

🌐 Queue Display API Specification

🔐 Base URL

arduino

Copy

Edit

https://api.makati.gov.ph/queue

📌 1. Get All Current Queue Statuses

GET /status

📥 Request:

bash

Copy

Edit

GET /status

📤 Response:

json

Copy

Edit

[

{

"counter": 1,

"code": "BP-002",

"service": "Business Permit",

"status": "Now Serving"

},

{

"counter": 2,

"code": "RT-0012",

"service": "Real Estate Tax",

"status": "Waiting"

},

...

]

🆕 2. Add New Queue Entry

POST /enqueue

📥 Request:

json

Copy

Edit

{

"service\_type": "Makatizen GCash Concern",

"counter": 11

}

📤 Response:

json

Copy

Edit

{

"success": true,

"ticket\_code": "GC-0034",

"message": "Ticket added to queue at Counter 11"

}

🔁 3. Update Queue Status

PATCH /status/{counter}

📥 Request:

json

Copy

Edit

{

"code": "RT-0012",

"status": "Now Serving"

}

📤 Response:

json

Copy

Edit

{

"success": true,

"message": "Queue updated for Counter 2"

}

🧹 4. Clear or Close a Ticket

DELETE /clear/{ticket\_code}

📥 Request:

arduino

Copy

Edit

DELETE /clear/GC-0034

📤 Response:

json

Copy

Edit

{

"success": true,

"message": "Ticket GC-0034 has been completed"

}

📊 5. Get Queue Summary by Counter

GET /summary

📥 Request:

pgsql

Copy

Edit

GET /summary

📤 Response:

json

Copy

Edit

{

"total\_active": 11,

"now\_serving": 6,

"waiting": 5,

"counters": [

{"counter": 1, "code": "BP-002", "status": "Now Serving"},

{"counter": 4, "code": "HC-0021", "status": "Waiting"}

]

}

🧾 Service Type Mapping

Each ticket code prefix corresponds to a service:

Prefix Service Counter

BP Business Permit 1

RT Real Estate Tax 2

CT Community Tax 3

HC Health Certificate 4

VP VaxCert PH 5

MC Makatizen Card (General) 6

MR Makatizen Renewal 7

MB Makatizen Biometrics 8

MI Makatizen Inquiry/Follow-ups 9

CR Makatizen Card Releasing 10

GC Makatizen GCash Concern 11

API queue display specification

would you like this delivered as:

✅ OpenAPI / Swagger spec

✅ Postman Collection

✅ Sample frontend (HTML/CSS) integrated with this API?

🗄️ Database Name: makati\_queue\_system

📂 1. services – Service type definitions

sql

Copy

Edit

CREATE TABLE services (

id INT AUTO\_INCREMENT PRIMARY KEY,

service\_code VARCHAR(5) UNIQUE NOT NULL, -- e.g., 'BP', 'RT'

service\_name VARCHAR(100) NOT NULL, -- e.g., 'Business Permit'

default\_counter INT NOT NULL, -- Suggested default counter number

active BOOLEAN DEFAULT TRUE

);

📂 2. queue\_tickets – Queue entries

sql

Copy

Edit

CREATE TABLE queue\_tickets (

id INT AUTO\_INCREMENT PRIMARY KEY,

ticket\_code VARCHAR(10) UNIQUE NOT NULL, -- e.g., 'BP-002'

service\_id INT NOT NULL, -- FK to services

counter INT NOT NULL, -- Assigned counter number

status ENUM('Waiting', 'Now Serving', 'Done') DEFAULT 'Waiting',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

FOREIGN KEY (service\_id) REFERENCES services(id)

);

📂 3. counters – Counter information (optional if dynamic)

sql

Copy

Edit

CREATE TABLE counters (

id INT AUTO\_INCREMENT PRIMARY KEY,

counter\_number INT UNIQUE NOT NULL,

is\_active BOOLEAN DEFAULT TRUE

);

📂 4. queue\_logs – Audit trail (optional)

sql

Copy

Edit

CREATE TABLE queue\_logs (

id INT AUTO\_INCREMENT PRIMARY KEY,

ticket\_code VARCHAR(10) NOT NULL,

counter INT NOT NULL,

action ENUM('Created', 'Updated', 'Called', 'Completed', 'Cancelled') NOT NULL,

timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

remarks TEXT

);

✅ Sample services data:

sql

Copy

Edit

INSERT INTO services (service\_code, service\_name, default\_counter) VALUES

('BP', 'Business Permit', 1),

('RT', 'Real Estate Tax', 2),

('CT', 'Community Tax', 3),

('HC', 'Health Certificate', 4),

('VP', 'VaxCert PH', 5),

('MC', 'Makatizen Card', 6),

('MR', 'Makatizen Renewal', 7),

('MB', 'Makatizen Biometrics', 8),

('MI', 'Makatizen Inquiry / Follow-ups', 9),

('CR', 'Makatizen Card Releasing', 10),

('GC', 'Makatizen GCash Concern', 11);

Would you like me to:

Generate sample data (INSERT INTO queue\_tickets…)?

Provide PHP/Laravel or Node.js backend code to connect to this?

Create admin panel or REST API to control this queue?

🎯 Complete Display Monitor & Backend Integration

I've created a full integration between the display monitor and backend. Here's how everything works together:

📺 Display Monitor Features:

Real-time Updates

WebSocket connection for instant updates

Shows ticket being called with green glow animation

Updates counter status (Open/Closed/Break)

Connection status indicator

Auto-refresh Data

Counter status every 30 seconds

Statistics every minute

Automatic reconnection if disconnected

Visual Feedback

Active counters highlighted

"Now Serving" animation

Toast notifications for new calls

Statistics bar at bottom

🔄 How It Works:

1. Citizen at Kiosk → Creates ticket → Backend generates number

↓

2. Backend broadcasts → WebSocket → Display shows new ticket

↓

3. Staff calls next → Backend updates → Display highlights counter

↓

4. Service complete → Backend logs → Display updates stats

🛠️ Setup Steps:

Start Backend Server

bashcd backend

npm install

npm start

Deploy Display Monitor

Open the integrated display HTML in browser

Set to fullscreen (F11)

Configure backend URL in the code

Test Integration

Create ticket from kiosk

Watch display update in real-time

Call ticket from staff interface

📊 Admin Dashboard Bonus

I've also included an admin dashboard for supervisors to:

Monitor all counters

View real-time statistics

Control counter status

Export reports

🔧 Key Integration Points:

API Endpoints Used by Display:

GET /api/display - Initial counter data

GET /api/analytics - Statistics

WebSocket Events:

ticketCalled - When ticket is called

ticketCompleted - Service finished

counterStatusUpdate - Counter opens/closes

Automatic Features:

Reconnects if connection lost

Caches last state

Graceful error handling