

JunkShop Website - Deployment & Configuration Guide

Last Updated: October 17, 2025

Status: Ready for deployment after SumUp configuration

Table of Contents

1. [Prerequisites](#)
 2. [SumUp Account Setup](#)
 3. [Firebase Configuration](#)
 4. [Deployment Steps](#)
 5. [Post-Deployment Configuration](#)
 6. [Testing](#)
 7. [Troubleshooting](#)
-

Prerequisites

Before deploying, ensure you have:

- [x] Firebase project created (`junkshop-website-gem`)
- [x] Node.js installed (v18 or later)
- [x] Firebase CLI installed (`npm install -g firebase-tools`)
- [x] Git installed
- [] SumUp merchant account (with API credentials)
- [] Admin email/password for Firebase Authentication

Install Firebase CLI

```
npm install -g firebase-tools
```

Login to Firebase

```
firebase login
```

SumUp Account Setup

1. Create SumUp Merchant Account

1. Go to <https://sumup.com/en-gb/>
2. Sign up for a merchant account

3. Complete verification process
4. Note your merchant email: `junkshopdumfries@gmail.com`

2. Get API Credentials

1. Login to SumUp Dashboard: <https://me.sumup.com/>
2. Navigate to **Developer Tools** or **API Settings**
3. Create a new API application
4. Copy your **Secret API Key** (looks like: `sup_sk_...`)
5. **IMPORTANT:** Store this key securely - you'll need it for deployment

3. Configure Webhook (After Deployment)

You'll configure this AFTER deploying the Cloud Functions. The webhook URL will be:

```
https://europe-west2-junkshop-website-gem.cloudfunctions.net/handleSumupWebhook
```

Firestore Configuration

1. Enable Authentication

1. Go to Firebase Console: <https://console.firebase.google.com/>
2. Select your project: **junkshop-website-gem**
3. Go to **Authentication** → **Sign-in method**
4. Enable **Anonymous** authentication (required for customers)
5. Enable **Email/Password** authentication (required for admin)

2. Create Admin User

```
# In Firebase Console → Authentication → Users  
# Click "Add User"  
# Email: your-admin@email.com  
# Password: [Choose a strong password]
```

3. Configure Firestore Security Rules

Go to **Firestore Database** → **Rules** and paste:

```

rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    // Products - Public read, admin write
    match /products/{productId} {
      allow read: if true;
      allow write: if request.auth != null && request.auth.token.email != null;
    }

    // Orders - Authenticated users can create, admins can read/update
    match /orders/{orderId} {
      allow create: if request.auth != null;
      allow read, update: if request.auth != null && request.auth.token.email != null;
      allow delete: if request.auth != null && request.auth.token.email != null;
    }
  }
}

```

4. Configure Storage Rules

Go to **Storage** → **Rules** and paste:

```

rules_version = '2';
service firebase.storage {
  match /b/{bucket}/o {
    // Product images - Admin only
    match /products/{allPaths=**} {
      allow read: if true;
      allow write: if request.auth != null && request.auth.token.email != null;
    }
  }
}

```

Deployment Steps

Step 1: Clone Repository (if not done)

```

cd /home/landsendsolo
git clone https://github.com/landsendsolo/junkshop_live.git
cd junkshop_live

```

Step 2: Install Dependencies

```

cd functions
npm install
cd ..

```

Step 3: Configure SumUp Secret Key

CRITICAL: Replace `YOUR_SUMUP_SECRET_KEY` with your actual SumUp API key:

```

firebase functions:config:set sumup.secret_key="YOUR_SUMUP_SECRET_KEY"

```

Verify configuration:

```
firebase functions:config:get
```

You should see:

```
{
  "sumup": {
    "secret_key": "sup_sk_..."
  }
}
```

Step 4: Deploy Cloud Functions

```
firebase deploy --only functions
```

Expected output:

```
✓ Deploy complete!

Functions:
  createSumupCheckout(europe-west2)
  handleSumupWebhook(europe-west2)
```

Copy the webhook URL from the output - you'll need it for SumUp configuration.

Step 5: Deploy Website

```
firebase deploy --only hosting
```

Expected output:

```
✓ Deploy complete!

Hosting URL: https://junkshop-website-gem.web.app
```

Post-Deployment Configuration

1. Configure SumUp Webhook

1. Login to SumUp Dashboard: <https://me.sumup.com/>
2. Go to **Developer Tools** → **Webhooks**
3. Add new webhook:
 - **URL**: `https://europe-west2-junkshop-website-gem.cloudfunctions.net/handleSumupWebhook`
 - **Events**: Select `transaction.success` or `checkout.paid`
4. Save webhook configuration

2. Test Admin Access

1. Go to: `https://junkshop-website-gem.web.app/admin.html`
2. Login with your admin credentials
3. Verify you can:
 - View/add/edit products
 - View orders
 - Upload images

3. Add Initial Products

1. In admin dashboard, click **Add New Product**
 2. Fill in product details
 3. Upload images
 4. Mark some products as “Featured” and “Hero”
 5. Save product
-

Testing

Test Checklist

Frontend Testing

- ☐ Homepage loads correctly
- ☐ Products display properly
- ☐ Cart functionality works
- ☐ Theme switcher works
- ☐ Mobile responsive design

Payment Flow Testing

- ☐ Can add items to cart
- ☐ Checkout form validates correctly
- ☐ Delivery vs Collection options work
- ☐ Payment button triggers SumUp

