


# はじめての GoogleAppEngine/Go

2016/03/10 APG Engineer Night

GMOアドマーケティング株式会社  
横越 光

# 自己紹介

- 横越 光(よこごし ひかる)
- 27さい
- 普段はPHPたまにAndroid
- 最近SSPをやることになりました→

# GoogleAppEngineとは？

- サーバのメンテナンス・スケーリングを自動でしてくれる！ PaaS(Platform-as-a-Service)
- CloudDatastore (NoSQL)、Memcache、全文検索などの機能がビルトイン
- アクセス量によってはずっと**無料**で使用できる
- MySQL(CloudSQL)も使える
  - ただし有料

# 使える言語

- Python
- Java
- PHP
- **Go**
- Managed VMっていうのを使えばRubyやNode.jsもいけるらしい...？
  - だれか教えてください

# SDKのインストール

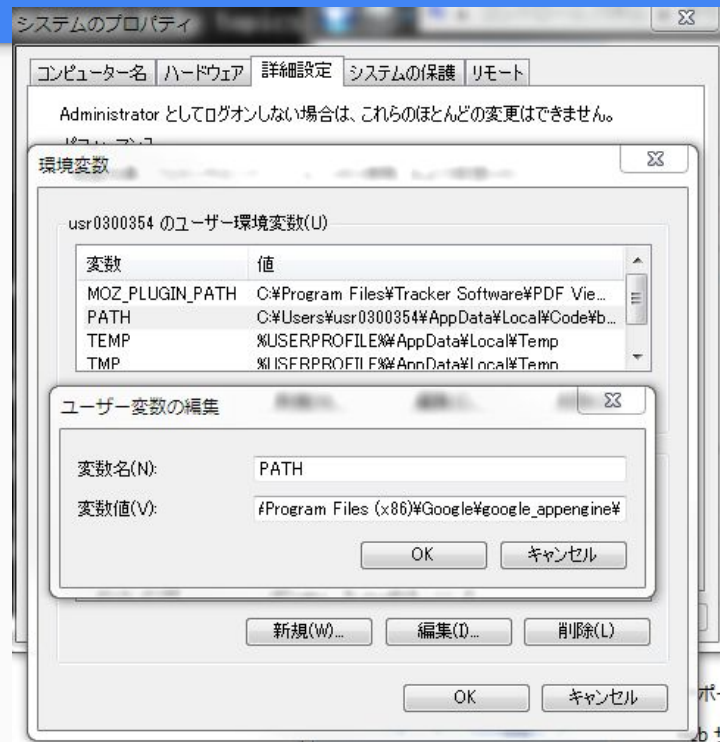
- [https://cloud.google.com/appengine/downloads?hl=ja#Google\\_App\\_Engine\\_SDK\\_for\\_Go](https://cloud.google.com/appengine/downloads?hl=ja#Google_App_Engine_SDK_for_Go)
- Python2.7が必要
- 今回はWindowsでやります

## Google App Engine SDK for Go

Platform	Version	Package	Size	SHA1 Checksum
Linux 64-bit	1.9.33 - 2016-02-22	<a href="#">go_appengine_sdk_linux_amd64-1.9.33.zip</a>	36.0 MB	b27702571641f8502a914a695dc0366efa469fe4
Linux 32-bit	1.9.33 - 2016-02-22	<a href="#">go_appengine_sdk_linux_386-1.9.33.zip</a>	34.8 MB	0ca3163af0a8d2321127733f149954c1645b7a98
Mac OS X 64-bit	1.9.33 - 2016-02-22	<a href="#">go_appengine_sdk_darwin_amd64-1.9.33.zip</a>	34.2 MB	e8e97118b8b4f5565da9d436499d2f45a8260bdb
Mac OS X 32-bit	1.9.33 - 2016-02-22	<a href="#">go_appengine_sdk_darwin_386-1.9.33.zip</a>	33.2 MB	92060a41c7aa8532ba1a6b13806564e139838294
Windows 64-bit	1.9.33 - 2016-02-22	<a href="#">go_appengine_sdk_windows_amd64-1.9.31.zip</a>	35.2 MB	38500dd69e4c8fbd9c503ea7c55e0a19a04db986
Windows 32-bit	1.9.33 - 2016-02-22	<a href="#">go_appengine_sdk_windows_386-1.9.31.zip</a>	34.3 MB	4ca246af49f5125c15722521e54b7ce1811b083e

