

Annex A

1. API DOCUMENT

This API document is a technical content deliverable, containing instructions about how to effectively use and integrate with API of the centralized database. It's a concise reference manual containing all the information required to work with the API, with details about the functions, return types, arguments.

The API document will focus on API method request for System Management Service and Transaction Service for Supplier.

The API is using the following API method.

API Methods	Functions
GET	Retrieve data from server at the specified resource.
POST	Send data to the API server to create a resource.
PUT	Send data to the API to update a resource.
DELETE	Delete the resource at the specified path parameter

The summary of API method requests is as below, each API method request will be explained in this API document.

Path Parameter	GET	POST	PUT	DELETE
/machines	✓	✓		
/machines/{id}		✓	✓	✓
/locations	✓	✓		
/locations/{id}		✓	✓	✓
/user/{id}	✓			
/transaction		✓		

The list of response code is as follows.

Code	Description
200	Search results matching criteria.
201	Action completed.
400	Bad input parameter.
404	Bad input parameter. Please make sure the input type is correct and ID should be unique.
409	An existing item already exists.

The Schemas of each resources, Machine, Location, User and Transaction are as below.

Machine

Parameter	Type	Description
machineID	string	The machine ID of registered machine.
locationID	string	The location ID of registered machine.
machineTyp	string	The machine type of register machine. [smart bin/ smart balance/ gift redemption unit]
serialNumber	string	The serial number of register machine.
startTime	string	The operation start time in HH:MM format.
closeTime	string	The operation end time in HH:MM format.
remarks	string	Any remarks for the register machine.

Location

Parameter	Type	Example
locationID	string	The location ID of register location.
latitude	Number(\$float)	The latitude of register location.
longitude	Number(\$float)	The longitude of register location.
address	string	The address of register location.
remarks	string	Any remarks for the register location.

User

Parameter	Type	Example
userID	string	The user ID.
legacyUserID	string	The legacy user ID.
status	boolean	The status of user account. [true for valid account/ false for invalid account]

Transaction

Parameter	Type	Example
transactionID	string	123456789A
machineID	string	TKTH200-1
userID	string	abc1234567
legacyUserID	string	8828907917
locationID	string	TKT
transactionType	string	recycle or redemption
weight	number(\$float)	3

greenScore	number(\$float)	-20
status	string	success
remarks	string	This is a manual transaction by operator operator ID#1234.

2. MACHINE CRUD OPERATIONS

Machine CRUD operations are API method request for generic machine data.

1 Path Parameter: /machines

API Method:

- GET [Get machines data by range.]
- POST [Add a machine.]

Parameters for GET/machines:

Parameter	Type	Required	Description
skip	integer(\$int32)	No	Number of records to skip for pagination.
limit	integer(\$int32)	No	Maximum number of records to return.

Request URL for GET/machines:

To get all machine data.

<http://18.162.211.65:5200/machines>

To get machine data in specific range.

(eg. number of records to skip = 0, maximum number of records to return = 0)

<http://18.162.211.65:5200/machines?skip=0&limit=0>

Example of machine data for GET/machines:

(eg. number of records to skip = 0, maximum number of records to return = 0)

```
curl -X 'GET' \
  'http://18.162.211.65:5200/machines?skip=0&limit=0' \
  -H 'accept: application/json'
```

Example of Response Body for GET/machines:

```
{
  "success": true,
  "data": [
    {
```

```

    "_id": "611b31ea9e86ff9c24435194",
    "machineID": "TKTH200-1",
    "locationID": "TKT",
    "machineType": "smart bin/ smart balance/ gift
redemption unit",
    "serialNumber": "xxxyyyzzz",
    "startTime": "09:00",
    "closeTime": "23:00",
    "remarks": "Scrap. Out of services",
    "createdAt": "2021-08-17T03:50:02.979Z",
    "updatedAt": "2021-08-17T03:50:02.979Z",
    "__v": 0
  }
],
"total": "1",
"page": "1",
"last_page": "1"
}

```

Parameters for POST/machines:

Parameter	Type	Required	Description
machineID	string	Yes	The ID of machine.
locationID	string	Yes	The location ID of machine.
machineType	string	Yes	The type of machine [smart bin/ smart balance/ gift redemption unit].
serialNumber	string	Yes	The serial number of machine.
startTime	string	Yes	The operation start time of machine.
closeTime	string	Yes	The operation end time of machine.
remarks	string	No	Any remarks.

