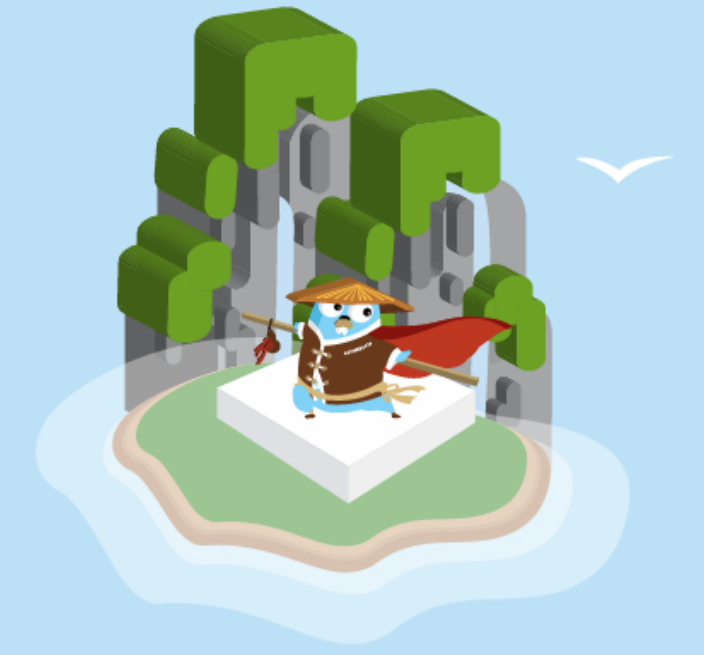


GopherChina2018



Bazel build //:Go

yuan@liulishuo.com





流利说

Help everyone become a global
citizen!

[github/lingochamp](https://github.com/lingochamp)



何源

yuan@liulishuo.com

2014 年底加入流利说， 目前主要负责 Platform Team

来流利说之前在 ThePlant 杭州工作

Agenda

- Package Management
- Code Management (Multi languages)
- Bazel build `//:Go`
- Demo
- Q & A



流利说 ❤️ GopherChina2018

Let's talk about

Package Management

vendor

Go **1.5** introduced experimental support for a "**vendor**" directory,
enabled by the **GO15VENDOREXPERIMENT** environment variable.

Go 1.6 enabled this behavior by **default**, and in Go **1.7**,
this switch has been removed and the "vendor" behavior is

ALWAYS enabled.



流利说 ❤️ GopherChina2018

vendor 很棒! 棒?

@2015 – 2016?


```
$ cd $GOPATH/src/upspin.io && tree vendor/
```



流利说 ❤️ GopherChina2018

```
vendor
├── bazil.org
│   └── fuse
│       ├── LICENSE
│       ├── README.md
│       ├── buffer.go
│       ├── debug.go
│       ├── error_darwin.go
│       ├── error_freebsd.go
│       ├── error_linux.go
│       ├── error_std.go
│       ├── fs
│       │   ├── serve.go
│       │   └── tree.go
│       ├── fuse.go
│       ├── fuse_darwin.go
│       ├── fuse_freebsd.go
│       ├── fuse_kernel.go
│       ├── fuse_kernel_darwin.go
│       ├── fuse_kernel_freebsd.go
│       ├── fuse_kernel_linux.go
│       ├── fuse_kernel_std.go
│       ├── fuse_linux.go
│       ├── fuseutil
│       │   └── fuseutil.go
│       ├── mount.go
│       ├── mount_darwin.go
│       └── mount_freebsd.go
├── blackfriday
│   ├── LICENSE.txt
│   ├── README.md
│   ├── block.go
│   ├── doc.go
│   ├── html.go
│   ├── inline.go
│   ├── latex.go
│   ├── markdown.go
│   └── smartypanths.go
├── github.com
│   ├── NYTimes
│   │   ├── README.md
│   │   ├── CONTRIBUTING.md
│   │   ├── LICENSE.md
│   │   ├── README.md
│   │   ├── gzip.go
│   │   └── gzip_go18.go
│   ├── golang
│   │   └── protobuf
│   │       ├── AUTHORS
│   │       ├── CONTRIBUTORS
│   │       ├── LICENSE
│   │       ├── Make.protobuf
│   │       ├── Makefile
│   │       ├── README.md
│   │       └── proto
│   │           ├── Makefile
│   │           └── clone.go
│   └── NYTimes
│       ├── README.md
│       ├── CONTRIBUTING.md
│       ├── LICENSE.md
│       ├── README.md
│       ├── gzip.go
│       └── gzip_go18.go
├── golang.org
│   └── x
│       └── crypto
│           ├── AUTHORS
│           ├── CONTRIBUTING.md
│           ├── CONTRIBUTORS
│           ├── LICENSE
│           ├── PATENTS
│           ├── README
│           ├── acme
│           │   ├── acme.go
│           │   └── autocert
│           │       ├── autocert.go
│           │       ├── cache.go
│           │       └── listener.go
│           └── renewal.go
│               ├── jws.go
│               └── types.go
├── codereview.cfg
└── htdf
```