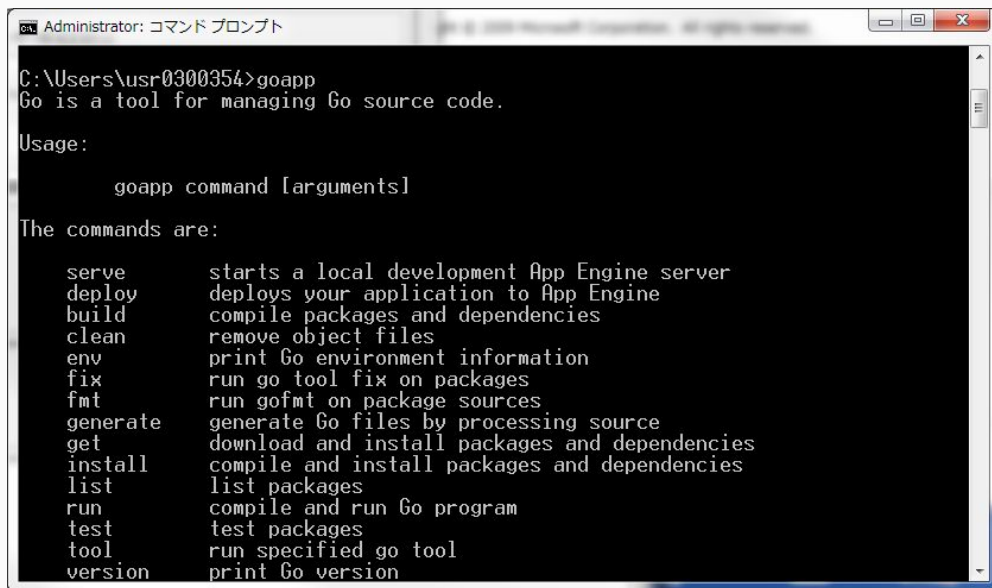
# PATH通す

- インストールした所にパスを通しましょう



# goappコマンドの確認

- コンソールでgoappと打つとヘルプがいっぱい出てくる



```
Administrator: コマンド プロンプト
C:\Users\usr0300354>goapp
Go is a tool for managing Go source code.

Usage:

    goapp command [arguments]

The commands are:

serve      starts a local development App Engine server
deploy     deploys your application to App Engine
build      compile packages and dependencies
clean      remove object files
env        print Go environment information
fix        run go tool fix on packages
fmt        run gofmt on package sources
generate   generate Go files by processing source
get        download and install packages and dependencies
install    compile and install packages and dependencies
list       list packages
run        compile and run Go program
test       test packages
tool       run specified go tool
version    print Go version
```

# プロジェクトの作成

- 「Google Cloud Platform」とぐぐってコンソールに移動します
- 右上あたりの「プロジェクトの作成」からプロジェクトを作ります
- アカウント登録していない人は登録しよう！



プロジェクト: appengine-go-sample  
ID: elated-bus-124514 (#224606131811)

### Compute Engine を試す

このチュートリアルに従って操作すると、仮想マシンを起動し、Google Compute Engine、Node.js、MongoDB を使ったダストバックアプリを作成できます。

試してみる

### Try App Engine (standard environment)

このチュートリアルに従って操作すると、基礎となるインフラストラクチャを築くことなく Hello World アプリを作成し、デプロイできます。

- App Engine ドキュメント
- Go のチュートリアルを開始する
- Java™ Java™ Runtime のチュートリアルを開始する
- PHP のチュートリアルを開始する
- Python™ のチュートリアルを開始する

### Google API を利用する

API を有効にし、認証情報を作成して、使用状況を監視します

API を有効にして管理しましょう

### App Engine (管理対象 VM 環境) を試す

お好みの言語で、管理対象 VM (ベータ版) を使ってサンプルアプリケーションをデプロイします

- 管理対象 VM ドキュメント
- Node.js のチュートリアルを開始する
- Ruby のチュートリアルを開始する

### Cloud Storage バケットの作成

Cloud Storage を使用して非構造化データを安全に保存し、高可用性を確保します

試してみる

- ### ドキュメント
- Google Cloud Platform のドキュメント
  - Cloud Platform ショーケース
  - Cloud Platform のチュートリアル
- ### 既製のソリューションで開始
- 数回クリックするだけで Google Cloud Platform 上でソフトウェアパッケージを起動できます
- パッケージを探す

最近選択したプロジェクト

ekos-web	ekos-web
geinin-app	geinin-app
num-gift	num-gift
wordpress	ekos-wordpress
game-timer	game-timer

プロジェクト名またはプロジェクト ID でフィルタ

すべてのプロジェクト

API Project	ekos-over1
✓ appengine-go-sample	elated-bus-124514
ekos-web	ekos-web
game-timer	game-timer
geinin-app	geinin-app

すべてのプロジェクトの管理  
プロジェクトの作成...

### Hello World!

shows how to deploy a sample Go Google App Engine using the gcloud

steps you will be taking.

run your "Hello, world!" app learn how to run your app using Google Cloud in your browser. At the end you'll deploy your web using the gcloud command.

app... will be real and you'll be able to experiment after you deploy, or you can remove it and start

gopher was designed by Renee French and drawn by censored under Creative Commons 3.0 Attribution.

