



EGS (ezcat Gateway System)
HTTP API Reference

(V1)

西元 2019 年 05 月 30 日

EGS, ezcat Gateway System

HTTP API Reference

- 1 EGS 系統以 Browser / Server 的結構提供用戶端連線使用。
- 2 用戶端 (client) 透過 http api 使用伺服器上的軟體服務。
 - 2.1 EGS 接收到由 client 傳來的 Http Post 後，解析 REQUEST 資訊後執行相關的處理後產生相對應的 RESPONSE 資料回傳到 client。
 - 2.2 Http Post 對象 URL 格式說明

http://localhost:8800/egs

| | | |
|---|-------------|-----------------------|
| 1 | Protocol | http:// |
| 2 | Host | localhost (視實際主機位置而定) |
| 3 | Port | 8800 (視實際設定而定) |
| 4 | Script Name | /egs |

- 2.3 Post 與 Response 內容中的每一個參數以"&"相連，參數的名稱與內容以"="相連。
- 2.4 Post 中所有中文字以 UTF-8 編碼，內容若有中文或是符號者 (含折行) 再以 URL 方式編碼。
- 2.5 Response 中所有中文字以 UTF-8 編碼。
- 3 系統採用的 HTTP 傳輸機制遵循標準的 W3C 規範，client 每個 http transaction 的 REQUEST 都可以對應到一個 RESPONSE 的結果。
- 4 系統環境及處理能力說明
 - 4.1 支援作業系統：Microsoft Windows 2003/XP/Vista/Windows 7,8,10 作業系統 (正體中文版)
 - 4.2 建議硬體環境：
 - Intel Pentium 4 或同等級以上的個人電腦。
 - 可以連線到網際網路。
 - 主記憶體 (RAM) 建議 1GB 以上。
 - 至少 1GB 的可用硬碟空間。
 - 4.3 處理能力：在符合建議的硬體環境下，每分鐘約可解析 1200 筆地址(每秒可以處理 20 筆的地址資料，每筆地址的處理時間約 0.05 秒)



5 基本功能

5.1 查詢 EGS 資訊

Request

- cmd=query_egs_info

Response

- status= <string:OK|ERROR>
- message= <string:錯誤訊息>
- egs_version= <string:EGS 目前版本>
- address_db_version= <string:速達五碼郵號地址庫目前版本>
- sandbox_mode= <0 or 1:處於 Sandbox 模式>
- internet_online= <0 or 1:啟用網際網路連線>

範例：取得現在 EGS 版本資訊。

Post URL：

http://localhost:8800/egs

Post Data：

cmd=query_egs_info

Response Data：

status=OK

&message=&egs_version=1.9.0&address_db_version=16063002&sanbox_mode=0&internet_online=1



5.2 查詢地址對應的速達五碼郵遞區號

Request

- cmd=query_suda5v2 (修改傳入指令增增 v2)
- address_<int:編號>= <string:地址>

Response

- status= <string:OK|ERROR>
- message= <string:錯誤訊息>
- suda5_<int:序號>= <string:速達五碼郵遞區號>

(注意！除了 cmd 外，address_*欄位可多筆上傳，sud5 會回應與 address_*對應序號的欄位，但需注意有最大 200 筆的上限限制，建議逐筆查詢)

範例 1：當我只有筆地址需要 egs 幫我解析 5 碼郵號，取得「桃園縣桃園市中山北路 17 之 1 號 2 樓」的速達五碼郵號。

Post URL：

`http://localhost:8800/egs`

Post Data：

```
cmd=query_suda5v2&address_1=%E6%A1%83%E5%9C%92%E7%B8%A3%E6%A1%83%E5%9C%92%E5%B8%82%E4%B8%AD%E5%B1%B1%E5%8C%97%E8%B7%AF17%E4%B9%8B1%E8%99%9F2%E6%A8%93
```

Response Data：

```
status=OK & suda5_1=33042A (回傳郵號會增加一碼英文字為理貨區碼)
```

範例 2：當我有多筆地址，想要一次性向 egs 取得解析 5 碼郵號的結果，取得「桃園縣桃園市中山北路 17 之 1 號 2 樓」與「新北市新莊區昌明街 48 號 3 樓」的速達五碼郵號。(注意！用這種方式，我們最多一次只能向 egs 要求分析 200 筆地址，建議逐筆查詢)

