

# 法律声明

---

□ 本课件包括：演示文稿，示例，代码，题库，视频和声音等，讲师及小象学院拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利。

□ 课程详情请咨询

■ 微信公众号：小象

■ 新浪微博：ChinaHadoop



---

# Kubernetes代码解读



# 目录

---

1. 开发环境搭建
2. 源码结构
3. 构建/测试等
4. Apiserver基础

# 1. 开发环境搭建

---

1. <https://gitforwindows.org/> windows下的git
2. <https://golang.google.cn/> Go语言国内唯一官方站点
3. <https://www.jetbrains.com/idea/download> 神器IDE
4. git最简教程 <http://www.runoob.com/git/git-tutorial.html>
5. Go最好的教科书 <https://github.com/golang-china/gopl-zh>

# 1. Git for windows

---

- 基本上为一路点击下一步
- 安装完后在windows下拥有了bash和git两大方便工具

```
$ cd
```

```
$ cat .gitconfig
```

```
[user]
```

```
email = hardysimpson1984@gmail.com
```

```
name = Hardy Simpson
```

# 1. Go环境安装

---

□ <https://golang.google.cn/dl>

```
$ cd
```

```
$ mkdir -p go/src/hello
```

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
    fmt.Printf("hello, world\n")
```

```
}
```

```
$ go build
```

```
$ ./hello.exe
```

```
hello, world
```

# 1. GOROOT 和 GOPATH

---

□ <https://www.jianshu.com/p/4e699ff478a5>

特性	操作	路径
import “aa/bb”	读	\$GOPATH/src/aa/bb.go
go get	写	\$GOPATH/src
go install	写	\$GOPATH/bin
go build	写	\$GOPATH/pkg/*.a

# 1. 下载k8s代码

---

# 注意路径

