

Certification Process

Ecosystem Partners

Table of Contents

[1 Overview 2](#_Toc64297419)

[2 Test requirements and scope 2](#_Toc64297420)

[2.1 Front end ecommerce implementation 2](#_Toc64297421)

[2.2 Execution of Digital River’s validation scenarios 2](#_Toc64297422)

[2.3 Scope 3](#_Toc64297423)

[2.3.1 Ecosystem partners 3](#_Toc64297424)

[2.3.2 Digital River 3](#_Toc64297425)

[2.4 Out of scope 3](#_Toc64297426)

[3 Certification phases 3](#_Toc64297427)

[3.1 Planning and prerequisites 3](#_Toc64297428)

[3.2 Tool development and test site validation by ecosystem partner 4](#_Toc64297429)

[3.3 Security validation by Digital River 4](#_Toc64297430)

[3.4 Ecommerce front end test site validation by Digital River 7](#_Toc64297431)

[3.5 Production payment validation by Digital River 8](#_Toc64297432)

[4 Issues and prioritization 9](#_Toc64297433)

[4.1 Issue triage process 9](#_Toc64297434)

[4.2 Defect severity matrix 9](#_Toc64297435)

[4.3 Priority matrix 10](#_Toc64297436)

[5 Glossary 10](#_Toc64297437)

[Production Trial Environment 11](#_Toc64297438)

# Overview

This document explains the quality and certification process for ecosystem partners who create commerce connectors using Digital River’s build standards. Only the initial integration is defined in this document. Client integrations into the ecosystem are not covered.

As part of the connector quality and security certification process, Digital River requests that ecosystem partners provide an ecommerce front end, created in your ecosystem, that demonstrates how the connector functions. More specifically, the front-end must:

* Include a test commerce site
* Contain an admin interface for conducting configurations and customer service
* Allow orders to be placed, using all available payment methods, in the Production Test Environment (PTE)
* Execute a minimal scenario set, provided by Digital River, that provides evidence of your connector's health
* Provide access to a source code wrapping SDK that enables the ecommerce workflow
* Supply us credentials for both the admin portal and a test customer account

# Test requirements and scope

## Front end ecommerce implementation

Using the ecommerce front end application, the ecosystem partner must build a store in their commerce ecosystem. Digital River needs administrative access so we can validate how API keys are setup, SKU’s are created, and customer service functions.

The front end must validate that a successful integration exists between the partner’s commerce system and Digital River’s reference library. Digital River performs functional testing of the integration and verifies the flow to our systems. We also validate error handling and implementation efficiency (e.g., not making redundant API calls).

## Execution of Digital River’s validation scenarios

Digital River provides ecosystem partners basic scenarios to execute via their ecommerce front-end system. The scenarios we provide are in addition to those planned by the partner. However, the scenarios we supply do not cover all the requirements. Instead, they are meant to ensure minimal system functionality prior to commencing the certification process.

## Scope

### Ecosystem partners

* Ensure that the ecommerce front end uses the build standards to implement all the agreed upon requirements
* Prior to the certification handoff, execute Digital River’s test scenarios and then provide us details on the results

### Digital River

* Use the provided storefront and administrative portal to functionally validate the commerce system’s reference library implementation
* For each implemented payment method, create live transactions in the production environment
* Document test scenarios, requirements coverage, pass/fail rates, open issues, and risks
* Assess whether the source code is securely developed
* For both the admin portal and the commerce store, test the strength of the application’s security controls

## Out of scope

This certification process does not include cross browser, mobile, and performance testing. However, they should not be considered as out of scope for the overall project and its deliverables.

# Certification phases

For each phase in the certification process, this section identifies the major tasks, the exit criteria and deliverables, and the responsible parties.

## Planning and prerequisites

During the planning phase, the ecosystem partner and Digital River agree on the scope of validation for each subsequent phase in the certification process.

|  |  |  |
| --- | --- | --- |
|  | **Ecosystem partner** | **Digital River** |
| **Major tasks** | * Creates and agrees upon functional specs and documents * Devises a schedule that lists major milestones | Same as ecosystem partner |
| **Exit criteria and deliverables** | Digital River is provided with test scenarios | The system integrator is provided with the test scenarios they must execute and those Digital River plans to execute |

## Tool development and test site validation by ecosystem partner

The ecosystem partner must demonstrate that they have met all the requirements. This is accomplished by using the build standards to implement a test site on the ecommerce front end. The partner is also responsible for testing the implementation and providing Digital River their test cases and execution status. This validation is completed against Digital River’s Production Test Environment (PTE).

|  |  |  |
| --- | --- | --- |
|  | **Ecosystem partner** | **Digital River** |
| **Major tasks** | * Creates a store using the ecommerce front end and integrates it with Digital River’s PTE * Delivers planned test cases to Digital River | * Provides the ecosystem partner the test scenarios that they must execute * Shares the tests that we plan to execute with the ecosystem partner |
| **Exit criteria and deliverables** | * All planned and reviewed scenarios are executed * No priority 1 defects are open * After consulting with Digital River, any open priority 2 or 3 defects are either deferred with estimated resolution timelines Ecommerce front end demoed for Digital River | In the event of open priority 2 or 3 defects, work with the ecosystem partner to determine the resolution timeline |

## Security validation by Digital River

As part of preparing for security certification, the ecosystem partner must be able to affirmatively answer the following questions:

3.3.1 Authorization

* Are the authentication and authorization checks correctly placed?
* Does the execution terminate after invalid requests? In other words, does the execution stop when authentication or authorization checks fail?
* Are the checks correctly implemented? Are there any backdoor parameters?
* Is the check applied on all the required files and folders within the web root directory?
* Are security checks placed prior to input processing?
* For container-managed authentication, is authentication based on web methods only and applied to all resources?
* Is a complexity check enforced on the password?
* Does the application ensure that the password is not disclosed to the user, or written to a file, log, or console?
* Does the application design require server authentication (e.g., anti-spoofing measures)?
* Does the application support password expiration?