Post URL：

`http://localhost:8800/egs`

Post Data：

```
cmd=query_suda5v2&address_1=%E6%A1%83%E5%9C%92%E7%B8%A3%E6%A1%83%E5%9C%92%E5%B8%82%E4%B8%AD%E5%B1%B1%E5%8C%97%E8%B7%AF17%E4%B9%8B1%E8%99%9F2%E6%A8%93&address_2=%E6%96%B0%E5%8C%97%E5%B8%82%E6%96%B0%E8%8E%8A%E5%8D%80%E6%98%8C%E6%98%8E%E8%A1%9748%E8%99%9F3%E6%A8%93
```

Response Data：

```
status=OK& suda5_1=33042A&suda5_2=24273B
```



範例 3：當我有超過 200 筆以上的地址，想要 egs 幫我解析 5 碼郵號時，系統該怎麼呼叫 egs 呢？其實前端系統只要將多筆地址，用迴圈方式一筆一筆向 egs 要求解析即可達到目的!!取得「桃園縣桃園市中山北路 17 之 1 號 2 樓」、「新北市新莊區昌明街 48 號 3 樓」、「台北市南京東段三段 261 號」的速達五碼郵號。

說明：處理方式等同於範例 1，但需要先端程式，先將要解析的地址用迴圈的方式，一筆一筆呼叫 egs 進行解析。

```
for (int i = 0; i < 需要解析5碼郵號的筆數; i++)  
{  
    回傳的5碼 = query_suda5v2&address_1=這次需解析的地址  
}
```

[備註]

1. Request 的參數 address_* 不可以是空白字串。
2. address_* 欄位可多筆上傳，suda5 會回應與 address_* 對應序號的欄位。
3. 若地址是空白時，請傳入 noaddress 進行解析



5.3 查詢速達五碼郵遞區號對應的轉運站名稱

Request

- cmd=query_base
- suda5_<int:編號>=<string:速達五碼郵遞區號>

Response

- status=<string:OK|ERROR>
- message=<string:錯誤訊息>
- base_<int:序號>=<string:轉運站名稱>

(注意！除了 cmd 外，suda5_*欄位可多筆上傳，base 會回應與 suda5_*對應序號的欄位，但需注意有最大 200 筆的上限限制)

範例 1：當我只有一筆 5 碼需要 egs 幫我解析轉運站，取得「33042」的速達對應轉運站名稱。

Post URL：

http://localhost:8800/egs

Post Data：

cmd=query_base&suda5_1=33042

Response Data：

status=OK&base_1=32

範例 2：當我有多筆 5 碼郵號，想要一次性向 egs 取得解析轉運站的結果，取得「33042」與「24273」的速達對應轉運站名稱。(注意！用這種方式，我們最多一次只能向 egs 要求分析 200 筆轉運站)

Post URL：

http://localhost:8800/egs

Post Data：

cmd=query_base&suda5_1=33042&suda5_2=24273

Response Data：

status=OK&base_1=32&base_2=31



範例 3：當我有超過 200 筆以上的 5 碼郵號，想要 egs 幫我解析轉運站時，系統該怎麼呼叫 egs 呢？其實前端系統只要將多筆 5 碼郵號，用迴圈方式一筆一筆向 egs 要求解析即可達到目的!! 取得「33042」、「24273」、「11102」的速達對應轉運站名稱。

說明：處理方式等同於範例 1，但需要前端程式，先將要解析的 5 碼郵號用迴圈的方式，一筆一筆呼叫 egs 進行解析。

```
for (int i = 0; i < 需要解析轉運站的筆數; i++)  
{  
    回傳的轉運站 = query_base&suda5_1=這次需解析的5碼郵號  
}
```

[備註]

1. Request 的參數 suda5_* 不可以是空白字串。
2. suda5_* 欄位可多筆上傳，base 會回應與 suda5_* 對應序號的欄位。
3. 若五碼是空白時，請傳入 99999 或是 00000 進行解析



5.4 查詢契客資料

Request

- cmd=query_customers

Response

- status= <string:OK|ERROR>
- message= <string:錯誤訊息>
- customer_<int:序號>= <string:契客代號>,"<string:契客名稱>",<0 or 1:已驗證通過>,<0 or 1:屬於客樂得契約客戶>,<