ホーム

ダッシュボード

ダッシュボード

アクティビティ

プロジェクト: appengine-go-sample

プロジェクトの詳細

プロジェクト ID

**elated-bus-124514**

プロジェクト番号

224606131811

メモっておく!

Google API を利用する

APIを有効にし、認証情報を作成して、使用状況を追跡します

API APIを有効にして管理しましょう

ドキュメント

- Google Cloud Platform のドキュメント
- Cloud Platform ソリューション
- Cloud Platform のチュートリアル

Compute Engine を試す

このチュートリアルに従って操作すると、仮想マシンを起動し、Google Compute Engine、Node.js、MongoDB を使ったゲストブックアプリを作成できます。

使ってみる

App Engine (管理対象 VM 環境) を試す

お好みの言語で、管理対象 VM (ベータ版) を使ってサンプル アプリケーションをデプロイします

- 管理対象 VM ドキュメント
- Node.js のチュートリアルを開始する
- Ruby のチュートリアルを開始する

既製のソリューションで開始

数回クリックするだけで Google Cloud Platform 上でソフトウェア パッケージを起動できます

パッケージを探す

Try App Engine (standard environment)

このチュートリアルに従って操作すると、基礎となるインフラストラクチャを気にすることなく Hello World アプリを作成し、デプロイできます。

- App Engine ドキュメント
- Go のチュートリアルを開始する
- Java™ Java™ Runtime のチュートリアルを開始する
- PHP のチュートリアルを開始する
- Python™ のチュートリアルを開始する

Cloud Storage バケットの作成

Cloud Storage を使用して非構造化データを安全に保存し、高可用性を確保します

使ってみる

Deploying "Hello World"

Introduction

This tutorial shows you how to deploy a sample Go application to Google App Engine using the gcloud command.

Here are the steps you will be taking.

- Build and run your "Hello, world" app. You will learn how to run your app using Google Cloud Shell, right in your browser. At the end you'll deploy your app to the web using the gcloud command.
- After the app... Your app will be real and you'll be able to experiment with it after you deploy, or you can remove it and start fresh.

The Go Language gopher was designed by Renee French and drawn by Takuya Ueda. Licensed under Creative Commons 3.0 Attributions.

次へ

× チュートリアルをキャンセル

# ソースコード準備

- 最低限「main.go」とコンフィグファイルである「app.yaml」が必要

# main.go

- martiniというフレームワークを使用しています
  - Rubyで言うSinatra
- 12行目でルーティング、クライアント返却要素の登録

```
1 package main
2
3 import (
4     "net/http"
5     "github.com/go-martini/martini"
6     "appengine"
7 )
8
9 func init() {
10     m := martini.Classic()
11     m.Use(AppEngine)
12     m.Get("/", func() string {
13         return "Hello world!"
14     })
15     http.Handle("/", m)
16 }
17 func AppEngine(c martini.Context, r
18 *http.Request) {
19     c.MapTo(appengine.NewContext(r),
20 (*appengine.Context)(nil))
21 }
```

# app.yaml

- versionはint値を指定。  
AppEngine側でバージョンごとにリソースを別で扱ってくれる
  - コンソール上で前バージョンに切り戻したりできる
- handlersでは例えば静的ファイル等を配信するためにPathをマッピングしたりする

```
1 version: 1
2 runtime: go
3 api_version: go1
4
5 handlers:
6 - url: /*
7   script: _go_app
```



コンピュータ > ローカルディスク (C:) > work > go > appengine\_go\_sample

appengine\_go\_sampleの検索

整理 ライブラリに追加 共有 書き込む 新しいフォルダー

jdk1.6.0\_45

ライブラリ

- Git
- Subversion
- ドキュメント
- ピクチャ
- ビデオ
- ミュージック

コンピューター

- ローカルディスク (C:)
- Apple iPhone
- HTL22

3 個の項目

名前	更新日時	種類	サイズ
.git	2016/03/10 0:03	ファイル フォル...	
app.yaml	2016/03/10 0:52	YAML ファイル	1 KB
main.go	2016/03/10 1:07	GO ファイル	1 KB

# プロジェクトとリソースの紐付

- `$ appcfg.py -A <PROJECT_ID> update /path/to/project`
  - `<PROJECT_ID>`は先ほどプロジェクトを作った際にメモったやつ
  - `/path/to/project`は`main.go`と`app.yaml`が置いてある場所
- 自動的にブラウザが立ち上がり、認証画面が出てくるので画面に従い許可すればOK
  - GUIがない環境はどうなるのだろうか

Administrator: コマンド プロンプト

tions and saving the datastore

INFO 2016-03-10 01:24:37,423 api\_server.py:6511 Saving search indexes

Terminate batch job (Y/N)? Y

C:\Users\usr0300354>appcfg.py -A elated-bus-124514 update C:\work\go\appengine\_g  
o\_sample

01:30 AM Application: elated-bus-124514; version: 1

01:30 AM Host: appengine.google.com

01:30 AM

Starting update of app: elated-bus-124514, version: 1

01:30 AM Getting current resource limits.

