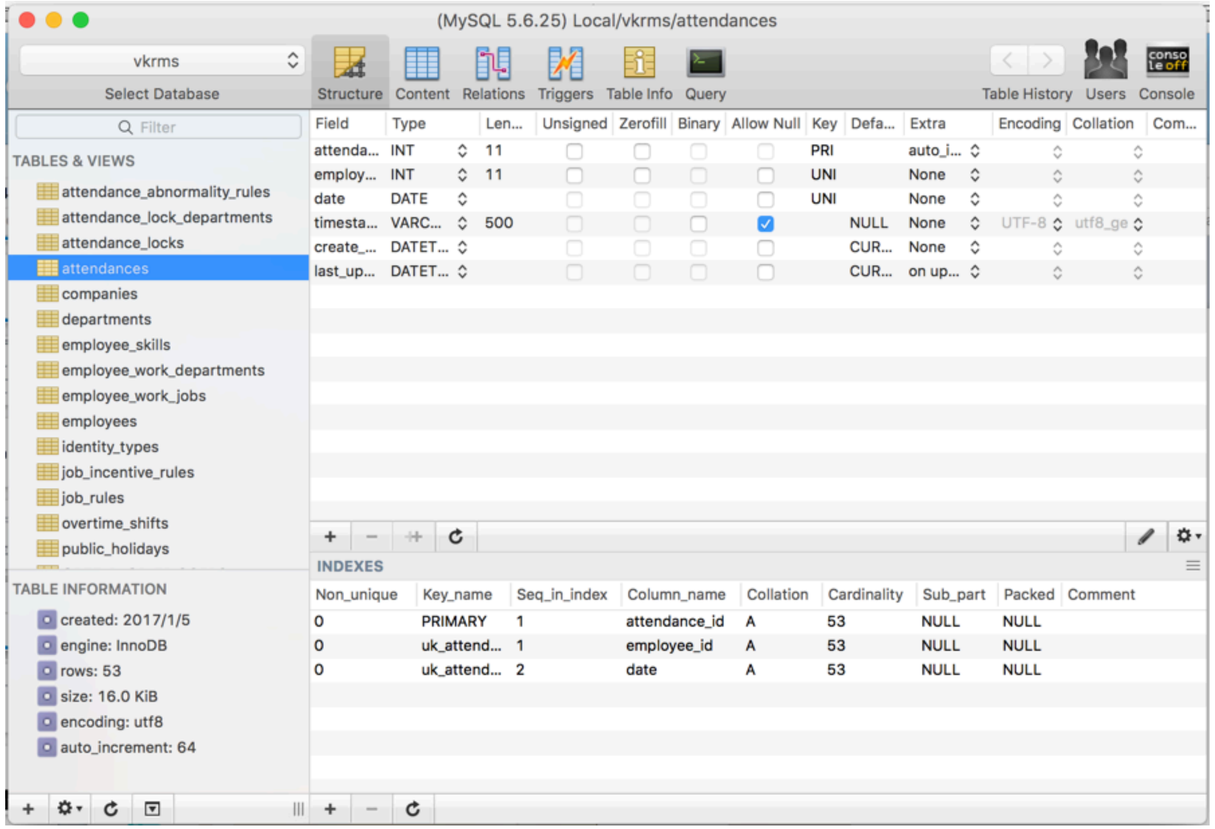
- 逐渐失去免费和开源“灵魂”，不再社区驱动

MySQL 安装

数据库安装：

1. 图形化安装包：MySQL Community Server
2. 包管理工具：brew install mysql

GUI管理界面：Sequel Pro



MySQL

启动/关闭服务

sudo /usr/local/mysql/support-files/mysql.server start
sudo /usr/local/mysql/support-files/mysql.server stop

(如果使用brew安装，以上MySQL路径改为/usr/local/Cellar/mysql/5.7.22)

```
┌─[xhji@GoAshton] - [/usr/local/mysql/support-files] - [2018-06-03 12:28:41]
└─[0] <git:(master 3469f177f✗*) > sudo /usr/local/mysql/support-files/mysql.server stop
Password:
Shutting down MySQL
.... SUCCESS!
┌─[xhji@GoAshton] - [/usr/local/mysql/support-files] - [2018-06-03 12:29:14]
└─[0] <git:(master 3469f177f✗*) > sudo /usr/local/mysql/support-files/mysql.server start
Starting MySQL
. SUCCESS!
```

MySQL

监控器 - 命令行

#输入root用户密码，进入控制台
sudo /usr/local/mysql/bin/mysql -u root -p

#列出数据库
show databases;
#使用dharma_mall_app数据库
use dharma_mall_app;
#列出所有表
show tables;
#列出表结构
describe product;

#退出控制台
\q

```
└─[0] <git:(master 3469f177f✗*) > sudo /usr/local/mysql/bin/mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.6.25 MySQL Community Server (GPL)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| dharma_mall_app    |
| lunchBox           |
| mysql              |
| performance_schema |
| supervisor         |
| test               |
| vkrms              |
+--------------------+
8 rows in set (0.00 sec)

mysql> use dharma_mall_app;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+---------------------------+
| Tables_in_dharma_mall_app |
+---------------------------+
| hibernate_sequence        |
| product                   |
+---------------------------+
2 rows in set (0.00 sec)

mysql> describe product;
+-------+------------------+------+-----+---------+----------------+
| Field | Type             | Null | Key | Default | Extra          |
+-------+------------------+------+-----+---------+----------------+
| id    | int(11) unsigned | NO   | PRI | NULL    | auto_increment |
| name  | varchar(50)      | NO   |     |         |                |
| price | double           | NO   |     | NULL    |                |
+-------+------------------+------+-----+---------+----------------+
3 rows in set (0.01 sec)

mysql> \q
Bye
```

Thanks!

Any questions?