範例：查詢現在的契客資料。

Post URL：

<http://localhost:8800/egs>

Post Data：

cmd=query_customers

Response Data：

status=OK&customer_0=8066237501,測試契客 1,1,1&customer_1=8066237502,測試契客 2,0,0



5.5 申請速達託運單號碼

Request

- cmd=query_waybill_id_range
- customer_id= <string:契約客戶代號>
- waybill_type= <string:託運單類別 "A"=一般託運單|"B"=代收託運單|"G"=報值託運單>
- count= <int:數量>

Response

- status= <string:OK|ERROR>
- message= <string:錯誤訊息>
- waybill_type= <string:託運單類別 "A"=一般託運單|"B"=代收託運單|"G"=報值託運單>
- waybill_id= <string:速達託運單號>, <string:速達託運單號>, <string:速達託運單號>...

範例：申請 10 個託運單號碼。

Post URL：

http://localhost:8800/egs

Post Data：

cmd=query_waybill_id_range&customer_id=8066237502&waybill_type=A&count=10

10 碼託運單回傳資料

Response Data：

status=OK&waybill_type=A&waybill_id=9000000002,9000000013,9000000024,9000000035,9000000046,9000000050,9000000061,9000000072,9000000083,9000000094

12 碼託運單回傳資料

Response Data：

status=OK&waybill_type=A&waybill_id=900000000002,900000000013,900000000024,900000000035,900000000046,900000000050,900000000061,900000000072,900000000083,900000000094

其他說明

日後因應新業務導入託運單單號長度會有 10 碼、11 碼、12 碼情況，並會混合輸出的情況；及 waybill_type 也會因應業務不同而開發新的代碼種類。



申請速達託運單號碼 (起迄) (於 EGS 1.8.0 版後不支援此方式)

Request

- cmd=query_waybill_id_between
- customer_id=<string:契約客戶代號>
- waybill_type=<string:託運單類別 "A"-一般託運單|"B"-代收託運單>
- count=<int:數量>

Response

- status=<string:OK|ERROR>
- message=<string:錯誤訊息>
- waybill_type=<string:託運單類別 "A"-一般託運單|"B"-代收託運單>
- waybill_id_start=<string:起始速達託運單號>
- waybill_id_end=<string:結束速達託運單號>
- count=<int:回傳託運單號數量>

範例：申請 10 個託運單號碼。

Post URL:

<http://localhost:8800/egs>

Post Data:

cmd=query_waybill_id_between&customer_id=8066237502&waybill_type=A&count=10

Response Data:

status=OK&message=&waybill_type=A&waybill_id_start=9000000002&waybill_id_end=9000000094&count=10



5.6 查詢速達託運單號碼存量

Request

- cmd=query_waybill_id_remain
- customer_id= <string:契約客戶代號>
- waybill_type= <string:託運單類別 "A"=一般託運單|"B"=代收託運單|"G"=報值託運單>

Response

- status= <string:OK|ERROR>
- message= <string:錯誤訊息>
- waybill_type= <string:託運單類別 "A"=一般託運單|"B"=代收託運單|"G"=報值託運單>
- waybill_id_remain= <int:速達託運單號剩餘存量>

範例：查詢託運單號碼存量。

Post URL：

<http://localhost:8800/egs>

Post Data：

cmd=query_waybill_id_remain&customer_id=8066237502&waybill_type=A

Response Data：

status=OK&waybill_id_remain=1024&waybill_type=A



5.7 檢查五碼郵號是否可以使用指定的配達時段

Request

- cmd=test_delivery_timezone
- service_type=<int:配達時段>
- suda5_<int:編號>=<string:速達五碼郵遞區號>

Response

- status=<string:OK|ERROR>
- message=<string:錯誤訊息>
- delivery_timezone_=<int:可用的配達時段代碼>

範例：查詢 5 碼是否可以使用指定的配達時段。

Post URL：

http://localhost:8800/egs

Post Data：

cmd=test_delivery_timezone&service_type=2&suda5_1=11501&suda5_2=99999

Response Data：

status=OK&delivery_timezone_1=2&delivery_timezone_2=2

[備註]

