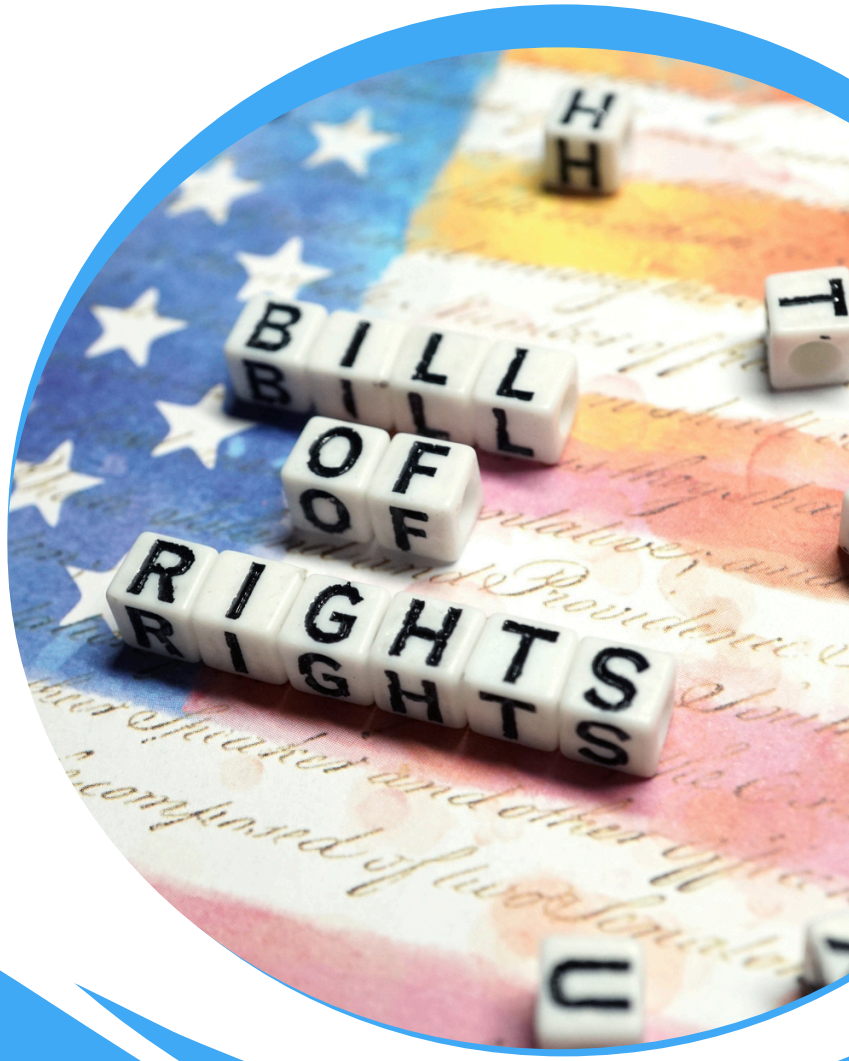
LEAGUE OF ASSOCIATION TECHNOLOGISTS

INTEGRATION BILL OF RIGHTS

FOR ASSOCIATION SOFTWARE VENDORS

2025

DRAFT



Help Shape the Future of Association Technology Integration

Sponsored by the League of Association Technologists

As association technology continues to evolve, so do the expectations around seamless data flow and system connectivity. Too often, associations find themselves trapped by inadequate APIs, poor documentation, or integration promises that fall short of reality.

We're developing an **Integration Bill of Rights** - a comprehensive framework that association software vendors can use to benchmark their integration capabilities and that associations can reference when evaluating potential solutions. This isn't about creating barriers; it's about establishing clear guidelines that benefit everyone in our ecosystem.

Your expertise matters. Whether you're a vendor who has wrestled with integration challenges or an association executive who has experienced the pain of poor connectivity, your insights will help refine these guidelines into something truly valuable for our industry.

Please take a few minutes to review the framework below and share your feedback. What resonates? What's missing? What would you change?

