

BONRIX

10.1" Android POS Display HTTP API Reference — Programmer's Guide

Protocol: HTTP POST | Format: JSON | Default Port: 7000 | Auth: None
(LAN)

01 — OVERVIEW

Connection Overview

The Bonrix 10.1" Android Display runs a lightweight HTTP server on your local Wi-Fi network. Your POS software sends a POST request with a JSON body to the display's IP and port. The display immediately switches to the requested screen and renders the provided data.

DEVICE IP	HTTP PORT	ENDPOINT PATH	CONTENT-TYPE
Assigned by router (e.g. 192.168.1.50)	7000 (default)	/api/show	application/json

❏ **IMPORTANT:** The display and your POS system must be on the same Wi-Fi network. Assign a static IP to the display in your router's DHCP settings to prevent the IP from changing on reboot.

02 — ENDPOINT

API Endpoint

All API calls are sent to a single unified endpoint. The screen field in the JSON body determines which screen is displayed.

Method	URL	Content-Type	Description
POST	http://<DEVICE_IP>:<PORT>/api/show	application/json	Send any screen command to the display

Base JSON Structure

Every request uses this envelope. The screen field selects the screen; the data object carries screen-specific fields.

```
{
  "screen": "<ScreenName>", // Which screen to show
  "data": {                 // Screen-specific payload
    ...
  }
}
```

}

All Available Screens

Quick reference of all 13 screen values and their purpose:

#	screen value	Purpose	Use Case
1	<code>WelcomeScreen</code>	Idle / welcome display	Show when no transaction is active
2	<code>DisplayTotalScreen</code>	Cart running total	Update after each item is scanned
3	<code>SingleItemInfo</code>	Last scanned item	Show item name, price, qty when scanned
4	<code>SubmitBill</code>	Full itemised bill	Show complete receipt before payment
5	<code>DisplayQRCodeScreen</code>	UPI QR payment	Show QR code for customer to scan & pay
6	<code>DisplaySuccessQRCodeScreen</code>	Payment success	Show green success after payment confirmed
7	<code>DisplayFailQRCodeScreen</code>	Payment failed	Show failure message, prompt retry
8	<code>DisplayPendingQRCodeScreen</code>	Payment pending	Show spinner while awaiting payment callback
9	<code>DisplayCancelQRCodeScreen</code>	Payment cancelled	Show when customer cancels QR payment
10	<code>PrintScreen</code>	Print receipt trigger	Trigger receipt print with order summary
11	<code>resetWifi</code>	Reset Wi-Fi settings	Send device back to Wi-Fi setup screen
12	<code>CustomScreen</code>	Custom display	Developer-defined custom content
13	<code>Diagnostics</code>	Device diagnostics	View device health and connectivity info

03 — SCREEN API

Screen: WelcomeScreen

	<p>Welcome / Idle Screen</p> <p>Display when no transaction is active. Shows company branding.</p>	<code>WelcomeScreen</code>
---	---	----------------------------

Fields

Field	Type	Required	Description
<code>title</code>	<code>string</code>	Optional	Greeting text (e.g. Welcome!)
<code>CompanyNam</code>	<code>string</code>	Required	Company/brand name
<code>ShopName</code>	<code>string</code>	Required	Branch or shop name
<code>img</code>	<code>string</code>	Optional	Base64 encoded logo image or URL

JSON BODY

```
{
  "screen": "WelcomeScreen",
  "data": {
    "title": "Welcome!",
    "CompanyNam": "Fast Food Corner",
    "ShopName": "Main Branch",
    "img": ""
  }
}
```

CURL COMMAND

```
curl -X POST \
  http://192.168.1.50:7000/api/show \
  -H "Content-Type: application/json" \
  -d '{
    "screen": "WelcomeScreen",
    "data": {
      "title": "Welcome!",
      "CompanyNam": "Fast Food Corner",
      "ShopName": "Main Branch",
      "img": ""
    }
  }'
```

Screen: DisplayTotalScreen

**Display Total**

Shows running cart totals.
Call after every item scan.