IT WORKS.

Russ Cox 有话要讲

A Proposal for Package Versioning in Go

– Russ Cox

Russ Cox 说:

- Rust 的 Cargo 做的不错，于是我们做了 Dep
- 八年的 go install 和 go get 辛苦大家了， go module 了解一下
- go module 会干掉大部分的 vendor 目录



Russ Cox 说:

||

the new concept of a **Go module**, which is a collection of **packages versioned** as a unit; **verifiable and verified builds**; and version-awareness throughout the go command, enabling **work** outside **\$GOPATH** and the **elimination** of (most) **vendor** directories.



流利说 ❤️ GopherChina2018

vgo

Go & Versioning

Posted on Tuesday, February 20, 2018.

These are the posts in the “Go & Versioning” series.

- “[Go += Package Versioning](#)” [PDF].
- “[A Tour of Versioned Go \(vgo\)](#)” [PDF].
- “[Semantic Import Versioning](#)” [PDF].
- “[Minimal Version Selection](#)” [PDF].
- “[Reproducible, Verifiable, Verified Builds](#)” [PDF].
- “[Defining Go Modules](#)” [PDF].
- “[Versioned Go Commands](#)” [PDF].

The official Go proposal is at <https://golang.org/issue/24301>.

1. Packages Versioned
2. Verifiable and verified builds
3. Work outside \$GOPATH

proposal: add package version support to Go toolchain

New issue

#24301

Open rsc opened this issue on 8 Mar · 106 comments



rsc commented on 8 Mar • edited

Contributor +

proposal: add package version support to Go toolchain

It is long past time to add versions to the working vocabulary of both Go developers and our tools. The [linked proposal](#) describes a way to do that. See especially the Rationale section for a discussion of alternatives.

This GitHub issue is for discussion about the substance of the proposal.

Other references:

- <https://research.swtch.com/vgo>, the detailed designs
- [A Tour of Versioned Go](#), a walkthrough of what it's like to use
- `go get golang.org/x/vgo`, the prototype implementation
- <https://golang.org/wiki/vgo>, links to related posts, videos, etc
- [vgo milestone](#) on issue tracker



141



4



4



51



46

Assignees

No one assigned

Labels

Proposal

Projects

None yet

Milestone

Proposal

Notifications

Subscribe

You're not receiving notifications from this thread.

37 participants



gopherbot added this to the **Proposal** milestone on 8 Mar





流利说



GopherChina2018

Google 这么多年来怎么
build Go 的?

Code Management :)

Why should I care?



流利说

is scaling up, fast



流利说

```
$ kubectl get pods --all-namespaces | awk '{print $2}' | awk -F '-' '{print $1"-"$2}' | uniq | wc -l
```

213*

Why should I care?

- New product lines, features (D***n, Ba**ta, IELTS, etc.)
- Increasingly sophisticated systems (E.g., adaptive learning system)
- Ever tighter integration
 - Algorithm, backend, data, all part of one system
 - We are in the process of becoming a big data company
- Data exchange between heterogeneous components
 - Multi languages, Go, C++, Python, Java
 - Evolving independently, at varied pace

At scale, managing code becomes
highly non-trivial!