01:30 AM Scanning files on local disk.

01:30 AM Cloning 11 application files.

01:30 AM Uploading 1 files and blobs.

01:30 AM Uploaded 1 files and blobs.

01:30 AM Compilation starting.

01:30 AM Compilation: 10 files left.

01:30 AM Compilation completed.

01:30 AM Starting deployment.

01:30 AM Checking if deployment succeeded.

01:30 AM Deployment successful.

01:30 AM Checking if updated app version is serving.

01:30 AM Completed update of app: elated-bus-124514, version: 1

C:\Users\usr0300354>



# リソースのダウンロード

- main.goで使用していたgithub.com/go-martini/martini のようなパッケージを使用するには事前にSDKの機能を使ってダウンロードを行っておく必要がある
- `$ goapp get github.com/go-martini/martini`

# 検証

- `$ goapp serve /path/to/project`
- ローカル内でwebサーバを立ち上げ <http://localhost:8080/> で確認できるようにする機能。
- コンソールから起動している間はログが流れる。便利
  - 起動している間にソースコード変更があった場合は再起動せずとも勝手に変更を適用してくれる！ますます便利

Administrator: コマンド プロンプト - goapp serve C:\work\go\appengine\_go\_sample

```
C:\Users\usr0300354>goapp serve C:\work\go\appengine_go_sample
INFO      2016-03-10 01:39:06,075 devappserver2.py:763] Skipping SDK update check
INFO      2016-03-10 01:39:06,331 api_server.py:205] Starting API server at: http
://localhost:58040
INFO      2016-03-10 01:39:06,344 dispatcher.py:197] Starting module "default" ru
nning at: http://localhost:8080
INFO      2016-03-10 01:39:06,351 admin_server.py:118] Starting admin server at:
http://localhost:8000
INFO      2016-03-10 01:42:25,542 module.py:808] default: "GET / HTTP/1.1" 200 12
INFO      2016-03-10 01:42:36,433 module.py:808] default: "GET / HTTP/1.1" 200 12
INFO      2016-03-10 01:42:41,867 module.py:808] default: "GET / HTTP/1.1" 200 12
INFO      2016-03-10 01:42:46,875 module.py:808] default: "GET /error HTTP/1.1" 4
04 19
INFO      2016-03-10 01:43:12,841 module.py:808] default: "GET / HTTP/1.1" 200 12
INFO      2016-03-10 01:43:19,384 module.py:808] default: "GET / HTTP/1.1" 200 12
INFO      2016-03-10 01:43:24,618 module.py:808] default: "GET /test HTTP/1.1" 40
4 19
```



Hello world!

# デプロイ

- `$ goapp deploy /path/to/project`
- AppEngine内にアップロードされ、ここではじめてグローバルアクセスが可能になる。
- [https://<PROJECT\\_ID>.appspot.com/](https://<PROJECT_ID>.appspot.com/)

Administrator: コマンド プロンプト

```
C:\Users\usr0300354>goapp deploy C:\work\go\appengine_go_sample
01:50 AM Application: elated-bus-124514; version: 1
01:50 AM Host: appengine.google.com
01:50 AM
Starting update of app: elated-bus-124514, version: 1
01:50 AM Getting current resource limits.
01:50 AM Scanning files on local disk.
01:50 AM Cloning 11 application files.
01:50 AM Compilation starting.
01:50 AM Compilation: 10 files left.
01:50 AM Compilation completed.
01:50 AM Starting deployment.
01:50 AM Checking if deployment succeeded.
01:50 AM Deployment successful.
01:50 AM Checking if updated app version is serving.
01:50 AM Completed update of app: elated-bus-124514, version: 1

C:\Users\usr0300354>
```

# まとめ

- リソース編集から検証、本番反映までの流れが完成されているので非常にやりやすく感じる
- 導入がちょっとつまづく点が多かったです...が、一度導入してしまえば開発はスムーズ
  - でもGoのwebサーバを1からセットアップするよりは楽なんだろうな ...と思っています # 調べてはいない
- MVCなフレームワークは現状使えないっぽい
  - GAE/PHPならCakePHPが使える
- ホントはDB接続とかのとも紹介したかったです

# 本日の資料

- [https://github.com/ekos/appengine\\_go\\_sample](https://github.com/ekos/appengine_go_sample)
  - main.goとapp.yamlが入っています



# 質疑応答