

# API Integration Checklist™ for Digital Signage

**For:** Enterprise IT, operations, and procurement teams evaluating integration readiness before deployment.

**Purpose:** Use this checklist to assess whether your signage environment is ready for CRM, POS, ERP, and IoT integrations.

## 1. Business Objective Alignment

- Define the primary business outcome: revenue lift, operational visibility, safety response, personalization, or labor efficiency.
- Identify which source systems are required: CRM, POS, ERP, IoT, HRIS, BI, or external feeds.
- Confirm whether the use case is awareness-based or event-driven.

## 2. Source System Readiness

- Verify that each source system exposes usable APIs, webhooks, or middleware connectors.
- Document required data fields such as SKU, inventory, loyalty segment, production KPI, room status, or sensor state.
- Identify system owners and confirm access approval for each integration.

## 3. Integration Architecture

- Map the full flow: Data Source → API → Middleware → Cache → CMS → Screen.
- Decide whether the model is direct API, middleware/iPaaS, or event-stream based.
- Confirm fallback behavior for stale data, null values, and downtime.
- Confirm a caching layer is included to stabilize refresh behavior and reduce API load.

## 4. Content Trigger Logic

- Define clear if/then rules, such as If inventory < threshold → show scarcity overlay.
- Verify the CMS supports rules, variables, and dynamic rendering.
- Ensure fallback content exists for normal, warning, and outage states.

## 5. Security and API Governance

- Use OAuth 2.0, scoped API keys, or equivalent enterprise authentication.
- Apply least-privilege access so each integration only reaches required data and screens.
- Set rate limits, retry policies, and throttling controls.
- Enable schema validation to block malformed payloads.
- Turn on logging and monitoring for failed calls, auth events, and trigger errors.

## 6. Reliability and Failure Modes

- Test stale-screen scenarios caused by API downtime.
- Test data mismatch scenarios caused by schema or field changes.
- Test latency spikes under load and compare webhook versus polling behavior.
- Test CMS, player, and network disconnect behavior.

## 7. Measurement and ROI

- Define success metrics such as conversion rate, upsell rate, throughput, incident reduction, or response time.
- Track data-to-display latency as a core performance metric.
- Establish a baseline before rollout so lift can be measured after deployment.
- Map integration triggers to business outcomes using an attribution model.

## 8. Launch Readiness

- Assign technical owner.
- Assign business owner.
- Complete security review.
- Approve latency thresholds by use case.
- Test trigger logic and fallback logic.
- Enable monitoring and alerting.
- Approve pilot location or rollout scope.

## Latency Threshold Reference

Use Case	Target Latency
IoT safety alert	< 1 second
POS upsell / scarcity trigger	< 3 seconds
ERP production KPI	< 30 seconds
CRM leaderboard / recognition	< 5 minutes
Weather / social / contextual feed	< 15 minutes

## Procurement Review Table

Evaluation Criteria	What to Verify	Red Flag
System compatibility	REST / GraphQL / MQTT / webhook support	Proprietary-only API
Security compliance	Slow reporting	Negotiate SLAs
Latency commitment	Campaign failures	Enforce specs
Vendor support	Blank screens	Configure safe defaults
Scalability	Reporting crashes	Load test pre-launch

# Scoring Guide

How to interpret your score:

- **0–15 checks complete:** Early-stage readiness. Start with architecture mapping and source system validation.
- **16–30 checks complete:** Moderate readiness. Proceed with pilot design and latency testing.
- **31+ checks complete:** High readiness. Move toward production deployment with monitoring, governance, and scale planning.

© 2026 [BlinkSigns](#)

This checklist is a general framework. Requirements vary based on your existing tech stack, vendor ecosystem, and scale.