Code Management Challenges

How to propagate changes?

- Has the server / client got the new proto field?
 - Trivial if you just have one proto file
 - How about a.proto imports b.proto who imports c.proto?
 - How about a.proto imports b.proto and c.proto both of which import d.proto?
- I have marked this field deprecated for a month, can I delete? Or someone is still using it...
- Even worse, somebody might have re-purposed it...



```
// Next ID: 17
enum ActivityType {
    PRESENTATION_TYPE = 0; // default is Pre
    MULTI_CHOICE_PICTURE_1 = 1;
    MULTI_CHOICE_PICTURE_2 = 2;
    MULTI_CHOICE_QUESTION_1 = 3; // MultiCho
    MULTI_CHOICE_QUESTION_2 = 4; // MultiCho
    MULTI_CHOICE_QUESTION_6 = 5; // MultiCho
    MULTI_CHOICE_QUESTION_4A = 6; // MultiCh
    MULTI_CHOICE_QUESTION_4B = 8;
    MULTI_CHOICE_QUESTION_5 = 8;
}
```

```
enum CompActivityType {
    UNKNOWN = 0;
    MULTI_CHOICE_PICTURE_1 = 1;
    MULTI_CHOICE_PICTURE_2 = 2;
    MULTI_CHOICE_QUESTION_1 = 3; // Mult
    MULTI_CHOICE_QUESTION_2 = 4; // Mult
    MULTI_CHOICE_QUESTION_6 = 5; // Mult
    MULTI_CHOICE_QUESTION_4A = 6; // Mul
    MULTI_CHOICE_QUESTION_4B = 7; // Num
    MULTI_CHOICE_QUESTION_5 = 8; // Bool
}
```

```
enum ActivityType {
    UNKNOWN_ACTIVITY_TYPE = 0; // Used
    MULTI_CHOICE_PICTURE_1 = 1;
    MULTI_CHOICE_PICTURE_2 = 2;
    MULTI_CHOICE_QUESTION_1 = 3; // Mu
    MULTI_CHOICE_QUESTION_2 = 4; // Mu
    MULTI_CHOICE_QUESTION_6 = 5; // Mu
    MULTI_CHOICE_QUESTION_4A = 6; // M
    MULTI_CHOICE_QUESTION_4B = 7; // N
    MULTI_CHOICE_QUESTION_5 = 8; // Bo
}
```

How to reuse code / solution?

- 不同语言的 fingerprint 函数实现在不同的地方，有更新总是忘记
- proto 因为 import 太复杂放弃写 protoc Makefile 而直接 copy, 人为保证一致性
- proto 生成依赖 go_package option 声明导致没法编译

How to share knowledge?

Introducing Bazel

Open source version of Blaze Google's indoor build system

<https://bazel.build>

Blaze ca. 2017

- 2B LOC in monorepo
- 45K commits/day
- 800k builds/day
- 2 Pb build outputs/day
- 100M test cases/day

```

1 cc_library(
2     name = "status",
3     srcs = ["status.cc"],
4     hdrs = ["status.h"],
5     deps = [
6         ":slice",
7         "//liulishuo/storage/leveldb/port",
8     ],
9 )
10
11 cc_library(
12     name = "slice",
13     srcs = ["slice.h"],
14 )
15
16 cc_library(
17     name = "random",
18     srcs = ["random.h"],
19 )
20
21 cc_library(
22     name = "testharness",
23     testonly = 1,
24     srcs = ["testharness.cc"],
25     hdrs = ["testharness.h"],
26     deps = [
27         ":env",
28         ":random",
29         ":slice",
30     ],
31 )