Request URL for POST/machines:

<http://18.162.211.65:5200/machines>

Example of machine data for POST/machines:

```

curl -X 'POST' \
  'http://18.162.211.65:5200/machines' \

```

```
-H 'accept: */*' \
-H 'Content-Type: application/json'
\ -d '{
  "machineID": "HKPC-001",
  "locationID": "HKPC",
  "machineType": "smart bin",
  "serialNumber": "xxxyyyzzz",
  "startTime": "09:00",
  "closeTime": "23:00",
  "remarks": "Example"
}'
```

Example of Response Body for POST/machines:

```
{
  "success": true,
  "data": {
    "_id": "611f1f379e86ff9c2443523a",
    "machineID": "HKPC-001",
    "locationID": "HKPC",
    "machineType": "smart bin",
    "serialNumber": "xxxyyyzzz",
    "startTime": "09:00",
    "closeTime": "23:00",
    "remarks": "Example",
    "createdAt": "2021-08-20T03:19:19.275Z",
    "updatedAt": "2021-08-20T03:19:19.275Z",
    "__v": 0
  }
}
```

or

```
{
  "success": false,
  "message": "Failed to add machine"
}
```

2 **Path Parameter: /machines/{id}**

API Method:

- GET [Search machine data by ID]
- PUT [Update machine data by ID]
- DELETE [Delete machine by ID]

Parameters for GET/machines/{id}:

Parameter	Type	Required	Description
id	string	Yes	machineID

Request URL for GET/machines/{id}:

To get machine data for specific machine.

[http://18.162.211.65:5200/machines/\[machineID\]](http://18.162.211.65:5200/machines/[machineID])

Example of machine data for GET/machines/{id}:

(eg. To GET data of machineID TKTH200-1.)

```
curl -X 'GET' \
  'http://18.162.211.65:5200/machines/TKTH200-1' \
  -H 'accept: application/json'
```

Example of Response Body for GET/machines/{id}:

```
{
  "success": true,
  "data": [
    {
      "_id": "611b31ea9e86ff9c24435194",
      "machineID": "TKTH200-1",
      "locationID": "TKT",
      "machineType": "smart bin/ smart balance/ gift
redem ption unit",
      "serialNumber": "xxxyyyzzz",
      "startTime": "09:00",
      "closeTime": "23:00",
      "remarks": "Scrap. Out of services",
      "createdAt": "2021-08-17T03:50:02.979Z",
```

```
    "updatedAt": "2021-08-17T03:50:02.979Z",  
    "__v": 0  
  }  
]  
}
```


Parameters for PUT/machines/{id}:

Parameter	Type	Required	Description
id	string	Yes	Lookup the machine by machineID
locationID	string	No	Update the locationID.
machineType	string	No	Update the machineType.
serialNumber	string	No	Update the serial number of the machine.
startTime	string	No	Update operation time. Format: HH:MM
closeTime	string	No	Update closing time. Format: HH:MM
remarks	string	No	Update any remarks.

Request URL for PUT/machines/{id}:

To update machine data for specific machine.

[http://18.162.211.65:5200/machines/\[machineID\]](http://18.162.211.65:5200/machines/[machineID])

Example of machine data for PUT/machines/{id}:

(eg. To update locationID, machineType, closeTime for machineID HKPC-001.)

```
curl -X 'PUT' \
  'http://18.162.211.65:5200/machines/HKPC-001?locationID=HKPC&machineType=balance&closeTime=22%3A00' \
  -H 'accept: application/json'
```

Example of Response Body for PUT/machines/{id}:

```
{
  "success": true,
  "data": {
    "_id": "611b31ea9e86ff9c24435194",
    "machineID": "HKPC-001",
    "locationID": "TKT",
    "machineType": "smart balance",
    "serialNumber": "xxxyyyzzz",
    "startTime": "09:00",
```

```
"closeTime": "22:00",  
"remarks": "Scrap. Out of services",  
"createdAt": "2021-08-17T03:50:02.979Z",  
"updatedAt": "2021-08-17T09:24:07.532Z",  
"__v": 0  
},  
"message": "Machine updated successfully"  
}
```

Parameters for DELETE/machines/{id}:

Parameter	Type	Required	Description
id	string	Yes	machineID

Request URL for DELETE/machines/{id}:

To delete machine data for specific machine.

<http://18.162.211.65:5200/machines/> [machineID]

Example of machine data for DELETE/machines/{id}:

(eg. To DELETE machineID HKPC-001.)

```
curl -X 'DELETE' \
  'http://18.162.211.65:5200/machines/HKPC-001' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "machineID": "TKTH200-1"
  }'
```

Example of Response Body for DELETE/machines/{id}:

```
{
  "success": true,
  "message": "Deleted Machine"
}
```

Or

3 LOCATION CRUD OPERATIONS

Location CRUD operations are API method request for generic location data.