- 1.Request 的參數 suda5_* 不可以是空白字串。
- 2.suda5_* 欄位可多筆上傳，delivery_timezone 會回應與 suda5_* 對應序號的欄位。
- 3.目前配達時段 1,2,3,4 都不需要檢查，~~只有配達時段 5(夜配)需要檢查指定地區。~~
- 4.如果指定的五碼郵號地區無法支援 service_type 所指定的配達時段，系統將會回應該地區可使用的替代配達時段代碼。
- 5.若 5 碼是空白時，請傳入 99999 或是 00000 進行解析

5.8 傳送單筆託運單資料

Request

- cmd=transfer_waybill
- customer_id= <string:連線契客代號>
- tracking_number= <string:託運單號碼>
- order_no= <string:訂單編號>
- receiver_name= <string:收件人姓名>
- receiver_address= <string:收件人地址>
- receiver_suda5= <string:收件人地址的速達五碼郵遞區號>
- receiver_mobile= <string:收件人行動電話>
- receiver_phone= <string:收件人電話>
- sender_name= <string:寄件人姓名>
- sender_address= <string:寄件人地址>
- sender_suda5= <string:寄件人地址的速達五碼郵遞區號>
- sender_phone= <string:寄件人電話>
- product_price= <string:代收貨款金額>
- product_name= <string:品名>
- comment= <string:備註>
- package_size= <string:尺寸 "0001"=60cm,"0002"=90cm,"0003"=120cm,"0004"=150cm>
- temperature= <string:溫層 "0001"=常溫|"0002"=冷藏|"0003"=冷凍>
- distance= <string:距離 "00"=同縣市|"01"=外縣市|"02"=離島>
- delivery_date= <string:指定配達日期>
- delivery_timezone= <string:指定配達時段
"1"=9~12 時|"2"=12~17 時|"3"=17~20 時|"4"=不限時|~~"5"=20~21 時~~ >
- create_time= <string:建立時間>
- print_time= <string:列印時間>
- account_id= <string:託運單帳號>
- member_no= <string:會員編號>
- **taxin= <string: 關稅(進口)> (限用海外地區使用)**
- **insurance= <string: 報值金額>**

Response

- status= <string:OK|ERROR>
- message= <string:錯誤訊息>

範例：傳輸 1 筆託運單資料。

Post URL：

<http://localhost:8800/egs>



Post Data :

cmd=transfer_waybill&customer_id=8066237501&tracking_number=1127598154&order_no=11161-00001&receiver_name=%E8%A8%B1%E5%85%88%E7%94%9F&receiver_address=%E5%8F%B0%E5%8C%97%E5%B8%82%E5%A3%AB%E6%9E%97%E5%8D%97%E4%B8%AD%E6%AD%A3%E8%B7%AF601%E8%99%9F9%E6%A8%93%E4%B9%8B1&receiver_suda5=11111&receiver_mobile=0912-345678&receiver_phone=12345678&sender_name=%E7%B5%B1%E4%B8%97%E9%9F%E9%81%94&sender_address=%E5%8F%B0%E5%8C%97%E5%B8%82%E5%8D%97%E6%B8%AF%E5%8D%97%E9%87%8D%E9%99%BD%E8%B7%AF200%E8%99%9F4%E6%A8%93&sender_suda5=11501&sender_phone=02-27887887&product_price=0&product_name=egs%20%E6%B8%AC%E8%A9%A6%E8%A8%97%E9%81%8B%E5%96%AE&comment=%E8%A8%97%E9%81%8B%E5%96%AE%E7%94%B1%20egs%20%E4%B8%8A%E5%82%B3&package_size=0001&temperature=0001&distance=00&delivery_date=2011-04-25&delivery_timezone=4&create_time=2011-04-24%2011:27:59&print_time=2011-04-24%2011:27:59&account_id=0921150615&member_no=ICQ162233118&taxin=0

Response Data :

status=OK&message=

customer_id=8066237501
tracking_number=1127598154
order_no=11161-00001
receiver_name=許先生
receiver_address=台北市士林區中正路 601 號 9 樓之 1
receiver_suda5=11111
receiver_mobile=0912-345678
receiver_phone=12345678
sender_name=統一速達
sender_address=台北市南港區重陽路 200 號 4 樓
sender_suda5=11501
sender_phone=02-27887887
product_price=0
product_name=egs 測試託運單
comment=託運單由 egs 上傳
package_size=0001
temperature=0001
distance=00
delivery_date=2011-04-25
delivery_timezone=4
create_time=2011-04-24 11:27:59
print_time=2011-04-24 11:27:59
account_id=0921150615
member_no=ICQ162233118
taxin=0
insurance=0