DisplayTotalScreen

Fields

Field	Type	Required	Description
TotalAmount	number	Required	Subtotal before discount/tax
DiscountAmount	number	Required	Total discount applied
TaxAmount	number	Required	Total tax (GST/VAT)
GrandTotal	number	Required	Final payable amount
img	string	Optional	Logo image (base64 or URL)

JSON BODY

```
{
  "screen": "DisplayTotalScreen",
  "data": {
    "TotalAmount": 500,
    "DiscountAmount": 50,
    "TaxAmount": 22.5,
    "GrandTotal": 472.5,
    "img": ""
  }
}
```

CURL COMMAND

```
curl -X POST \
  http://192.168.1.50:7000/api/show \
  -H "Content-Type: application/json" \
  -d '{
    "screen": "DisplayTotalScreen",
    "data": {
      "TotalAmount": 500,
      "DiscountAmount": 50,
      "TaxAmount": 22.5,
      "GrandTotal": 472.5,
      "img": ""
    }
  }'
```

Screen: SingleItemInfo

**Single Item Info**

Displays the last scanned
item with price and running
cart total.

SingleItemInfo

Fields

Field	Type	Required	Description
ItemCode	number	Required	Internal item/SKU code
ItemName	string	Required	Product name
Price	number	Required	Unit price
Unit	string	Required	Unit of measurement (Pieces, Kg, etc.)
Qty	number	Required	Quantity of this item added so far
SubTotal	number	Required	Running cart subtotal
Total	number	Required	Price x Qty for this item
GrandTotal	number	Required	Cart grand total
img	string	Optional	Item image (base64 or URL)

JSON BODY

```
{
  "screen": "SingleItemInfo",
  "data": {
    "ItemCode": 2,
    "ItemName": "Hot And Sour Soup",
    "Price": 80,
    "Unit": "Pieces",
    "Qty": 2,
    "SubTotal": 160,
    "Total": 160,
    "GrandTotal": 160,
    "img": ""
  }
}
```

CURL COMMAND

```
curl -X POST \
  http://192.168.1.50:7000/api/show \
  -H "Content-Type: application/json" \
  -d '{
    "screen": "SingleItemInfo",
    "data": {
      "ItemCode": 2,
      "ItemName": "Hot And Sour Soup",
      "Price": 80,
      "Unit": "Pieces",
      "Qty": 2,
      "SubTotal": 160,
      "Total": 160,
      "GrandTotal": 160,
      "img": ""
    }
  }'
```

Screen: SubmitBill

	Submit Bill Full itemised receipt with store info, all line items, and payment summary.	SubmitBill
---	---	-------------------

Nested Object Structure

The SubmitBill payload uses nested objects. Below is the complete field reference:

Object	Field	Type	Required	Description
storeInfo	storeName	string	Required	Shop name
storeInfo	address	string	Optional	Shop address
storeInfo	phone	string	Optional	Contact number
storeInfo	gstIn	string	Optional	GST number

Object	Field	Type	Required	Description
billInfo	billNo	string	Required	Invoice/order number
billInfo	date	string	Required	Date (YYYY-MM-DD)
billInfo	time	string	Required	Time (HH:MM:SS)
billInfo	cashier	string	Required	Cashier name or ID
customerInfo	customerName	string	Optional	Customer name
customerInfo	customerPhone	string	Optional	Customer phone
items[]	itemName	string	Required	Product name
items[]	quantity	number	Required	Quantity ordered
items[]	pricePerUnit	number	Required	Unit selling price
items[]	total	number	Required	quantity x pricePerUnit
totals	subTotal	number	Required	Sum of all item totals
totals	tax	number	Required	Tax amount
totals	discount	number	Required	Discount amount
totals	totalAmount	number	Required	Final payable amount
totals	amountPaid	number	Required	Amount received from customer
totals	paymentMethod	string	Required	Cash / UPI / Card
footer	message	string	Optional	Thank you / footer message
footer	website	string	Optional	Shop website URL