```

VS

```

1 UTILS = \
2     db/db_bench \
3     db/leveldbutil
4
5 # Put the object files in a subdirectory, but the application at the top of
6 PROGRAMS := $(notdir $(TESTS) $(UTILS))
7
8 # On Linux may need libkyotocabinet-dev for dependency.
9 BENCHMARKS = \
10     doc/bench/db_bench_sqlite3 \
11     doc/bench/db_bench_tree_db
12
13 CFLAGS += -I. -I./include $(PLATFORM_CCFLAGS) $(OPT)
14 CXXFLAGS += -I. -I./include $(PLATFORM_CXXFLAGS) $(OPT)
15
16 LDFLAGS += $(PLATFORM_LDFLAGS)
17 LIBS += $(PLATFORM_LIBS)
18
19 SIMULATOR_OUTDIR=out-ios-x86
20 DEVICE_OUTDIR=out-ios-arm
21
22 ifeq ($(PLATFORM), IOS)
23 # Note: iOS should probably be using libtool, not ar.
24 AR=xcrun ar
25 SIMULATORSDK=$(shell xcrun -sdk iphonesimulator --show-sdk-path)
26 DEVICESDK=$(shell xcrun -sdk iphoneos --show-sdk-path)
27 DEVICE_CFLAGS = -isysroot "$(DEVICESDK)" -arch armv6 -arch armv7 -arch armv7
28 SIMULATOR_CFLAGS = -isysroot "$(SIMULATORSDK)" -arch i686 -arch x86_64
29 STATIC_OUTDIR=out-ios-universal
30 else
31 STATIC_OUTDIR=out-static
32 SHARED_OUTDIR=out-shared
33 STATIC_PROGRAMS := $(addprefix $(STATIC_OUTDIR)/, $(PROGRAMS))
34 SHARED_PROGRAMS := $(addprefix $(SHARED_OUTDIR)/, db_bench)
35 endif
36
37 STATIC_LIBOBJECTS := $(addprefix $(STATIC_OUTDIR)/, $(SOURCES:.cc=.o))
38 STATIC_MEMENVOBJECTS := $(addprefix $(STATIC_OUTDIR)/, $(MEMENV_SOURCES:.cc=

```

google/protobuf
grpc/grpc-go

```
cd $(bazel info execution_root) && bazel-out/host/bin/external/com_google_protobuf/protoc \  
--plugin=protoc-gen-go=bazel-out/host/bin/external/com_github_golang_protobuf/protoc-gen-go/protoc-gen-go \  
--go_out=ignore_go_package,Mgoogle/protobuf/any.proto=github.com/golang/protobuf/ptypes/any,Mgoogle/protobuf/  
compiler/plugin.proto=github.com/golang/protobuf/protoc-gen-go/plugin,Mgoogle/protobuf/  
descriptor.proto=github.com/golang/protobuf/protoc-gen-go/descriptor,Mgoogle/protobuf/duration.proto=github.com/  
golang/protobuf/ptypes/duration,Mgoogle/protobuf/empty.proto=github.com/golang/protobuf/ptypes/empty,Mgoogle/  
protobuf/struct.proto=github.com/golang/protobuf/ptypes/struct,Mgoogle/protobuf/timestamp.proto=github.com/  
golang/protobuf/ptypes/timestamp,Mgoogle/protobuf/wrappers.proto=github.com/golang/protobuf/ptypes/  
wrappers,Mliulishuo/backend/__example__/grpc/a.proto=git.llsapp.com/common/protos/liulishuo/backend/__example__/  
grpc,Mliulishuo/backend/__example__/grpc/b.proto=git.llsapp.com/common/protos/liulishuo/backend/__example__/  
grpc,Mliulishuo/backend/__example__/grpc/c.proto=git.llsapp.com/common/protos/liulishuo/backend/__example__/  
grpc,Mliulishuo/backend/__example__/grpc/d.proto=git.llsapp.com/common/protos/liulishuo/backend/__example__/  
grpc,Mliulishuo/backend/__example__/a.proto=git.llsapp.com/common/protos/liulishuo/backend/  
__example__,Mliulishuo/backend/__example__/a_modes.proto=git.llsapp.com/common/protos/liulishuo/backend/  
__example__,Mliulishuo/backend/__example__/__example__.proto=git.llsapp.com/common/protos/liulishuo/backend/  
__example__,Mliulishuo/backend/__example__/meta.proto=git.llsapp.com/common/protos/liulishuo/backend/  
__example__,Mliulishuo/backend/__example__/models.proto=git.llsapp.com/common/protos/liulishuo/backend/  
__example__,Mliulishuo/backend/__example__/d.proto=git.llsapp.com/common/protos/liulishuo/backend/  
__example__,Mliulishuo/backend/__example__/d_models.proto=git.llsapp.com/common/protos/liulishuo/backend/  
__example__,Mgoogle/api/annotations.proto=github.com/googleapis/googleapis/google/api/annotations,Mgoogle/api/  
http.proto=github.com/googleapis/googleapis/google/api/annotations,plugins=grpc:bazel-out/darwin-fastbuild/  
genfiles \  
--proto_path=external/com_google_protobuf/src \  
--proto_path=. \  
--proto_path=external/com_github_googleapis_googleapis \  
liulishuo/backend/__example__/grpc/a.proto \  
liulishuo/backend/__example__/grpc/b.proto \  
liulishuo/backend/__example__/grpc/c.proto \  
liulishuo/backend/__example__/grpc/d.proto
```



```
1 package(default_visibility = ["/visibility:public"])
2
3 load("@com_llvmapp_git_common_bazel//build_rules/grpc/go:rules.bzl", "go_proto_library")
4
5 filegroup(
6     name = "example_grpc_proto",
7     srcs = [
8         "a.proto",
9         "b.proto",
10        "c.proto",
11        "d.proto",
12    ],
13 )
14
15 go_proto_library(
16     name = "go_default_library",
17     proto_deps = [
18         "//liulishuo/backend/__example:go_default_library",
19         "@com_github_googleapis_googleapis//google/api:annotations",
20     ],
21     protos = [":example_grpc_proto"],
22     with_grpc = True,
23     verbose = 1,
24 )
```