4 Path Parameter: /locations

API Method:

- GET [Get location data by range.]
- POST [Add a location.]

Parameters for GET/locations:

Parameter	Type	Required	Description
skip	integer(\$int32)	No	Number of records to skip for pagination
limit	integer(\$int32)	No	Maximum number of records to return

Request URL for GET/locations:

To get all location data.

<http://18.162.211.65:5200/locations>

To get location data in specific range.

(eg. number of records to skip = 0, maximum number of records to return = 0)

<http://18.162.211.65:5200/locations?skip=0&limit=1>

Example of location data for GET/locations:

(eg. number of records to skip = 0, maximum number of records to return = 0)

```
curl -X 'GET' \  
  'http://18.162.211.65:5200/locations?skip=0&limit=0' \  
  -H 'accept: application/json'
```

Example of Response Body for GET/locations:

```
{  
  "success": true,  
  "data": [  
    {  
      "id": 1,  
      "name": "Location 1",  
      "lat": 12.3456,  
      "lon": 78.9012,  
      "created_at": "2023-01-01T00:00:00Z",  
      "updated_at": "2023-01-01T00:00:00Z"  
    },  
    {  
      "id": 2,  
      "name": "Location 2",  
      "lat": 23.4567,  
      "lon": 89.0123,  
      "created_at": "2023-01-02T00:00:00Z",  
      "updated_at": "2023-01-02T00:00:00Z"  
    },  
    {  
      "id": 3,  
      "name": "Location 3",  
      "lat": 34.5678,  
      "lon": 90.1234,  
      "created_at": "2023-01-03T00:00:00Z",  
      "updated_at": "2023-01-03T00:00:00Z"  
    },  
    {  
      "id": 4,  
      "name": "Location 4",  
      "lat": 45.6789,  
      "lon": 01.2345,  
      "created_at": "2023-01-04T00:00:00Z",  
      "updated_at": "2023-01-04T00:00:00Z"  
    },  
    {  
      "id": 5,  
      "name": "Location 5",  
      "lat": 56.7890,  
      "lon": 12.3456,  
      "created_at": "2023-01-05T00:00:00Z",  
      "updated_at": "2023-01-05T00:00:00Z"  
    },  
    {  
      "id": 6,  
      "name": "Location 6",  
      "lat": 67.8901,  
      "lon": 23.4567,  
      "created_at": "2023-01-06T00:00:00Z",  
      "updated_at": "2023-01-06T00:00:00Z"  
    },  
    {  
      "id": 7,  
      "name": "Location 7",  
      "lat": 78.9012,  
      "lon": 34.5678,  
      "created_at": "2023-01-07T00:00:00Z",  
      "updated_at": "2023-01-07T00:00:00Z"  
    },  
    {  
      "id": 8,  
      "name": "Location 8",  
      "lat": 89.0123,  
      "lon": 45.6789,  
      "created_at": "2023-01-08T00:00:00Z",  
      "updated_at": "2023-01-08T00:00:00Z"  
    },  
    {  
      "id": 9,  
      "name": "Location 9",  
      "lat": 90.1234,  
      "lon": 56.7890,  
      "created_at": "2023-01-09T00:00:00Z",  
      "updated_at": "2023-01-09T00:00:00Z"  
    },  
    {  
      "id": 10,  
      "name": "Location 10",  
      "lat": 01.2345,  
      "lon": 67.8901,  
      "created_at": "2023-01-10T00:00:00Z",  
      "updated_at": "2023-01-10T00:00:00Z"  
    }  
  ]  
}
```

```
{
  "_id": "611b3513d3ce4e5ddcaf3c17",
  "locationID": "SHATIN-01",
  "latitude": 22.3924,
  "longitude": 114.2088,
  "address": "10 On Ping Street, Shek Mun"
}
],
"total": "33",
"page": "1",
"last_page": "33"
}
```

Parameters for POST/locations:

Parameter	Type	Required	Description
locationID	string	Yes	The location ID.
latitude	Number(\$float)	Yes	The latitude of location.
longitude	Number(\$float)	Yes	The longitude of location.
address	string	Yes	The address of location.
remarks	string	No	Any remarks.