JSON BODY

```

{
  "screen": "SubmitBill",
  "data": {
    "storeInfo": {
      "storeName": "Fast Food Corner",
      "address": "Shop 12, City Mall, Surat",
      "phone": "9876543210",
      "gstIn": "24AAAAA0000A1Z5"
    },
    "billInfo": {
      "billNo": "INV-0061",
      "date": "2025-11-17",
      "time": "16:29:41",
      "cashier": "Ravi"
    },
    "customerInfo": {
      "customerName": "Amit Shah",
      "customerPhone": "9000000001"
    },
    "items": [
      { "itemName": "Hot And Sour Soup", "quantity": 2, "pricePerUnit": 80, "total":
160 },
      { "itemName": "Veg Fried Rice", "quantity": 1, "pricePerUnit": 120, "total":
120 }
    ],
    "totals": {
      "subTotal": 280,
      "tax": 14,
      "discount": 0,
      "totalAmount": 294,
      "amountPaid": 294,
      "paymentMethod": "UPI"
    },
    "footer": {
      "message": "Thank you! Visit again.",

```

```

    "website": "www.fastfoodcorner.in"
  }
}
}

```

CURL COMMAND

```

curl -X POST http://192.168.1.50:7000/api/show \
-H "Content-Type: application/json" \
-d @bill_payload.json

```

```

# Tip: Save the JSON above to bill_payload.json and use -d @filename
# for long payloads to avoid shell escaping issues.

```

Screen: DisplayQRCodeScreen

**Display QR Code**

Shows UPI QR code for customer to scan and pay.

DisplayQRCodeScreen

Fields

Field	Type	Required	Description
UPIUrl	string	Required	UPI deep link string — device auto-generates QR from this
QrCodeBase64	string	Optional	Pre-generated QR as Base64 image; leave empty to auto-generate
Amount	number	Required	Payment amount in INR
CompanyNam	string	Required	Company name shown on screen
ShopName	string	Required	Shop/branch name
title	string	Optional	Screen heading text
img	string	Optional	Logo image

i UPI URL Format:

upi://pay?pa=<VPA>&pn=<Name>&am=<Amount>&cu=INR&tn=<TxnNote>&tr=<TxnRef> | Replace pa with your UPI VPA (e.g. merchant@upi), pn with payee name, am with amount.

JSON BODY

```

{
  "screen": "DisplayQRCodeScreen",
  "data": {
    "QrCodeBase64": "",
    "UPIUrl":
"upi://pay?pa=shop@upi&pn=FastFood&am=294.00&cu=INR&tn=INV-0061&tr=INV-0061",
    "Amount": 294,
    "CompanyNam": "Fast Food Corner",
    "ShopName": "Main Branch",
    "title": "Scan to Pay",
    "img": ""
  }
}

```

CURL COMMAND

```

curl -X POST \
  http://192.168.1.50:7000/api/show \
  -H "Content-Type: application/json" \
  -d '{
    "screen": "DisplayQRCodeScreen",
    "data": {
      "QrCodeBase64": "",
      "UPIUrl":
"upi://pay?pa=shop@upi&pn=FastFood&am=294.00&cu=INR&tn=INV-0061&tr=INV-0061",
      "Amount": 294,
      "CompanyNam": "Fast Food Corner",
      "ShopName": "Main Branch",
      "title": "Scan to Pay"
    }
  }'

```

}

}'

QR PAYMENT STATUS SCREENS

QR Payment Status Screens

After polling your payment gateway, send the corresponding status screen. All four screens share the same payload structure.

□ Typical QR Payment Flow: DisplayQRCodeScreen → (poll gateway) → DisplayPendingQRCodeScreen → DisplaySuccessQRCodeScreen or DisplayFailQRCodeScreen or DisplayCancelQRCodeScreen

Common Fields (All QR Status Screens)

Field	Type	Required	Description
OrderId	string	Required	Your order/bill number
BankRef	string	Optional	Bank/UPI transaction reference (available after payment)
Amount	number	Required	Payment amount in INR
Date	number	Required	Unix timestamp (seconds) of transaction
CompanyNam	string	Required	Company name
ShopName	string	Required	Shop name
title	string	Optional	Screen title text
img	string	Optional	Logo image