How to use Bazel

Write a `WORKSPACE` file in project root



```
1 workspace(name = "com_github_yuan_expt")
2 http_archive(
3     name = "io_bazel_rules_go",
4     sha256 = "feba3278c13cde8d67e341a837f69a029f698d7a27ddb2a202be7a10b22142a",
5     url = "https://github.com/bazelbuild/rules_go/releases/download/0.10.3/rules_go-0.10.3.tar.gz",
6 )
7
8 http_archive(
9     name = "bazel_gazelle",
10    sha256 = "d03625db67e9fb0905bbd206fa97e32ae9da894fe234a493e7517fd25faec914",
11    url = "https://github.com/bazelbuild/bazel-gazelle/releases/download/0.10.1/bazel-gazelle-0.10.1.tar.gz",
12 )
13
14 load("@io_bazel_rules_go//go:def.bzl", "go_register_toolchains", "go_repository", "go_rules_dependencies")
15
16 go_rules_dependencies()
17 go_register_toolchains()
18
19 load("@bazel_gazelle//:deps.bzl", "gazelle_dependencies")
20 gazelle_dependencies()
21
22 go_repository(
23     name = "com_github_aws_aws_sdk_go",
24     commit = "57564ea051fab8ccbb8cac2aff285647a1aeefd1",
25     #tag = "master",
26     importpath = "github.com/aws/aws-sdk-go",
27 )
```

Write a BUILD file in every directory

```
1 package(default_visibility = ["//visibility:public"])
2
3 load("@com_github_lingochamp_bazel_essentials//build_rules:go.bzl", "static_go_binary")
4 load("@io_bazel_rules_go//go:def.bzl", "go_binary", "go_library", "go_prefix")
5
6 go_prefix("github.com/lingochamp/coursescript")
7
8 go_library(
9     name = "go_default_library",
10    srcs = ["main.go"],
11    visibility = ["//visibility:public"],
12    deps = ["//cmd:go_default_library"],
13 )
14
15 go_binary(
16    name = "coursescript",
17    library = ":go_default_library",
18    linkstamp = "git.llsapp.com/cc/coursescript/cmd",
19    visibility = ["//visibility:public"],
20 )
```



```
1 load("@io_bazel_rules_go//go:def.bzl", "go_library", "go_test")
2
3 package(
4     default_visibility = ["//visibility:public"],
5 )
6
7 load("@io_bazel_rules_go//go:def.bzl", "go_library")
8
9 go_library(
10     name = "go_default_library",
11     srcs = [
12         "config.go",
13         "sort.go",
14         "time.go",
15         "utils.go",
16         "validate.go",
17     ],
18     deps = [
19         "@com_llsapp_git_common_protos//liulishuo/backend/coursescript:services_go_proto",
20     ],
21 )
22
```



```
23 go_test(  
24     name = "go_default_test",  
25     timeout = "short",  
26     srcs = [  
27         "time_test.go",  
28         "config_test.go",  
29         "utils_text.go",  
30         "validate_test.go",  
31     ],  
32     library = ":go_default_library",  
33 )
```


