

PruneMobile マニュアル

Ver.1.0

目 次

1. PRUNEMOBILE とは	3
1.1. 使用環境	3
1.2. サンプルプログラムの環境	3
1.3. 現時点で確認している問題点で、いずれ直すもの	4
2. サンプルプログラムの動作方法	4
2.1. Step 1 サーバの起動	4
2.2. Step 2 ブラウザの起動	4
2.3. Step 3 クライアントの起動	4
3. クライアントプログラムで使う I/F(データ形式)	5
3.1. 前提	5
3.2. Step.1 マーカーの登録	6
3.3. Step.2 マーカーの移動	7
3.4. Step.3 マーカーの抹消	7

<略号>

履歴

日時	バージョン	内容
2020/10/08	Ver. 0.0	初版

1. PruneMobile とは

複数の人間や自動車等の移動体のリアルタイムの位置情報を、地図上に表示する、PruneCluster のアプリケーションです。

PruneMobile のサーバに対して、任意のタイミングで位置情報を送り込むだけで、地図上にマーカーが表示されます。

1.1. 使用環境

- golang(Go 言語)のインストールされていれば良いです。私(開発者)の環境では以下のようになっています。

```
$ PruneMobile¥server>go version
$ go version go1.14 windows/amd64
```

- 実際に動かせば、コンパイラから、あれこれ言われますので、それに対応して下さい。基本的には、

```
$ go get github.com/gorilla/websocket
```

は必要になると思います。

1.2. サンプルプログラムの環境

- <https://github.com/TomoichiEbata/PruneMobile.git> からダウンロードしてください。
- Web ブラウザで表示される地図は、東京都江東区の豊洲駅を中心にして作ってあります。

```
- PruneMobile.go (X は数字) の中にある、
var map = L.map("map", {
    attributionControl: false,
    zoomControl: false
}).setView(new L.LatLng(35.654543, 139.795534), 18);
```

の“35.654543, 139.795534”の座標を変更すれば、地図の中心位置が変わります。

- クライアントプログラムでは、豊洲駅を中心にランダムウォークさせています

- PruneMobile.go (Xは数字)を起動すると、10 秒間程、マーカーが移動して、その後消滅します。
- クライアントプログラム(clientX.go)は複数同時に起動させることができます。

1.3. 現時点で確認している問題点で、いずれ直すもの

- マーカーの消滅のタイミングが、同時になってしまう
- Web ブラウザの表示が、最初の 1 つめしか、正常に動作しない
- ローカルの js(javascript)のローディングに失敗した為、江端のプライベートサーバ(kobore.net)からローディングしている。PruneMobile.go の以下を参照

```
<script src="http://kobore.net/PruneCluster.js"></script>  
<link rel="stylesheet" href="http://kobore.net/examples.css"/>
```