Payment Success

Same payload structure as other QR status screens

DisplaySuccessQRCodeScreen

JSON BODY

```
{
  "screen": "DisplaySuccessQRCodeScreen",
  "data": {
    "OrderId": "INV-0061",
    "BankRef": "UPI123456789",
    "Amount": 294,
    "Date": 1732000000,
    "CompanyNam": "Fast Food Corner",
    "ShopName": "Main Branch",
    "title": "Payment Successful",
    "img": ""
  }
}
```

CURL COMMAND

```
curl -X POST \
  http://192.168.1.50:7000/api/show \
  -H "Content-Type: application/json" \
  -d '{
    "screen": "DisplaySuccessQRCodeScreen",
    "data": {
      "OrderId": "INV-0061",
      "BankRef": "UPI123456789",
      "Amount": 294,
      "Date": 1732000000,
      "CompanyNam": "Fast Food Corner",
      "ShopName": "Main Branch"
    }
  }'
```

**Payment Failed**

Same payload structure as other QR status screens

DisplayFailQRCodeScreen**JSON BODY**

```
{
  "screen": "DisplayFailQRCodeScreen",
  "data": {
    "OrderId": "INV-0061",
    "BankRef": "UPI123456789",
    "Amount": 294,
    "Date": 1732000000,
    "CompanyNam": "Fast Food Corner",
    "ShopName": "Main Branch",
    "title": "Payment Failed",
    "img": ""
  }
}
```

CURL COMMAND

```
curl -X POST \
  http://192.168.1.50:7000/api/show \
  -H "Content-Type: application/json" \
  -d '{
    "screen": "DisplayFailQRCodeScreen",
    "data": {
      "OrderId": "INV-0061",
      "BankRef": "UPI123456789",
      "Amount": 294,
      "Date": 1732000000,
      "CompanyNam": "Fast Food Corner",
      "ShopName": "Main Branch"
    }
  }'
```

**Payment Pending**

Same payload structure as other QR status screens

DisplayPendingQRCodeScreen**JSON BODY**

```
{
  "screen": "DisplayPendingQRCodeScreen",
  "data": {
    "OrderId": "INV-0061",
    "BankRef": "UPI123456789",
    "Amount": 294,
    "Date": 1732000000,
    "CompanyNam": "Fast Food Corner",
    "ShopName": "Main Branch",
    "title": "Waiting for Payment",
    "img": ""
  }
}
```

CURL COMMAND

```
curl -X POST \
  http://192.168.1.50:7000/api/show \
  -H "Content-Type: application/json" \
  -d '{
    "screen": "DisplayPendingQRCodeScreen",
    "data": {
      "OrderId": "INV-0061",
      "BankRef": "UPI123456789",
      "Amount": 294,
      "Date": 1732000000,
      "CompanyNam": "Fast Food Corner",
      "ShopName": "Main Branch"
    }
  }'
```

**Payment Cancelled**

Same payload structure as other QR status screens

DisplayCancelQRCodeScreen**JSON BODY**

```
{
  "screen": "DisplayCancelQRCodeScreen",
  "data": {
    "OrderId": "INV-0061",
    "BankRef": "UPI123456789",
    "Amount": 294,
    "Date": 1732000000,
    "CompanyNam": "Fast Food Corner",
    "ShopName": "Main Branch",
    "title": "Payment Cancelled",
    "img": ""
  }
}
```

CURL COMMAND

```
curl -X POST \
  http://192.168.1.50:7000/api/show \
  -H "Content-Type: application/json" \
  -d '{
    "screen": "DisplayCancelQRCodeScreen",
    "data": {
      "OrderId": "INV-0061",
      "BankRef": "UPI123456789",
      "Amount": 294,
      "Date": 1732000000,
      "CompanyNam": "Fast Food Corner",
      "ShopName": "Main Branch"
    }
  }'
```

Screen: PrintScreen

	<p>Print Receipt</p> <p>Triggers a receipt print with order and payment summary.</p>	PrintScreen
---	---	--------------------