Request URL for POST/locations:

<http://18.162.211.65:5200/locations>

Example of location data for POST/locations:

```
curl -X 'POST' \
  'http://18.162.211.65:5200/locations' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "locationID": "HKPC-001",
    "latitude": 22.3924,
    "longitude": 114.20886,
```

```
"address": "HKPC Building, 78 Tat Chee Avenue, Kowloon, Hong Kong",  
  "remarks": "string"  
'
```

Example of Response Body for POST/locations:

```
{  
  "success": true,  
  "data": {  
    "_id": "611f244c9e86ff9c24435245",  
    "locationID": "HKPC-001",  
    "latitude": 22.3924,  
    "longitude": 114.20886,  
    "address": "HKPC Building, 78 Tat Chee Avenue, Kowloon, Hong Kong",  
    "remarks": "string",  
    "createdAt": "2021-08-20T03:41:00.262Z",  
    "updatedAt": "2021-08-20T03:41:00.262Z",  
    "__v": 0  
  }  
}
```

or

```
{  
  "success": false,  
  "message": "Failed to add location"  
}
```

5 Path Parameter: /locations/{id}

API Method:

- GET [Search location data by ID]
- PUT [Update location data by ID]
- DELETE [Delete location by ID]

Parameters for GET/locations/{id}:

Parameter	Type	Required	Description
id	string	Yes	Location ID

Request URL for GET/locations/{id}:

To get machine data for specific machine.

[http://18.162.211.65:5200/machines/\[locationID\]](http://18.162.211.65:5200/machines/[locationID])

Example of location data for GET/locations/{id}:

(eg. To get location data for locationID HKPC-001.)

```
curl -X 'GET' \  
  'http://18.162.211.65:5200/locations/HKPC-001' \  
  -H 'accept: application/json'
```

Example of Response Body for GET/locations/{id}:

```
{  
  "success": true,  
  "data": [  
    {  
      "_id": "611f244c9e86ff9c24435245",  
      "locationID": "HKPC-001",  
      "latitude": 22.3924,  
      "longitude": 114.20886,  
      "address": "HKPC Building, 78 Tat Chee Avenue,  
Kowloon, Hong Kong",  
      "remarks": "string",  
      "createdAt": "2021-08-20T03:41:00.262Z",  
    }  
  ]  
}
```

```
    "updatedAt": "2021-08-20T03:41:00.262Z",  
    "__v": 0  
  }  
]  
}
```

Or

```
{  
  "success": false,  
  "message": "Failed to get location"  
}
```


Parameters for PUT/locations/{id}:

Parameter	Type	Required	Description
locationID	string	Yes	Update the location ID.
latitude	Number(\$float)	No	Update the of location.
longitude	Number(\$float)	No	Update the of location.
address	string	No	Update the of location.
remarks	string	No	Update any remarks.

Request URL for PUT/locations/{id}:

To update machine data for specific location.

[http://18.162.211.65:5200/locations/\[locationID\]](http://18.162.211.65:5200/locations/[locationID])

Example of location data for PUT/locations/{id}:

(eg. To PUT location data for locationID HKPC-001, remarks = Example.)

```
curl -X 'PUT' \  
  'http://18.162.211.65:5200/locations/HKPC-001?remarks=Example' \  
  -H 'accept: application/json'
```

Example of Response Body for PUT/locations/{id}:

```
{  
  "success": true,  
  "data": {  
    "_id": "611f244c9e86ff9c24435245",  
    "locationID": "HKPC-001",  
    "latitude": 22.3924,  
    "longitude": 114.20886,  
    "address": "HKPC Building, 78 Tat Chee Avenue,  
Kowloon, Hong Kong",  
    "remarks": "Example",  
    "createdAt": "2021-08-20T03:41:00.262Z",  
    "updatedAt": "2021-08-20T03:55:10.319Z",  
    "__v": 0  
  },  
}
```

```
"message": "Location updated successfully"
}
```

Or

```
{
  "success": false,
  "message": "Cannot find location"
}
```

Parameters for DELETE/locations/{id}:

Parameter	Type	Required	Description
id	string	Yes	locationID

Request URL for DELETE/locations/{id}:

To delete machine data for specific location.

[http://18.162.211.65:5200/locations/\[locationID\]](http://18.162.211.65:5200/locations/[locationID])

Example of DELETE location ID HKPC-001:

```
curl -X 'DELETE' \
  'http://18.162.211.65:5200/locations/ HKPC-001' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "locationID": "S"
  }'
```

Example of Response Body for /locations/{id}:

```
{
  "success": true,
  "message": "Deleted location "
}
```

Or

```
{
  "success": false,
  "message": "Cannot find location"
}
```

6 TRANSACTION CRUD OPERATIONS

Transaction Crud Operations are API method request for generic transaction data.