SumUp Integration Testing

IMPORTANT: Use SumUp test credentials for testing

1. Test Mode Setup:

- Request test API credentials from SumUp support
- Configure test webhook URL
- Use test credit card numbers

2. Test Purchase Flow:

Test Card: 4242 4242 4242 4242

Expiry: Any future date

CVV: Any 3 digits

3. Verify:

- ☐ Order created with status “pending_payment”
- ☐ Redirects to SumUp payment page

- [] After payment, webhook triggers
- [] Order status updates to “paid”
- [] Products marked as “sold”

Admin Dashboard Testing

- [] Can view all orders
- [] Order statuses display correctly
- [] Can update order status
- [] Can manage products

Production Testing

Before going live:

1. Complete at least 3 test purchases
2. Verify webhook receives payment confirmations
3. Check order confirmation emails (once implemented)
4. Test failed payment scenarios
5. Verify refund process

Troubleshooting

Issue: “Payment system not configured” error

Cause: SumUp secret key not set

Solution:

```
firebase functions:config:set sumup.secret_key="YOUR_KEY"
firebase deploy --only functions
```

Issue: “Anonymous sign-in failed” error

Cause: Anonymous authentication not enabled

Solution:

1. Firebase Console → Authentication → Sign-in method
2. Enable “Anonymous” provider
3. Refresh website

Issue: Webhook not receiving notifications

Causes & Solutions:

1. Webhook URL incorrect:

- Verify URL in SumUp dashboard matches Cloud Function URL

2. Function not deployed:

```
bash
firebase deploy --only functions:handleSumupWebhook
```

3. Check logs:

```
bash
firebase functions:log --only handleSumupWebhook
```

Issue: Orders not updating after payment

Check:

1. View Firebase Functions logs for webhook errors
2. Verify Firestore security rules allow updates
3. Check SumUp webhook is configured correctly

View logs:

```
firebase functions:log
```

Issue: Images not uploading

Cause: Storage security rules too restrictive

Solution: Update Storage rules to allow authenticated writes

Issue: “CORS error” on checkout

Cause: Firebase Functions CORS not configured

Solution: Cloud Functions should handle CORS automatically, but if issues persist:

```
// Add to functions/index.js
const cors = require('cors')({origin: true});

exports.createSumupCheckout = functions.https.onCall(async (data, context) => {
  // Your existing code
});
```

Monitoring

View Logs

```
# All functions
firebase functions:log

# Specific function
firebase functions:log --only createSumupCheckout

# Real-time logs
firebase functions:log --only handleSumupWebhook --follow
```

Check Function Status

```
firebase functions:list
```

Monitor Errors

1. Firebase Console → Functions → Logs

2. Look for errors in:

- `createSumupCheckout` - Payment creation errors
- `handleSumupWebhook` - Payment confirmation errors

Security Considerations

Production Checklist

- ☐ Change all default passwords
- ☐ Enable 2FA on Firebase account
- ☐ Review Firestore security rules
- ☐ Enable Firebase App Check (recommended)
- ☐ Set up monitoring/alerts for unusual activity
- ☐ Regular security audits
- ☐ Backup Firestore data regularly
- ☐ Keep dependencies updated

Data Protection

1. **Customer Data:**

- Stored in Firestore (encrypted at rest)
- Covered by Firebase security rules
- GDPR compliant

2. **Payment Data:**

- NOT stored locally
- Handled entirely by SumUp (PCI DSS compliant)
- Only transaction IDs stored

Backup & Recovery

Backup Firestore

```
# Install firestore-backup-restore
npm install -g firestore-backup-restore

# Backup
firestore-export --backupFile backup.json --accountCredentials serviceAccountKey.json







# Restore
firestore-import --backupFile backup.json --accountCredentials serviceAccountKey.json
```

Export Configuration

```
# Export Firebase config
firebase functions:config:get > firebase-config.json

# Backup security rules
firebase firestore:rules > firestore.rules.backup
```

Next Steps

1.  Complete deployment following this guide
2.  Test payment flow with SumUp test credentials
3.  Implement email notifications (see recommendations below)
4.  Set up Google Analytics
5.  Add customer account system
6.  Implement order tracking

Recommended Enhancements

1. **Email Notifications:**
 - Install Firebase Extension: “Trigger Email”
 - Or use SendGrid/Mailgun
 - Send order confirmations
 - Send shipping notifications
2. **Analytics:**
 - Add Google Analytics 4
 - Track conversions
 - Monitor cart abandonment
3. **Performance:**
 - Add Firebase Performance Monitoring
 - Optimize images (use CDN)
 - Implement lazy loading

Support Resources

- **Firebase Documentation:** <https://firebase.google.com/docs>
 - **SumUp API Docs:** <https://developer.sumup.com/docs>
 - **Firebase Community:** <https://firebase.google.com/support>
 - **SumUp Support:** <https://help.sumup.com/>
-

Quick Reference

Essential Commands

```
# Deploy everything
firebase deploy

# Deploy functions only
firebase deploy --only functions

# Deploy hosting only
firebase deploy --only hosting

# View logs
firebase functions:log

# Test locally
firebase emulators:start
```

Important URLs

- **Live Site:** <https://junkshop-website-gem.web.app>
- **Admin Panel:** <https://junkshop-website-gem.web.app/admin.html>
- **Firebase Console:** <https://console.firebase.google.com/project/junkshop-website-gem>
- **SumUp Dashboard:** <https://me.sumup.com/>

Deployment Guide Last Updated: October 17, 2025

Guide Version: 1.0

Contact: junkshopdumfries@gmail.com