Fields

Field	Type	Required	Description
OrderId	string	Required	Order / invoice number
BankRef	string	Optional	UPI/Bank transaction reference
Amount	number	Required	Total amount paid
Date	number	Required	Unix timestamp of transaction
CompanyNam	string	Required	Company name
ShopName	string	Required	Shop/branch name

JSON BODY

```
{
  "screen": "PrintScreen",
  "data": {
    "OrderId": "INV-0061",
    "BankRef": "UPI123456789",
    "Amount": 294,
    "Date": 1732000000,
    "CompanyNam": "Fast Food Corner",
    "ShopName": "Main Branch"
  }
}
```

CURL COMMAND

```
curl -X POST \
  http://192.168.1.50:7000/api/show \
  -H "Content-Type: application/json" \
  -d '{
    "screen": "PrintScreen",
    "data": {
      "OrderId": "INV-0061",
      "BankRef": "UPI123456789",
      "Amount": 294,
      "Date": 1732000000,
      "CompanyNam": "Fast Food Corner",
      "ShopName": "Main Branch"
    }
  }'
```

SYSTEM / UTILITY SCREENS

System Screens

resetWifi

	<p>Reset Wi-Fi</p> <p>Reopens Wi-Fi setup wizard on the device. Requires correct PIN.</p>	resetWifi
---	--	------------------

WARNING: The device will disconnect from the current Wi-Fi network after this call. Ensure you have physical access to reconfigure it.

JSON BODY

```
{
  "screen": "resetWifi",
  "data": {
    "pin": "1234"
  }
}
```

CURL COMMAND

```
curl -X POST \
  http://192.168.1.50:7000/api/show \
  -H "Content-Type: application/json" \
  -d '{
    "screen": "resetWifi",
    "data": { "pin": "1234" }
  }'
```

CustomScreen

**Custom Screen**

Developer-defined content. Structure of data depends on your custom screen implementation.

CustomScreen**JSON BODY**

```
{
  "screen": "CustomScreen",
  "data": {
    // Your custom fields here
  }
}
```

CURL COMMAND

```
curl -X POST \
  http://192.168.1.50:7000/api/show \
  -H "Content-Type: application/json" \
  -d '{
    "screen": "CustomScreen",
    "data": {}
  }'
```

Diagnostics

**Diagnostics**

Opens device health screen showing network info, app version, and device status.

Diagnostics**JSON BODY**

```
{
  "screen": "Diagnostics",
  "data": {}
}
```

CURL COMMAND

```
curl -X POST \
  http://192.168.1.50:7000/api/show \
  -H "Content-Type: application/json" \
  -d '{"screen":"Diagnostics","data":{}}'
```

INTEGRATION GUIDE

Typical POS Transaction Flow

Use this sequence as a guide when integrating the display into your POS application:

Step	API Screen	When to Call
1	WelcomeScreen	App starts / idle state
2	SingleItemInfo	Each barcode scanned
3	DisplayTotalScreen	After each item added

Step	API Screen	When to Call
4	<code>SubmitBill</code>	Checkout initiated
5a	<code>DisplayQRCodeScreen</code>	Customer chooses UPI payment
5b	<code>DisplayPendingQRCodeScreen</code>	While polling payment gateway
6a	<code>DisplaySuccessQRCodeScreen</code>	Payment confirmed ✓
6b	<code>DisplayFailQRCodeScreen</code>	Payment failed ✗
6c	<code>DisplayCancelQRCodeScreen</code>	Customer cancelled ☐
7	<code>PrintScreen</code>	After payment confirmed
8	<code>WelcomeScreen</code>	Back to idle state

Quick Test — Ping the Device

Use the Diagnostics screen to verify the device is reachable from your network:

```
# Test if device is reachable (replace IP and port with your device values)
curl -X POST http://192.168.1.50:7000/api/show \
  -H "Content-Type: application/json" \
  -d '{"screen":"Diagnostics","data":{}}' \
  -w "\nHTTP Status: %{http_code}\n"

# Expected: HTTP 200 = device online and responsive
# If no response: check IP address, port, and same Wi-Fi network
```

i Reference: This document was generated from https://www.bonrix.co.in/pos_api_test.html — Bonrix 10.1 inch Android POS Display HTTP API