7 Path Parameter: /transaction

API Method:

- POST [Add a transaction.]

Parameters for POST/transaction:

Parameter	Type	Required	Description
machineID	string	Yes	The machine ID.
userID	string	Yes	The user ID.
legacyUserID	string	Yes	The legacy user ID.
locationID	string	Yes	The location ID.
transactionType	string	Yes	The transaction type. [recycle or redemption]
weight	number(\$float)	Yes	The weight of recyclable.
greenScore	number(\$float)	Yes	The change of green score. [Positive for recycle, negative for redemption]
status	string	Yes	The status of transaction from machine.
remarks	string	No	Any remarks.

Request URL for POST/transaction:

<http://18.162.211.65:5000/transaction>

Example of location data for POST/transaction:

```
curl -X 'POST' \
  'http://18.162.211.65:5000/transaction' \
  -H 'accept: */*' \
  -H 'Content-Type: application/json' \
  -d '{
    "machineID": "00",
    "userID": "abc1234567",
    "legacyUserID": "8828907917",
    "locationID": "TKT",
```

```
"transactionType": "redemption",
"recycleableType": "paper",
"transactionTime": "2021-08-29T09:12:33.001Z",
"weight": 3,
"greenScore": 20,
"status": "success",
"remarks": ""
}'
```

Example of Response Body for POST/transaction:

```
{
  "success": true,
  "data": {
    "_id": "611f6ca1d56ba7a9df6d07eb",
    "machineID": "00",
    "legacyUserID": 8828907917,
    "locationID": "TKT",
    "transactionType": "redemption",
    "recycleableType": "paper",
    "transactionTime": "2021-08-29T09:12:33.001Z",
    "weight": 3,
    "greenScore": 20,
    "status": "success",
    "remarks": "",
    "createdAt": "2021-08-20T08:49:37.934Z",
    "updatedAt": "2021-08-20T08:49:37.934Z",
    "__v": 0
  }
}
```

or

```
{
  "success": false,
  "message": "Failed to add transaction."
}
```

8 USER CRUD OPERATIONS

User Crud Operations are API method request for generic user data.

9 Path Parameter: /user/{id}

API Method:

- GET [Search user by ID.]

Parameters for POST/ user/{id}:

Parameter	Type	Required	Description
userID	string	Yes	The user ID.

Request URL for GET/ user/{id}:

[http://18.162.211.65:5000/user/\[userID\]](http://18.162.211.65:5000/user/[userID])

Example of location data for GET/ user/{id}:

```
curl -X 'GET' \
  'http://18.162.211.65:5000/user/0' \
  -H 'accept: application/json'
```

Example of Response Body for GET/ user/{id}:

```
{
  "success": true,
  "data": []
}
```

or

```
{
  "success": false,
  "message": "Failed to get user."
}
```



Smart bin system and HKPC API test

Testing server IP: <https://hkpcdev.greenycoins.gov.hk>

Server port: 5000

Company name: _____

Tested by: _____

Test ID	Test description	Test procedure	Acceptance criteria	Pass	Partially accepted	Fail	Remarks
1	Check the membership and open the lid for valid member	1. Smart bin sends the request to HKPC server with valid member ID GET request: <code>https://hkpcdev.greenycoins.gov.hk:5000/user/{memberID}</code> 2. Smart bin receives the response from HKPC server with message ["success": true, "data":{"memberID": "XXX", "membership": true}]	Smart bin opens the lid to valid member after membership validation check.				
2	Check the membership and no action for invalid member	1. Smart bin sends the request to HKPC server with valid member ID GET request: <code>https://hkpcdev.greenycoins.gov.hk:5000/user/{memberID}</code> 2. Smart bin receives the response from HKPC server with message ["success": false]	Smart bin takes no actions for the invalid member.				
3	Send the normal transaction to database through HKPC API	Recycleable collected and Smart bin sends the transaction to HKPC database POST request: <code>https://hkpcdev.greenycoins.gov.hk:5000/transaction/recycle</code> Request body: { "transactionType": "RECYCLE", "machineID": "TKTH200-1", "memberID": "To be provided", "locationID": "TKT", "transactionTime": "2021-08-29T09:12:33.001Z", "recycleableType": "GB", "weight": 3, "greenScore": 1.5, "status": "valid", "remarks": "njl" }	The transaction record can be located in HKPC database.				