2. サンプルプログラムの動作方法

2.1. Step 1 サーバの起動

適当なシェルを立ち上げて

```
$ cd PruneMobile¥server  
$ go run serverX.go (Xは数字)
```

とすると、「Windows セキュリティの重要な警告(windows10 の場合)」が出てくるので、「アクセスを許可する」ボタンを押下して下さい。

2.2. Step 2 ブラウザの起動

Chromo ブラウザ(他のブラウザについては、T.B.D.)から、

http://localhost:8080/

と入力して下さい。豊洲地区の地図が表示されます。

2.3. Step 3 クライアントの起動

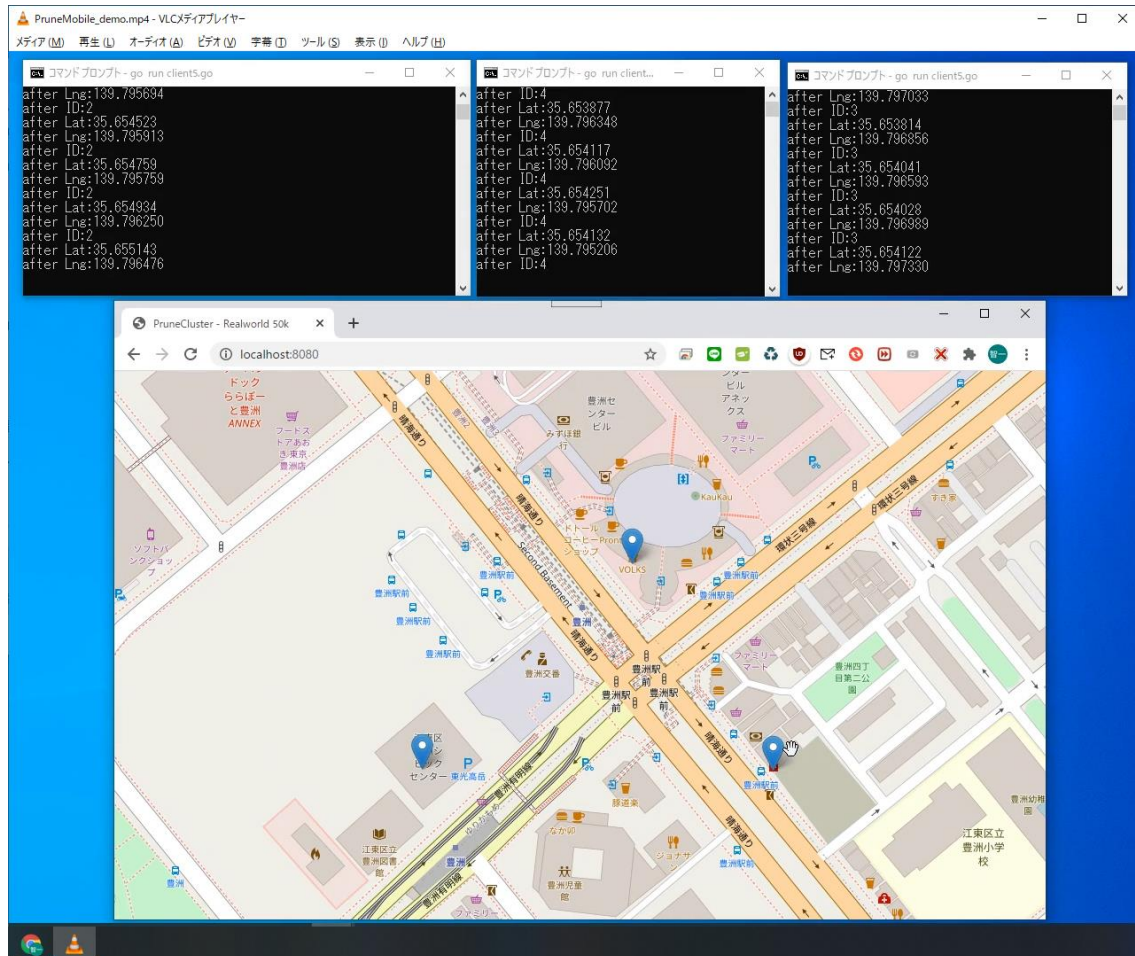
適当なシェルを立ち上げて

```
$ cd PruneMobile¥client
```

```
$ go run clientX.go (Xは数字)
```

とすると、マーカが 0.5 秒単位でランダムに動きます。

2.4. 動作状態の一例



3. クライアントプログラムで使う I/F(データ形式)

3.1. 前提

- サーバと websocket のコネクションを確立して下さい。方法は問いません。golang での記述方法は client/clientX.go を参考にして下さい。
- データ形式は JSON を用います。golang での記載サンプルは以下の通りです。

```
// GetLoc GetLoc
type GetLoc struct {
    ID int    `json:"id"`
    Lat float64 `json:"lat"`
    Lng float64 `json:"lng"`
    //Address string `json:"address"`
}
```

3.2. Step.1 マーカーの登録

ID を“0”にして、最初のマーカーの座標を入力した JSON を、サーバに送付して下さい。
golang での送信方法は以下の通りです。

```
gl := GetLoc{
    ID: 0,
    Lat: 35.653976,
    Lng: 139.796821,
}

err = c.WriteJSON(gl)
if err != nil {
    log.Println("write:", err)
}
```

返り値に入ってきた ID が、これからそのマーカーで使う ID 番号となります。golang での受信方法は以下の通りです。

```
gl2 := new(GetLoc)
err = c.ReadJSON(gl2)
log.Printf("after ID:%d", gl2.ID)
log.Printf("after Lat:%f", gl2.Lat)
log.Printf("after Lng:%f", gl2.Lng)
```

以後、この ID 番号(整数)を使用して下さい。この番号と異なる番号を使用した場合、動作は保証されません。

3.3. Step.2 マーカーの移動

指定された ID を使って、移動後の座標を送付して下さい。

```
gl := GetLoc{
    ID: 5,          // ID が"5"の場合
    Lat: 35.653923,
    Lng: 139.796852,
}

err = c.WriteJSON(gl)
if err != nil {
    log.Println("write:", err)
}
```

返り値は、入力と同じ JSON が戻ってきますが、必ず受信して下さい。

```
gl2 := new(GetLoc)
err = c.ReadJSON(gl2)
log.Printf("after ID:%d", gl2.ID)
log.Printf("after Lat:%f", gl2.Lat)
log.Printf("after Lng:%f", gl2.Lng)
```

3.4. Step.3 マーカーの抹消

指定された ID を使って、地球上の緯度経度の数値で現わせない座標を入力して下さい。
具体的に、lat に 90.0 より大きな値、または lng に 180 より大きな値を入力することで、マーカーが消去されます。

```
gl := GetLoc{
    ID: 5,          // ID が"5"の場合
    Lat: 999.9,
    Lng: 999.9,
}

err = c.WriteJSON(gl)
if err != nil {
```

```
        log.Println("write:", err)
    }
```

返り値は、入力と同じ JSON が戻ってきますが、必ず受信して下さい。

```
gl2 := new(GetLoc)
err = c.ReadJSON(gl2)
log.Printf("after ID:%d", gl2.ID)
log.Printf("after Lat:%f", gl2.Lat)
log.Printf("after Lng:%f", gl2.Lng)
```

以上