5.9 查詢速達 7 碼條碼資料(+符號+轉運站 + 速達 5 碼)

Request

- cmd=query_suda7
- address_<int:編號>=<string:地址>

Response

- status=<string:OK|ERROR>
- message=<string:錯誤訊息>
- suda7_<int:序號>=<string:速達五碼郵遞區號>

(注意！除了 cmd 外，address_*欄位可多筆上傳，suda7 會回應與 address_*對應序號的欄位，但需注意有最大 200 筆的上限限制，建議逐筆查詢)

範例 1：當我只要一筆地址需要 egs 幫我解析 7 碼條碼資料，取得「桃園縣桃園市中山北路 17 之 1 號 2 樓」的速達五碼郵號。

Post URL：

`http://localhost:8800/egs`

Post Data：

```
cmd=query_suda7&address_1=%E6%A1%83%E5%9C%92%E7%B8%A3%E6%A1%83%E5%9C%92%E5%B8%82%E4%B8%AD%E5%B1%B1%E5%8C%97%E8%B7%AF17%E4%B9%8B1%E8%99%9F2%E6%A8%93
```

Response Data：

```
status=OK&suda7_1=+3233042
```

範例 2：當我有多筆地址，想要一次性向 egs 取得解析 7 碼條碼資料的結果，取得「桃園縣桃園市中山北路 17 之 1 號 2 樓」與「新北市新莊區昌明街 48 號 3 樓」的速達五碼郵號。(注意！用這種方式，我們最多一次只能向 egs 要求分析 200 筆地址，建議逐筆查詢)

Post URL：

`http://localhost:8800/egs`

Post Data：

```
cmd=query_suda7&address_1=%E6%A1%83%E5%9C%92%E7%B8%A3%E6%A1%83%E5%9C%92%E5%B8%82%E4%B8%AD%E5%B1%B1%E5%8C%97%E8%B7%AF17%E4%B9%8B1%E8%99%9F2%E6%A8%93&address_2=%E6%96%B0%E5%8C%97%E5%B8%82%E6%96%B0%E8%8E%8A%E5%8D%80%E6%98%8C%E6%98%8E%E8%A1%9748%E8%99%9F3%E6%A8%93
```

Response Data：



status=OK&suda7_1=+3233042&suda7_2=+3124273

範例 3：當我有超過 200 筆以上的地址，想要 egs 幫我解析 7 碼條碼資料時，系統該怎麼呼叫 egs 呢？其實前端系統只要將多筆地址，用迴圈方式一筆一筆向 egs 要求解析即可達到目的!!取得「桃園縣桃園市中山北路 17 之 1 號 2 樓」、「新北市新莊區昌明街 48 號 3 樓」、「台北市南京東段三段 261 號」的速達 7 碼條碼資料。

說明：處理方式等同於範例 1，但需要先端程式，先將要解析的地址用迴圈的方式，一筆一筆呼叫 egs 進行解析。

```
for (int i = 0; i < 需要解析7碼條碼郵號的筆數; i++)  
{  
    回傳的7碼 = query_suda7&address_1=這次需解析的地址  
}
```

[備註]

- 1 Request 的參數 address_* 不可以是空白字串。
- 2 address_* 欄位可多筆上傳，suda7 會回應與 address_* 對應序號的欄位。
- 3 若地址是空白時，請傳入 noaddress 進行解析



5.10 查詢速達 7 碼資料(轉運站及速達 5 碼有 dash 分隔)

Request

- cmd=query_suda7_dashv2 (修改傳入指令增增 v2)
- address_<int:編號>=<string:地址>

Response

- status=<string:OK|ERROR>
- message=<string:錯誤訊息>
- suda7_<int:序號>=<string:速達五碼郵遞區號>

(注意！除了 cmd 外，address_* 欄位可多筆上傳，suda7 會回應與 address_* 對應序號的欄位，但需注意有最大 200 筆的上限限制，建議逐筆查詢)