“external”
“filesystem root”
“package”
@com_github_google_protobuf//:protoc
workspace name target name

```
WORKSPACE com_github_google_protobuf
```

```
FILESYSTEM (root)
```

```
PACKAGE (BUILD file)
```

```
RULE protoc (cc_binary)
```

```
RULE protobuf (cc_library)
```

```
RULE protobuf-java (java_library)
```



Bazel

```
build //...
```

```
build //:demo
```

```
run //:demo
```

```
test //:demo_test
```



流利说



GopherChina2018

```
# cross compile
```

```
bazel build --experimental_platforms=@io_bazel_rules_go/go/
```

```
toolchain:linux_amd64 //:demo
```

```
# list all targets
```

```
bazel query @com_github_grpc_grpc//... --output
```

```
label_kind
```

```
# ignore dir
```

```
bazel build //... --//ignored_dir/...
```

protoc

```
bazel run @com_github_google_protobuf//:protoc -- --  
help
```

run gazelle

```
bazel run //:gazelle
```

query deps

```
bazel query "rdeps(deps(//examples/...), @com_llvm//:zlib)"
```

Bazel build //:Go

```
cd <your_go_project> && rm -rf vendor
```



Warning!

行走江湖稳为先

rm -rf 要三思

Go rules for Bazel

github.com/bazelbuild/rules_go

Support

- libraries → go_library
- binaries → go_binary
- tests → go_test
- vendoring
- cgo
- cross compilation
- auto generating BUILD files via gazelle
- protocol buffers

Do not support

- bazel-style auto generating BUILD (where the library name is other than `go_default_library`)
- C/C++ interoperation except `cgo` (`swig` etc.)
- coverage

bazelbuild/bazel-gazelle



流利说

github.com/lingochamp/bazel_essentials

三步走 :)



1. Add **WORKSPACE** file in your project root
2. Run `bazel run //:gazelle` to gen **BUILD**
3. `bazel run //:gazelle -- update-repos` to
add deps

DEMO

build **astaxie/beego** with bazel



Terminal — -zsh — 111x36

-zsh

-zsh



!10148 [16:00:58] ~ \$



Go & Versioning

Posted on Tuesday, February 20, 2018.

These are the posts in the “Go & Versioning” series.

- “[Go += Package Versioning](#)” [PDF].
- “[A Tour of Versioned Go \(vgo\)](#)” [PDF].
- “[Semantic Import Versioning](#)” [PDF].
- “[Minimal Version Selection](#)” [PDF].
- “[Reproducible, Verifiable, Verified Builds](#)” [PDF].
- “[Defining Go Modules](#)” [PDF].
- “[Versioned Go Commands](#)” [PDF].

The official Go proposal is at <https://golang.org/issue/24301>.

1. [x] Packages Versioned
2. [x] Verifiable and verified builds
3. [x] Work outside \$GOPATH

Minimal Version Selection?