Join the conversation at one of our virtual town halls or submit your comments online. Town Halls will be held via Zoom on:

- **July 15**
- **August 20**
- **September 17**

Together, we can create standards that drive better outcomes for associations and their technology partners alike.

Integration Bill of Rights for Association Software Vendors

1. Standards Based

Integration should utilize industry-standard protocols and data formats, specifically REST/JSON or SOAP/XML. Vendors should implement these standards consistently across all APIs, avoiding proprietary formats that lock customers into specific technologies. Data structures should follow established patterns, with consistent field naming conventions and data types that align with industry best practices.

2. Documented

Complete API documentation should be freely available to both existing customers and prospective buyers. This documentation must include:

- Comprehensive endpoint descriptions
- Request and response examples
- Authentication requirements
- Error handling procedures
- Detailed change logs and release notes for all versions
- Deprecation schedules for outdated endpoints
- Sample code in multiple programming languages

3. Secure & Accessible

Security should be paramount without sacrificing accessibility:

- All data transmission must be encrypted using industry-standard protocols (TLS 1.2+)
- Connection-level security options like IP filtering must be available
- Standard authentication methods such as OAuth 2.0 should be implemented
- Role-based access controls should be configurable to limit API capabilities based on user permissions
- Regular security audits and vulnerability testing should be conducted
- Clear documentation on security best practices should be provided

4. Consistent

Integration points must maintain perfect consistency with the core platform:

- Data queries through APIs should return identical results to the same queries performed through the administrative console
- Business logic should be applied uniformly regardless of access method
- Transactional operations (member registrations, event signups, payments) should function identically whether initiated via API or user interface
- Data validation rules should be consistent across all access methods
- Status codes and error messages should align with actions performed in the administrative interface

5. Performant

APIs should deliver robust performance without arbitrary limitations:

- Request limits should be reasonable and based on actual system capabilities, not artificial constraints
- Rate limits, if necessary, should be clearly documented and configurable based on customer needs
- Batch operations should be supported for efficient processing of multiple records
- Response times should be optimized and consistent with performance expectations
- Pagination options should be available for handling large data sets efficiently
- Performance metrics should be published and regularly updated

6. Reliable

Integration reliability must match or exceed the core platform's stated reliability:

- Uptime guarantees should extend to API availability
- Results should be idempotent—identical requests should produce identical results
- Clear status reporting should be available for API health
- Monitoring tools should be provided to track API performance and availability
- Scheduled maintenance windows should be clearly communicated
- SLAs should explicitly include API reliability metrics

7. Resilient

Critical transactions must be designed with recovery mechanisms:

- Clearly defined error handling protocols must be implemented and documented
- Retry mechanisms should be built in for transient failures
- Transaction rollback capabilities should be available for failed operations
- Comprehensive error messages should provide actionable information
- Logging should be sufficient to troubleshoot failed transactions
- Recovery procedures should be documented for various failure scenarios

8. Deferential

APIs must respect system of record hierarchies:

- Clear designation of authoritative data sources should be established
- Merge operations should defer to the system of record for conflict resolution
- Delete operations should follow proper cascading rules based on the core system
- Data sovereignty principles should be respected when multiple systems interact
- Change history should be maintained across integration points
- Conflict resolution mechanisms should be documented and consistent

9. Uniqueness

Record uniqueness must be enforced consistently:

- Unique identifiers should be generated and maintained according to documented standards
- Duplicate detection algorithms should match those in the core platform
- Merge procedures should be available via API with the same capabilities as the administrative interface
- Identity resolution frameworks should be documented
- Cross-system identifier mapping should be supported
- Audit trails for identity changes should be maintained

10. Supported

Integration should receive the same level of support as the core platform:

- APIs should be version-controlled with clear support timelines
- Upgrade paths should preserve API functionality
- Backward compatibility should be maintained for a documented period
- Clear governance and ownership of APIs should be established within the vendor organization
- Support staff should be trained on API functionality
- Developer support resources should be readily available
- APIs should be clearly identified as official vendor products, not third-party additions

This framework provides both vendors and customers with clear standards for evaluation, promoting transparency and setting expectations for modern association management software integration capabilities.