範例 1：當我只有一筆地址需要 egs 幫我解析 7 碼資料，取得「桃園縣桃園市中山北路 17 之 1 號 2 樓」的速達五碼郵號。

Post URL：

http://localhost:8800/egs

Post Data：

```
cmd=query_suda7_dashv2
&address_1=%E6%A1%83%E5%9C%92%E7%B8%A3%E6%A1%83%E5%9C%92%E5%B8%82%E4%B8%AD%E5%B1%B1%E5%8C%97%E8%B7%AF17%E4%B9%8B1%E8%99%9F2%E6%A8%93
```

Response Data：

```
status=OK&suda7_1=32-330-42-A (回傳郵號會增加一碼英文字為理貨區碼)
```

範例 2：當我有多筆地址，想要一次性向 egs 取得解析 7 碼條碼資料的結果，取得「桃園縣桃園市中山北路 17 之 1 號 2 樓」與「新北市新莊區昌明街 48 號 3 樓」的速達五碼郵號。(注意！用這種方式，我們最多一次只能向 egs 要求分析 200 筆地址，建議逐筆查詢)

Post URL：

http://localhost:8800/egs

Post Data：

```
cmd=query_suda7_dashv2&address_1=%E6%A1%83%E5%9C%92%E7%B8%A3%E6%A1%83%E5%9C%92%E5%B8%82%E4%B8%AD%E5%B1%B1%E5%8C%97%E8%B7%AF17%E4%B9%8B1%E8%99%9F2%E6%A8%93&address_2=%E6%96%B0%E5%8C%97%E5%B8%82%E6%96%B0%E8%8E%8A%E5%8D%80%E6%98%8C%E6%98%8E%E8%A1%9748%E8%99%9F3%E6%A8%93
```

Response Data：

```
status=OK&suda7_1=32-330-42-A&suda7_2=31-242-73-B
```



範例 3：當我有超過 200 筆以上的地址，想要 egs 幫我解析 7 碼資料時，系統該怎麼呼叫 egs 呢？其實前端系統只要將多筆地址，用迴圈方式一筆一筆向 egs 要求解析即可達到目的!!取得「桃園縣桃園市中山北路 17 之 1 號 2 樓」、「新北市新莊區昌明街 48 號 3 樓」、「台北市南京東段三段 261 號」的速達 7 碼資料。

說明：處理方式等同於範例 1，但需要先端程式，先將要解析的地址用迴圈的方式，一筆一筆呼叫 egs 進行解析。

```
for (int i = 0; i < 需要解析7碼郵號的筆數; i++)  
{  
    回傳的7碼 = query_suda7_dashv2&address_1=這次需解析的地址  
}
```

[備註]

- 1 Request 的參數 address_* 不可以是空白字串。
- 2 address_* 欄位可多筆上傳，suda7 會回應與 address_* 對應序號的欄位。
- 3 若地址是空白時，請傳入 noaddress 進行解析



5.11 查詢速達收寄件人地址的距離

Request

- cmd=query_distance
- suda5_senderpostcode_<int:編號>=<string:寄件人郵號>
- suda5_customerpostcode_<int:編號>=<string:收件人郵號>

Response

- status=<string:OK|ERROR>
- message=<string:錯誤訊息>
- distance_<int:序號>=<string:速達距離>

(注意！除了 cmd 外，suda5_senderpostcode_*欄位及 suda5_customerpostcode_*欄位可多筆上傳(但收寄件人一次需一組傳入，distance 會回應與 suda5_senderpostcode_*及 suda5_customerpostcode_*對應序號的欄位，但需注意有最大 200 筆的上限限制，建議逐筆查詢，distance 回傳的意義：00-同縣市 01-外縣市 02-離島)

範例 1：當我只有一組收寄件人郵號時，需要 egs 幫我解析收寄件人距離，取得寄件人郵號「10302」及收件人郵號「893」的速達距離。

Post URL：

`http://localhost:8800/egs`

Post Data：

`cmd=query_distance&suda5_senderpostcode_1=10302&suda5_customerpostcode_1=893`

Response Data：

`status=OK &message=&distance_1=02`

範例 2：當我有多組收寄人郵號時，想要一次性向 egs 取得解析距離的結果，取得寄件人郵號「10302」、收件人郵號「893」與寄件人郵號「10573」、收件人郵號「10302」的速達距離。(注意！用這種方式，我們最多一次只能向 egs 要求分析 200 筆地址，建議逐筆查詢)