简单点，处理的方式简单点

又不是个演员，别设计那些情节

最新版 master/HEAD

指定 commit



- Remote Caching ([buchgr/bazel-remote](#))
- Remote Execution ([bazelbuild/bazel-buildfarm](#))



Bazel

build Docker

build Android

build iOS

build Web



Build and test software of any size,
quickly and reliably



流利说使用 Bazel 中遇到的问题



#1 不能每次都从 master/HEAD pull 最新代码

Skylark

https://github.com/lingochamp/bazel_essentials/blob/master/build_rules/lls.bzl

```
1 """Repository rules for pulling liulishuo repositories."""
2
3 load("@io_bazel_rules_go//go:def.bzl", "go_repository")
19
20 def _lls_repository_head_impl(ctx):
21     result = ctx.execute([
22         ctx.path(ctx.attr._fetch_repo),
23         "--dest", ctx.path(""),
24         "--remote", ctx.attr.remote,
25         "--rev", _fetch_commit(ctx, ctx.attr.remote, "refs/heads/master"),
26         "--vcs", "git",
27         "--importpath", ctx.attr.importpath,
28     ])
29     if result.return_code:
30         fail("%s: failed to fetch from repo %s, err: %s" % (ctx.name, ctx.attr.remote, result.stderr))
31
32 _lls_repository_head = repository_rule(
33     implementation = _lls_repository_head_impl,
34     attrs = {
35         "importpath": attr.string(mandatory = True),
36         "remote": attr.string(mandatory = True),
37         "_fetch_repo": attr.label(
38             default = Label("@io_bazel_rules_go_repository_tools//:bin/fetch_repo"),
39             allow_files = True,
40             single_file = True,
41             executable = True,
42             cfg = "host",
43         ),
44     },
45     # Fetch the repository everytime env BAZEL_RUNID changes.
46     environ = ["BAZEL_RUNID"],
47 )
```



流利说 ❤️ GopherChina2018

https://github.com/lingochamp/bazel_essentials/blob/master/build_rules/lls.bzl

```
48
49 def lls_repository(name, path, commit=None, tag=None, use_local_version=0):
50     """A liulishuo repository.
51
52     If neither commit nor tag is specified, which should be the default case, then
53     we pull the refs/heads/master from the repository for each build. This ensures
54     that any internal change propagates to all the other repositories.
55
56     name: (string) Name of the repository.
57     path: (string) Name relative to git.llsapp.com. E.g.,
58         git.llsapp.com/common/protos will have path name common/protos.
59     commit: (string) Commit.
60     tag: (string) Tag.
61     use_local_version: (int) If 1, use the local mirror of the repository. This
62         assumes that the local folder structure mirrors that one on gitlab.
63     """
64     if use_local_version:
65         return native.local_repository(
66             name = name,
67             path = "../../" + path,
68         )
69     importpath = "git.llsapp.com/" + path
70     remote = "git@git.llsapp.com:" + path + ".git"
71     if commit or tag:
72         # This is a temporary hack as only go_repository supports pulling code using
73         # git@... protocol. Bazel only supports https, which happens to be not
74         # supported by our gitlab: (.)
75         #
76         # TODO(yi.sun): Figure out a way to always pull the master head.
77     return go_repository(
78         name = name,
79         importpath = importpath,
80         remote = remote,
81         vcs = "git",
82         commit = commit,
83         tag = tag,
84     )
85
86     # No commit or tag specified, just pull from the master
87     return _lls_repository_head(
88         name = name,
89         importpath = importpath,
90         remote = remote,
91     )
```



流利说 ♥ GopherChina2018



```
$ export BAZEL_RUNID=$RANDOM
```

```
DEBUG: /private/var/tmp/_bazel_lenage/9643b246fad34c1c5881e264d9c030e9/external/  
com_github_lingochamp_bazel_essentials/build_rules/lls.bzl:16:5: com_llsapp_git_common_bazel: ref refs/  
heads/master from repo git@git.llsapp.com:common/bazel.git is at 9ecefda5d856726787d79faae88c223cb2305ebf
```

```
DEBUG: /private/var/tmp/_bazel_lenage/9643b246fad34c1c5881e264d9c030e9/external/  
com_llsapp_git_common_bazel/build_rules/requires.bzl:53:5: com_github_madler_zlib: date is 2013-04-28  
before 2015, consider updating.
```

```
DEBUG: /private/var/tmp/_bazel_lenage/9643b246fad34c1c5881e264d9c030e9/external/  
com_llsapp_git_common_bazel/build_rules/requires.bzl:53:5: com_google_code_gson_gson: date is 2014-08-11  
before 2015, consider updating.
```

```
DEBUG: /private/var/tmp/_bazel_lenage/9643b246fad34c1c5881e264d9c030e9/external/  
com_llsapp_git_common_bazel/build_rules/requires.bzl:53:5: junit_junit_4: date is 2014-12-04 before 2015,  
consider updating.
```

```
DEBUG: /private/var/tmp/_bazel_lenage/9643b246fad34c1c5881e264d9c030e9/external/  
com_llsapp_git_common_bazel/build_rules/grpc/protobuf/internal/proto_compile.bzl:329:5:
```