```
$ mkdir -p ~/go/src/k8s.io/
```

```
$ cd ~/go/src/k8s.io/
```

```
$ git clone https://github.com/kubernetes/kubernetes.git
```

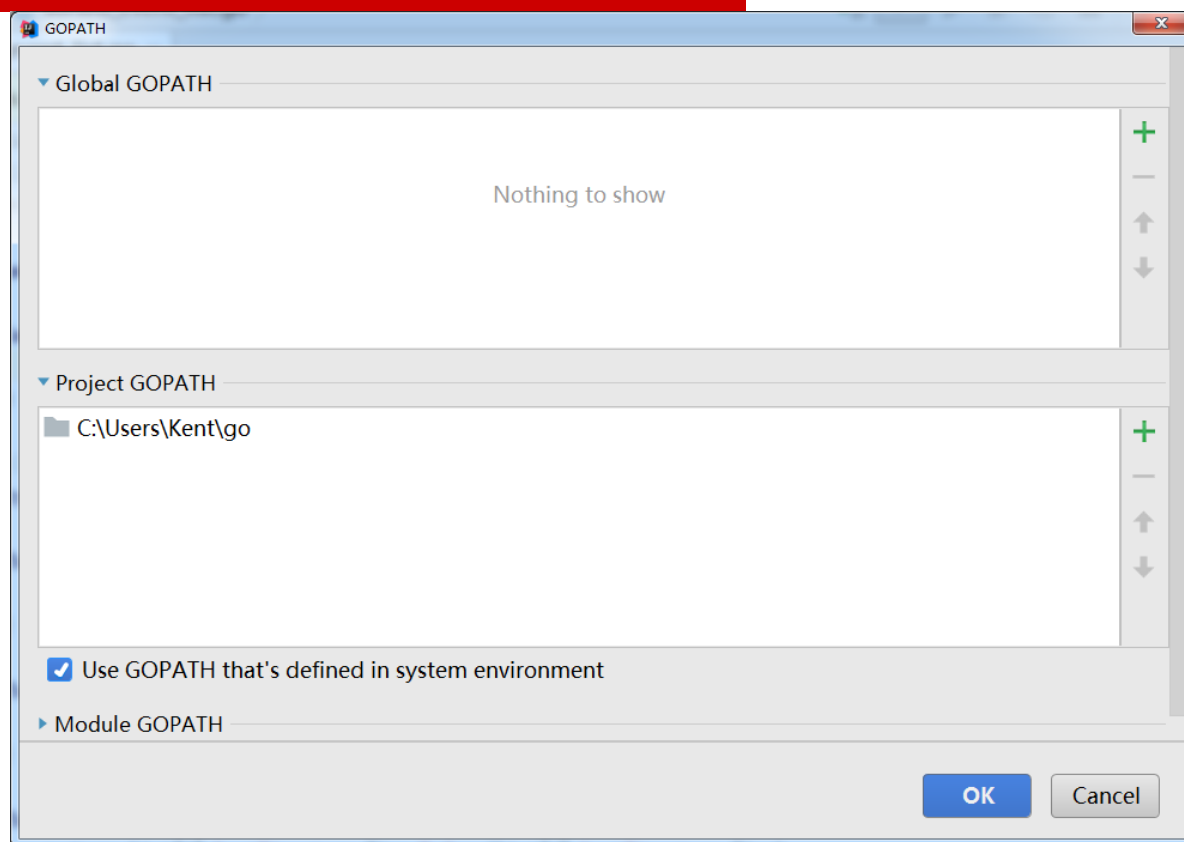


# 1. IDEA安装

---

- ❑ Version 2017.1 为最后一个支持社区Go版的
- ❑ 下载地址 <https://download.jetbrains.com/idea/ideaIC-2017.1.6.exe>
- ❑ 使用360/QQ 安装JDK (oracle sucks)
- ❑ 基本为一路点击下一步, community版足够了
- ❑ File | Settings | Plugin
- ❑ 必装Plugin
  - Go lang
- ❑ 选装plugin
  - Vim

# 1. IDEA项目路径设置



- ❑ GOPATH的路径和前面的项目路径对应，决定了源代码里面的import是否有检查和自动补齐

## 2. 源码结构

路径	说明
api	输出接口文档用
build	构建脚本
cluster	适配不同I层的云，例如亚马逊AWS，微软Azure，谷歌GCE的集群启动脚本
cmd	所有的二进制可执行文件入口代码，例如apiserver/scheduler/kubelet
contrib	项目贡献者
docs	文档，包括了用户文档、管理员文档、设计、新功能提议
example	使用案例
Godeps	项目中依赖使用的Go第三方包，例如docker客户端SDK，rest等

## 2. 源码结构

路径	说明
hack	工具箱，各种编译、构建、测试、校验的脚本都在这里
hooks	git提交前后触发的脚本
<b>pkg</b>	<b>项目代码主目录，cmd的只是个入口，这里是所有的具体实现</b>
plugin	插件
release	应该是Google发版本用的
test	测试相关的工具
third_party	一些第三方工具，应该不是强依赖的
vendor	第三方依赖库，Go规范，详见 <a href="#">《理解go vendor》</a>

### 3. 构建

---

- 要构建在本地运行的二进制程序：

```
$ sudo make # _output/local/go/bin
```

- 清理

```
$ sudo make clean
```

- 要生成一个版本：

```
$ make release
```

- 要生成一个版本而不运行测试：

```
$ make release-skip-tests
```

### 3. 测试

---

□ 单元测试:

```
$ sudo make test
```

□ 仅运行 pkg/api 测试:

```
$ sudo WHAT=../k8s.io/kubernetes/pkg/api make test
```

□ 集成测试:

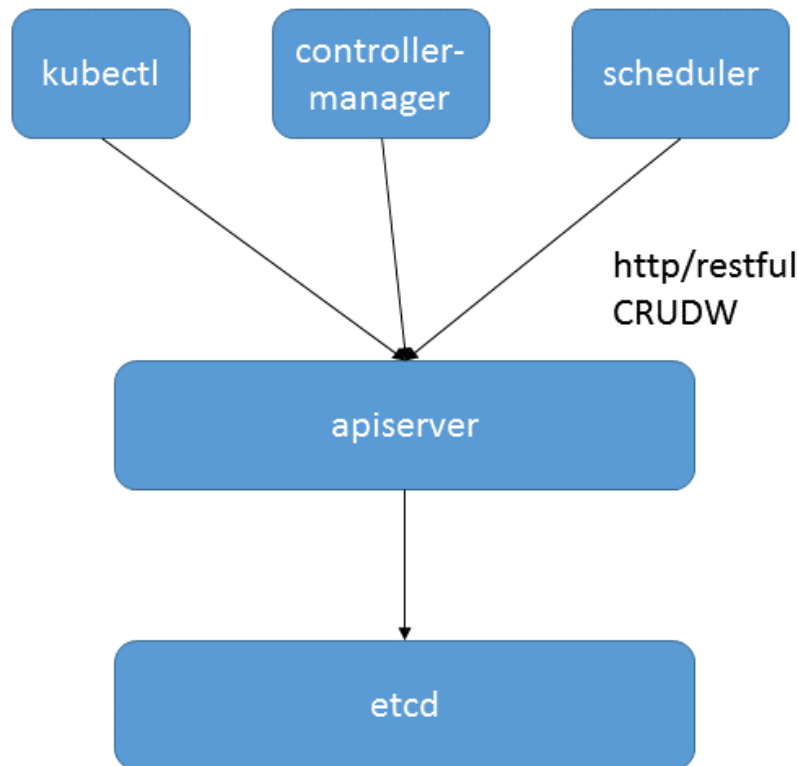
```
$ sudo make test-integration
```

## 4. 代码阅读

---

- ❑ 遍历pkg下所有的目录，概览大概知道这个目录是干啥的
- ❑ 从cmd这个入口来看apiserver的代码，然后一点点由浅入深，看apiserver的大致实现
- ❑ 分特性，看具体某个大的特性是怎么实现的，例如安全，例如和etcd存储对接
- ❑ 在上面这几步的过程中可以看看别人的代码阅读文档，能有效的节省时间

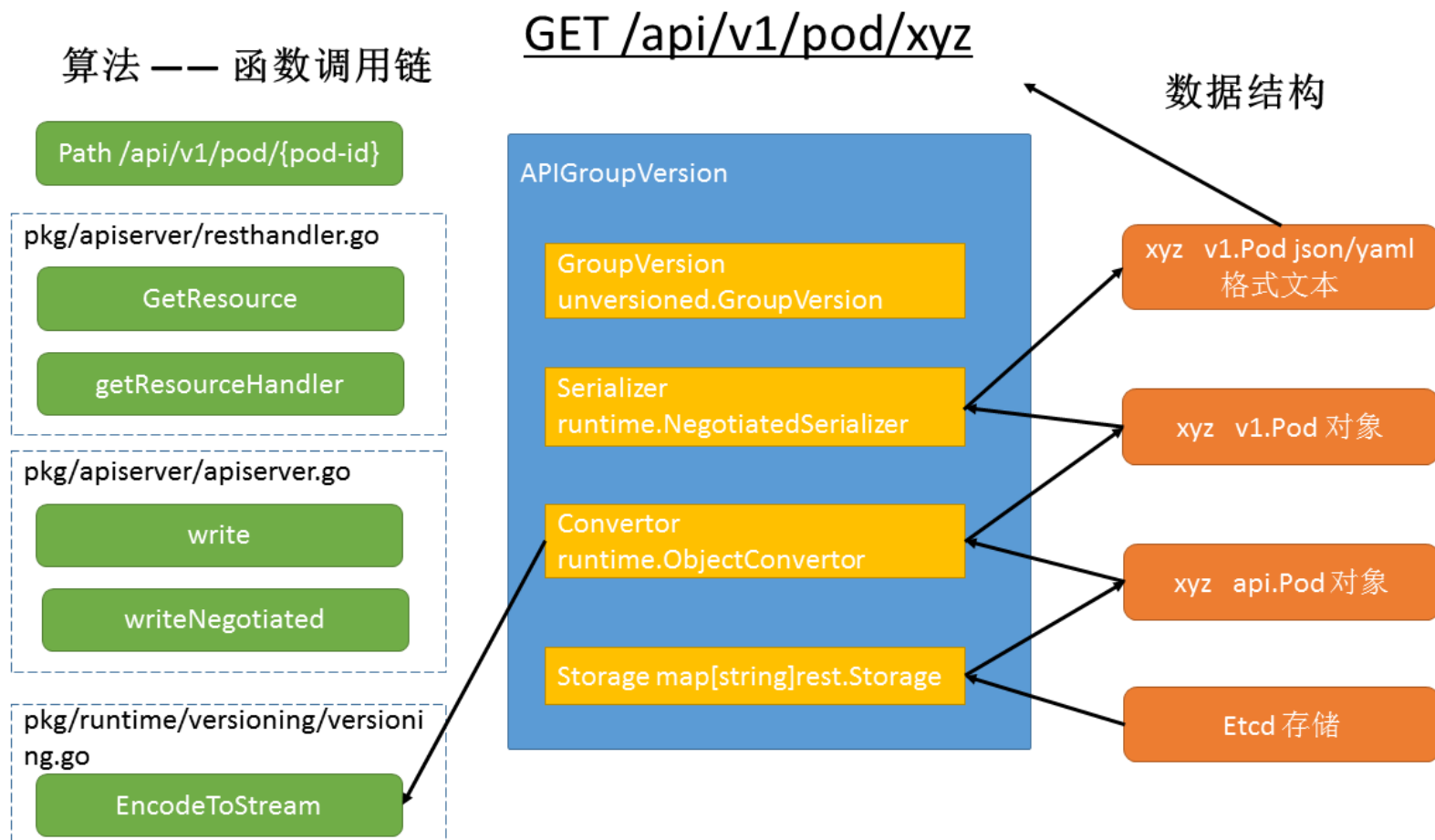
