

AbuByte POS - API Documentation

Complete REST API Integration Guide

PAGE 1: FIREBASE SERVICES & AUTHENTICATION

Overview

AbuByte POS uses **Firestore REST APIs exclusively** (no SDKs) and operates on the **free Spark plan**. All services are pre-configured and ready for immediate use.

Firestore Project Details

- **Project ID:** abubyte-pos-new
- **Status:** Complete project included in sale
- **Plan:** Firestore Spark (Free) - No subscription required
- **Billing:** Free within usage limits; upgrade to Blaze plan if scaling needed

Authentication Flow

1. User Login

dart

// Endpoint: Firestore Auth REST API

POST

https://identitytoolkit.googleapis.com/v1/accounts:signInWithPassword?key=AIzaSyADK63YTC8lyNna0FeM8K151R4_Ilq3-yk

// Request Body

```
{  
  "email": "admin@yourbusiness.com",  
  "password": "password123",  
  "returnSecureToken": true  
}
```

// Response

```
{  
  "idToken": "eyJhbGciOiJSUzI1NiIsImtpZCI6Ij...",  
  "refreshToken": "AEu4IL0m...",  
  "expiresIn": "3600",  
  "localId": "user123"  
}
```

2. Token Refresh

dart

// Endpoint: Secure Token API

POST

https://securetoken.googleapis.com/v1/token?key=AIzaSyADK63YTC8lyNna0FeM8Kl51R4_Ilq3-yk

// Request Body

```
{  
  "grant_type": "refresh_token",  
  "refresh_token": "AEu4IL0m..."  
}
```

3. Password Reset

dart

// Endpoint: Send OOB Code

POST

https://identitytoolkit.googleapis.com/v1/accounts:sendOobCode?key=AIzaSyADK63YTC8lyNna0FeM8Kl51R4_Ilq3-yk

// Request Body

```
{
```

```
"requestType": "PASSWORD_RESET",  
"email": "user@example.com"  
}
```

API Rate Limits & Quotas

Service	Free Tier Limit	Current Usage
Firestore Reads	50,000/day	Cached (5-min expiry)
Firestore Writes	20,000/day	Optimized batches
Firestore Deletes	20,000/day	Minimal usage
Auth Requests	Unlimited*	Token-based
Storage	5GB + 1GB/day	Product images only

PAGE 2: FIRESTORE & STORAGE APIs

Firestore REST API

Base Endpoint

[https://firestore.googleapis.com/v1/projects/abubyte-pos-new/databases/\(default\)/documents/](https://firestore.googleapis.com/v1/projects/abubyte-pos-new/databases/(default)/documents/)

Core Collections & Operations

1. Users Management

dart

// Get User Document

GET /users/{userId}

// Create/Update User

PATCH /users/{userId}

```
{
  "fields": {
    "uid": {"stringValue": "user123"},
    "email": {"stringValue": "admin@business.com"},
    "name": {"stringValue": "Admin User"},
    "role": {"stringValue": "admin"},
    "businessId": {"stringValue": "business123"}
  }
}
```

// Query Users by Business

POST /users:runQuery

```
{
  "structuredQuery": {
    "where": {
      "fieldFilter": {
        "field": {"fieldPath": "businessId"},
        "op": "EQUAL",
        "value": {"stringValue": "business123"}
      }
    }
  }
}
```

2. Products Management

dart

// Business-Specific Products Collection

GET /businesses_{businessId}_products/{productId}

// Add Product

```
POST /businesses_{businessId}_products
{
  "fields": {
    "id": {"stringValue": "prod123"},
    "name": {"stringValue": "Product Name"},
    "price": {"doubleValue": 19.99},
    "stock": {"integerValue": 100},
    "category": {"stringValue": "category123"}
  }
}
```

3. Sales Transactions

dart

// Create Sale

```
POST /businesses_{businessId}_sales
{
  "fields": {
    "id": {"stringValue": "sale123"},
    "total": {"doubleValue": 59.97},
    "items": {"arrayValue": {"values": [...]}},
    "cashier": {"stringValue": "cashier123"},
    "timestamp": {"timestampValue": "2024-01-15T10:30:00Z"}
  }
}
```

Atomic Batch Operations

Transactional Endpoint

dart

// Firestore Native Commit API

POST [https://firestore.googleapis.com/v1/projects/abubyte-pos-new/databases/\(default\)/commit](https://firestore.googleapis.com/v1/projects/abubyte-pos-new/databases/(default)/commit)

// Request Body - Multiple Operations Atomically

```
{
  "writes": [
    {
      "update": {
        "name": "projects/abubyte-pos-new/databases/(default)/documents/sales/sale-123",
        "fields": {"total": {"doubleValue": 59.97}, ...}
      }
    },
    {
      "update": {
        "name": "projects/abubyte-pos-new/databases/(default)/documents/products/prod-456",
        "fields": {"stock": {"integerValue": 95}, ...}
      }
    }
  ]
}
```