#2 上手 bazel 比较慢

CodeLab: Bazel + GRPC

Codelab for Bazel + GRPC

This codelab demonstrates how to use Bazel to *properly* build GRPC services in various languages.

After this codelab, you should feel comfortable

- Using Bazel;
- Writing *protocol buffers* for GRPC services;
- Implementing services in **C++**, **Go**, **Python** or **Java**;
 - Usage of other programming languages are *most strongly* discouraged.
- Writing *unit* and *regression* tests, and running them as part of CI.

#3 第三方依赖问题



```
17 load("@com_github_lingochamp_bazel_essentials//build_rules:github.bzl", "github_http_archive", "github_go_repository")
18 load("@com_github_lingochamp_bazel_essentials//build_rules:lls.bzl", "lls_repository")
19
20 lls_repository(
21     name = "com_llsapp_git_common_bazel",
22     path = "common/bazel",
23 )
24
25 load("@com_llsapp_git_common_bazel//build_rules/grpc/go:rules.bzl", "go_proto_repositories")

64
65 github_go_repository(
66     name = "com_github_qiniu_x",
67     # 2015-07-21
68     commit = "946c4a16076d6d98aeb78619e2bd4012357f7228",
69     package = "qiniu/x",
70 )
71
72
```



流利说



GopherChina2018

```
$ bzb @com_github_qiniu_x//...
```

```
ERROR: /private/var/tmp/_bazel_lenage/f1f0f8f5b66c92414ecb10226f21ed13/  
external/com_github_qiniu_x/xlog.v7/BUILD.bazel:3:1: no such package  
'@com_qiniupkg_x//reqid.v7': The repository could not be resolved and  
referenced by '@com_github_qiniu_x//xlog.v7:go_default_library'
```

```
ERROR: Analysis of target '@com_github_qiniu_x//  
xlog.v7:go_default_library' failed; build aborted: no such package  
'@com_qiniupkg_x//reqid.v7': The repository could not be resolved
```

```
INFO: Elapsed time: 11.576s
```

```
FAILED: Build did NOT complete successfully (17 packages loaded)
```

```
1 workspace(name = "com_llsapp_git_cc_coursescript")
2
3 http_archive(
4     name = "com_github_lingochamp_bazel_essentials",
5
55
56 # NOTE(yuan): since qiniu using different import path in same package,
57 # so we need to duplicate deps to fix it
58 github_go_repository(
59     name = "com_qiniupkg_x",
60     importpath = "qiniupkg.com/x",
61     # 2015-07-21
62     commit = "946c4a16076d6d98aeb78619e2bd4012357f7228",
63     package = "qiniu/x",
64 )
65
66 github_go_repository(
67     name = "com_github_qiniu_x",
68     # 2015-07-21
69     commit = "946c4a16076d6d98aeb78619e2bd4012357f7228",
70     package = "qiniu/x",
71 )
72
73
```

16 code results in [qiniu/x](#)

[config.v7/load_conf.go](#)

Showing the top match Last indexed on 20 Sep 2016

```
1 package config
2
3 import (
4     "bytes"
5     "encoding/json"
6     "flag"
7     "io/ioutil"
8
9     "qiniupkg.com/x/log.v7"
```

[rpc.v7/gob/gobrpc_client.go](#)

Showing the top match Last indexed on 20 Sep 2016

```
9     "strconv"
10
11     "qiniupkg.com/x/rpc.v7"
12
13     . "golang.org/x/net/context"
```

如果你想折腾 Go 和 bazel
欢迎来流利说 :p

公众号: lls_tech



Thank you & Questions