## 4. apiserver基础



- ❑ apiserver是k8s系统中所有对象的增删查改的http/restful式服务端，其中W是指watch操作。数据最终存储在分布式一致的etcd存储内，apiserver本身是无状态的，提供了这些数据访问的认证鉴权、缓存、api版本适配转换等一系列功能。
- ❑ 对于http服务和使用go语言实现方式，可以看go-restful的[文档](#)和[例子](#)，对这个有基本的了解，这个文档对入门者和一知半解者极为有效！



## 4. apiserver基础

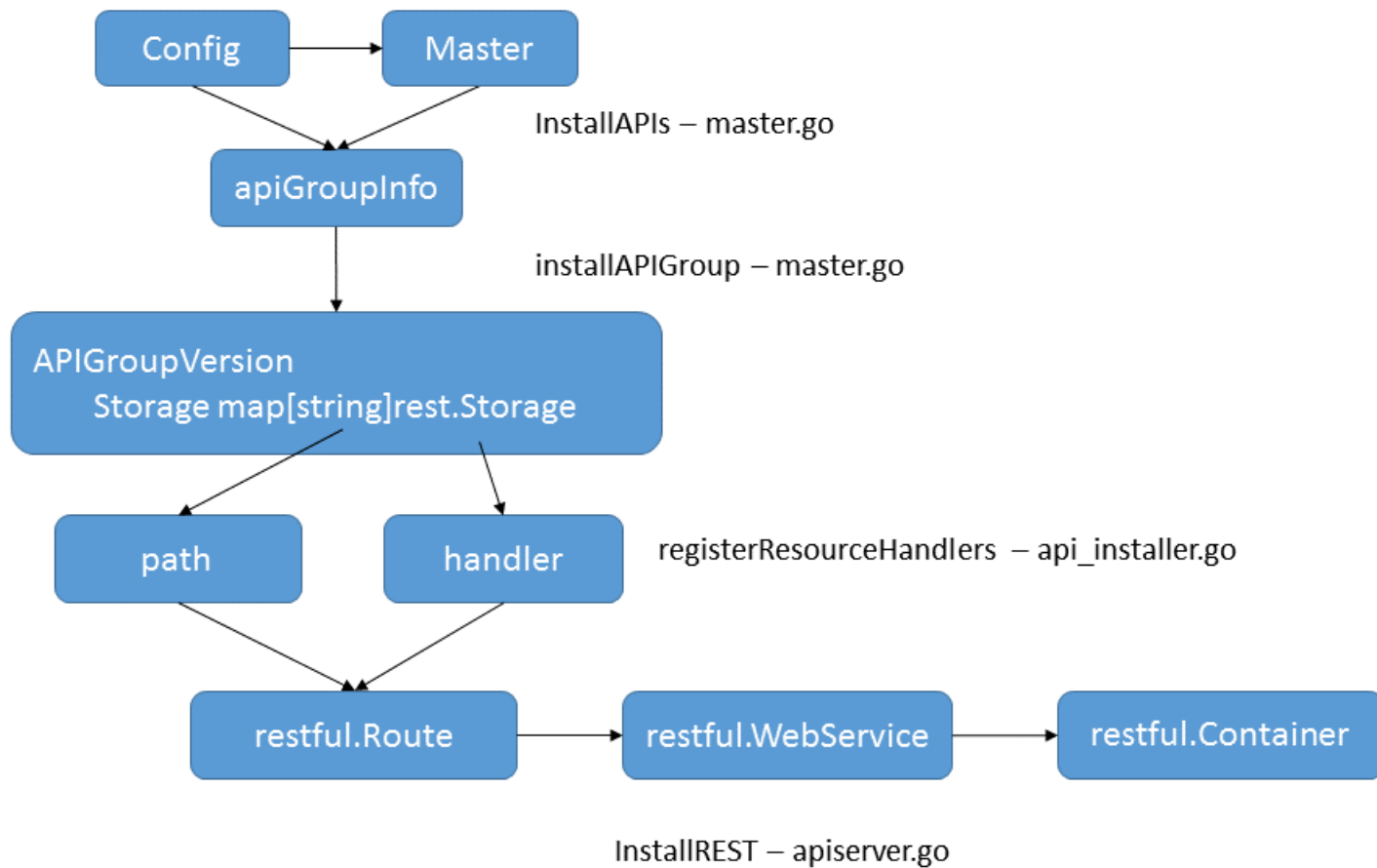


## 4. apiserver基础

---

成员	作用
GroupVersion	包含 api/v1 这样的string，用于标识这个实例
Serializer	对象序列化和反序列化器