Post URL：

`http://localhost:8800/egs`

Post Data：

`cmd=query_distance&suda5_senderpostcode_1=10302&suda5_customerpostcode_1=893&suda5_senderpostcode_2=10573&suda5_customerpostcode_2=10302`

Response Data：

`status=OK&message=&distance_1=02&distance_2=00`



範例 3：當我有超過 200 筆以上的收寄人郵號，想要 egs 幫我解析距離資料時，系統該怎麼呼叫 egs 呢？其實前端系統只要將多組收寄件人郵號，用迴圈方式一筆一筆向 egs 要求解析即可達到目的!!取得寄件人郵號「10302」及收件人郵號「893」、寄件人郵號「10573」及收件人郵號「10302」、寄件人郵號「893」及收件人郵號「893」的速達距離資料。

說明：處理方式等同於範例 1，但需要先端程式，先將要解析的收寄件人郵號用迴圈的方式，一筆一筆呼叫 egs 進行解析。

```
for (int i = 0; i < 需要解析距離的收寄件人筆數; i++)
{
    回傳的距離 = query_distance&suda5_senderpostcode_1=寄件人郵號
    &suda5_customerpostcode_1=收件人郵號
}
```

[備註]

- 1 Request 的參數 suda5_senderpostcode_* 及 suda5_customerpostcode_* 不可以是空白字串。
- 2 suda5_senderpostcode_* 及 suda5_customerpostcode_* 欄位可多筆上傳，distance 會回應與 suda5_senderpostcode_* 及 suda5_customerpostcode_* 對應序號的欄位。
- 3 若收寄件人郵號是空白時，請傳入 99999 或是 00000 進行解析。

5.12 產生託運單 QR Code

Request

- cmd= gen_qrcode
- customer_id= <string:寄件契客代號>
- tracking_number= <string:託運單號碼>
- product_price= <string:代收貨款金額>
- temperature= <string:溫層 "01"=常溫|"02"=冷藏|"03"=冷凍>
- package_size= <string:尺寸 "01"=60cm,"02"=90cm,"03"=120cm,"04"=150cm>
- receiver_suda5= <string:收件人地址的速達五碼郵遞區號 "5 碼長度，不需間隔符號">
- delivery_date= <string:指定配達日期 "YYYYMMDD-長度為 8 碼，不需間隔符號">
- delivery_timezone= <string:指定配達時段 "01"=13 點前|"02"=14~18 點|"04"=不指定>
- insurance= <string: 報值金額>
- print_invoice= <string: 發票是否列印>(未啟用)
- print_donation= <string: 是否捐贈>(未啟用)
- invoice_uni= <string:發票統一編號>(未啟用)
- common= <string: 共通性載具>(未啟用)



Response

- 回傳圖檔為 JPG 格式
- 回應檔頭內容

HTTP/1.1 200 OK

Connection: close

Content-Disposition: attachment; filename="qrcode_託運單號.jpg";

Content-Type: application/jpeg

Content-Length: 2610

Server: Ezcat Gateway System

範例：傳輸 1 筆託運單資料。

Post URL：

<http://localhost:8800/egs>

Post Data：

cmd=gen_qrcode&customer_id=1265635401&tracking_number=620003402646&product_price=20000&temperature=01&package_size=02&receiver_suda5=11534A&delivery_date=20190430&delivery_timezone=04&insurance=0&print_invoice=Y&print_donation=N&invoice_uni=12656354&common=/.Y12456

Response Data：

HTTP/1.1 200 OK

Connection: close

Content-Disposition: attachment; filename="qrcode_620003402646.jpg";

Content-Type: application/jpeg

Content-Length: 2610

Server: Ezcat Gateway System

範例中的託運單內容是

```
customer_id=1265635401
tracking_number=620003402646
product_price=20000
temperature=01
package_size=02
receiver_suda5=11534A
delivery_date=20190430
delivery_timezone=04
insurance =0
print_invoice =Y
print_donation =N
invoice_uni =12656354
common =/.Y12456
```

產生的 QR Code 圖檔 (已放大 2 倍)



QR Code 的內容

```
01|620003402646|10|126563540100|N|20000|01|01|02|11534A|20190430|04||0|||||||
```