Implementation: lib/sync/services/atomic_batch_service.dart

Firestore Storage API

File Upload

dart

// Upload Product Image

POST https://storage.googleapis.com/upload/storage/v1/b/abubyte-pos-new.firebaseio.com/app/o?uploadType=media&name=product-images/image123.jpg

// Headers

Content-Type: image/jpeg

Authorization: Bearer {idToken}

// Response

```
{  
  "name": "product-images/image123.jpg",  
  "bucket": "abubyte-pos-new.firebaseio.com/app",  
  "size": "154996"  
}
```

Generate Download URL

dart

// Get Download URL with Token

GET https://firebasestorage.googleapis.com/v0/b/abubyte-pos-new.firebaseio.com/app/o/product-images%2Fimage123.jpg?alt=media&token={downloadToken}

Menu PDF Storage

dart


// Upload Menu PDF

POST /upload/storage/v1/b/abubyte-pos-new.firebaseio.com/app/o?uploadType=media&name=menus/business123/latest_menu.pdf

PAGE 3: EXTERNAL APIs & CONFIGURATION

Pixabay Image Search API

API Configuration

- **Service:** Pixabay Image Search
- **Status:**  Active & Integrated
- **Key Security:** Environment variable protected
- **Rate Limit:** 5,000 requests/hour (Free)
- **Location:** lib/core/config/api_keys.dart
- **Build Command:** flutter build windows --release --dart-define=PIXABAY_API_KEY=your_key_here

Search Endpoint

dart

// Image Search

GET

https://pixabay.com/api/?key={PIXABAY_API_KEY}&q=product+name&image_type=photo&safesearch=true&per_page=20

// Response

```
{
  "hits": [
    {
      "id": 12345,
      "webformatURL": "https://pixabay.com/get/...",
      "largeImageURL": "https://pixabay.com/get/...",
      "tags": "product, item, object"
    }
  ],
  "total": 125,
  "totalHits": 125
}
```

Implementation Flow

1. User searches for images in product/menu dialogs
2. App calls Pixabay API with search query
3. Results displayed as thumbnail grid
4. Selected image downloaded and converted to base64
5. Image saved locally and uploaded to Firebase Storage

Service: lib/features/menu/services/online_image_search_service.dart

UI: lib/widgets/online_image_search_dialog.dart

API Keys Configuration

File Structure

lib/core/config/

|— api_keys.dart # External service keys (Pixabay)

|— firebase_config.dart # Firebase configuration

Current Keys

dart

// api_keys.dart - Secure environment variable approach

```
class ApiKeys {  
  static const String pixabayApiKey = String.fromEnvironment(  
    'PIXABAY_API_KEY',  
    defaultValue: '49995190-80f81fa57ec40ceb950911989', // Development fallback  
  );  
}
```

// firebase_config.dart

```
class FirebaseConfig {  
  static const String apiKey = 'AIzaSyADK63YTC8lyNna0FeM8Kl51R4_Ilq3-yk';  
  static const String projectId = 'abubyte-pos-new';  
  static const String storageBucket = 'abubyte-pos-new.firebaseiostorage.app';  
}
```

}

Connectivity Monitoring

Flutter Package

- **Package:** connectivity_plus: ^7.0.0
- **Purpose:** Monitor online/offline status
- **Usage:** Triggers sync operations when connectivity restored

Production Build Instructions

Secure Build with Environment Variables

bash

Production build with secure API keys

```
flutter build windows --release --dart-define=PIXABAY_API_KEY=your_actual_production_key
```

Development build (uses fallback key)

```
flutter build windows --release
```

Handover Instructions

For Buyer:

- ✓ Complete Firebase project included (abubyte-pos-new)
- ✓ All REST APIs pre-configured and working
- ✓ No code changes required for Firebase connectivity
- ✓ Free Spark plan sufficient for small-medium businesses
- ✓ Upgrade to Blaze plan only if scaling beyond free limits
- ✓ Professional API key security with environment variables






Firebase Project Transfer:

1. **We transfer** complete Firebase project ownership
2. **Buyer updates** billing information if scaling needed
3. **No reconfiguration** required - everything works immediately

4. **All API keys** remain valid and functional

Security Implementation

Current Security:

-  Firebase API keys are public and safe to expose
-  Pixabay key uses environment variables (secure)
-  All authentication uses secure token-based approach
-  Firestore security rules enforce data isolation
-  Professional key management practices

Cost Management

Current Costs:





- **Firebase Services:** \$0.00 (Free Spark plan)
- **Pixabay API:** \$0.00 (Free tier)
- **Total Monthly Cost:** \$0.00

Scaling Considerations:

- **Firestore:** Upgrade to Blaze plan if exceeding 50K reads/day
- **Storage:** Monitor 5GB free limit for product images
- **Pixabay:** 5,000 requests/hour sufficient for normal usage

Support & Maintenance

Included Support:

-  30-day transition assistance
-  Firebase project handover guidance
-  API integration troubleshooting
-  Complete code documentation

Monitoring Recommendations:

1. Set up Firebase usage alerts
2. Monitor Pixabay API rate limits

3. Track storage usage growth
4. Review Firestore query performance

API Documentation - *Enterprise-Ready Secure Integration* 