Converter	这是一个强大的数据结构，这里放的是个接口，本体在 /pkg/conversion/conversion.go，几乎可以转换任意一种对象到另一种，只要你事先注入了相应的转换函数
Storage	这个map的key，用于对象的url，value是一个rest.Storage结构，用于对接etcd存储，在初始化注册时，会把这个map化开，化为真正的rest服务到存储的一条龙服务

## 4. apiserver基础

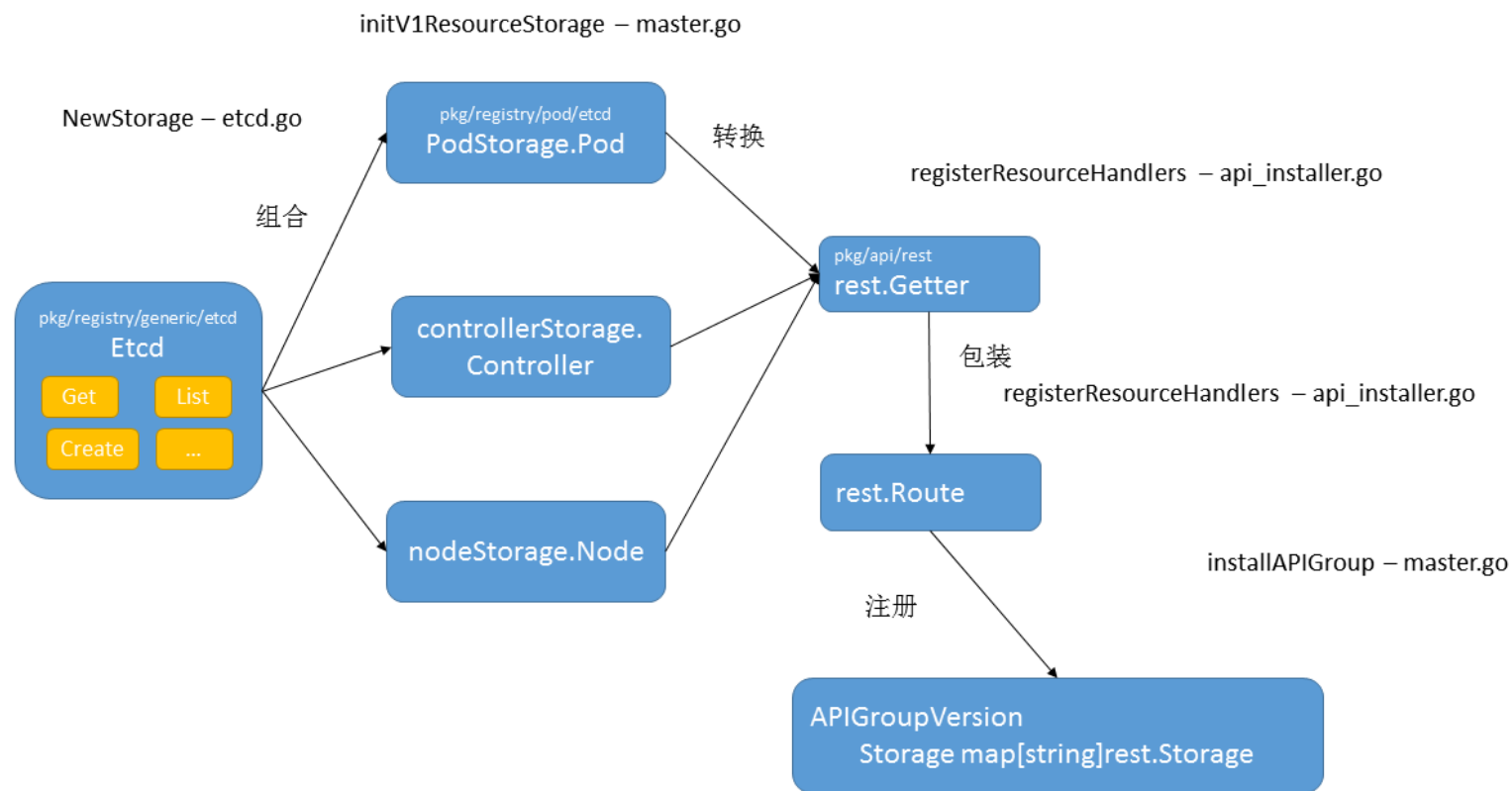


## 4. apiserver基础

---

- ❑ <https://kubernetes.io/docs/reference/api-overview>
- ❑ 核心组，REST路径在 /api/v1，但这个路径不是固定的，v1是当前的版本。与之相对应的代码里面的 apiVersion 字段的值是v1。
- ❑ 扩展组，REST路径在 /apis/extensions/\$VERSION，相对应的代码里面的 apiVersion: extensions/\$VERSION (例如当前的apiVersion: extensions/v1beta1)。这里提供的API对象今后有可能会被移动到别的组内。
- ❑ [CRD](#)，用户自定义资源

## 4. apiserver基础



□ 详见博文 [《kubernetes代码阅读-apiserver基础篇》](#)

# 作业

---

☐ 安装环境，看代码

# 联系我们

---

## 小象学院：互联网新技术在线教育领航者

- 微信公众号：大数据分析挖掘
- 新浪微博：ChinaHadoop