3.3.2 Business logic and design

* Are there unused business logic configurations?
* If request parameters are used to identify business logic methods, are user privileges properly mapped to their allowed methods and actions?
* Does the application check whether unexposed instance variables are present in form objects that get bound to user inputs? If present, does the application check whether they have default values?
* Are the checks correctly implemented? Are there any backdoor parameters?
* Are the checks applied on all the required files and folder within the web root directory?
* Are there any default configurations, such as Access – ALL?
* Does the configuration get applied to all files and users?
* Does the centralized validation get applied to all requests and inputs?
* Does the centralized validation check block all special characters?
* Are all special request types validated?
* Does the design maintain a list of parameters and features that are excluded from validation?
* Are there any known flaws, such as DWR, in the API’s or other technology?
* Does the design framework provide any inbuilt security control, such as <%:%> in ASP.NET MVC? Does the application take advantage of these controls?
* Are privileges reduced whenever possible?
* Is the program designed to fail gracefully?
* Are all entry points and trust boundaries identified by the design and in risk analysis?

3.3.3 Cryptography

* Are passwords stored in encrypted format?
* Are database credentials stored in encrypted format?
* Is the data sent on an encrypted channel? Does the application use HTTP client for making external connections?
* Is all PII and sensitive information encrypted when sent over the network?
* Does the application ensure that keys are not held in code?

3.3.4 Data management

* Is data validated on the server side?
* For each request, are HTTP headers validated?
* Is all XML input data validated against an agreed upon schema?
* Is output that contains untrusted data supplied input that has the correct encoding URL type?
* Is the correct encoding applied to all data outputted by the application?

3.3.5 Design

* Are there default values?
* Do default values get initialized before form binding?

3.3.6 General

* Did you ensure there are no backdoor or unexposed business logic classes?
* Are the application’s external libraries, tools, and plugins up to date? Is there a process in place to keep them updated?
* Are classes that contain security secrets (like passwords) only accessible through protected API’s?
* Does the application ensure that plain text secrets are not stored in memory for extended periods of time?
* Are array bounds checked?
* Is all sensitive information used by the application identified?

3.3.7 Input validation

* Does the application constrain and validate input data by type, length, format, and range?
* Are all untrusted inputs validated?

3.3.8 Logging and auditing

* Did you ensure that the application does not log any information, passwords, or other sensitive information?
* Does the application create audit logs for both successful and failed connection attempts?
* Is there a process in place to read audit logs for unintended and malicious behavior?

3.3.9 Session management

* Does the design securely handle session management?
* Does the design ensure that session management parameters are not passed in URLs?
* Does session management determine that:
* Cookies expire in a reasonably short time?
* Cookies are encrypted?
* Data is validated?
* ID is complex?
* Storage is secure?

3.3.10 User management and authentication

* Are user and role-based privileges documented?
* Does the application ensure that authentication cookies are not persisted?
* Are authentication cookies encrypted?
* Does the application ensure that authentication credentials are not passed by HTTP GET methods?
* Do authorization checks support page and directory level granularity?
* Is authorization based on clearly defined roles?
* Does the application ensure that authorization cannot be circumvented by parameter manipulation?
* Does the application ensure authorization cannot be bypassed by cookie manipulation?

3.3.11 Web services

* Are web service endpoint addresses in Web Services Description Language (WSDL) checked for validity?
* Are unnecessary Web service protocols, such as HTTP GET and HTTP POSTs, disabled?

## Ecommerce front end test site validation by Digital River

During this phase, Digital River uses the ecommerce front end and the admin portal to validate the build standards implementation. We re-execute the scenarios provided to the ecosystem partner and also use our own. These include logging validation and other backend Digital River specific scenarios.

This is the final validation in the PTE environment. The next phase conducts validation in Digital River’s production environment.

|  |  |  |
| --- | --- | --- |
|  | **Ecosystem partner** | **Digital River** |
| **Major tasks** | * Completes validation against the ecommerce front end * Resolves all priority 1 issues * After consulting with Digital River, defers any open priority 2 or 3 defects with estimated resolution timelines * Makes the ecommerce front end available to Digital River and provides URLs and credentials | * Decides whether the requirements are met * Examines the implementation’s best practices * Determines whether unnecessary or inefficient calls exist (e.g., identical requests or failing to send SKUs deltas) * Reviews the hard-coded data |
| **Exit criteria and deliverables** | * All prior tasks are complete * Each priority 1 issue is resolved * Planned tests provided by Digital River are executed and documented in an Application Lifecycle Management (ALM) or automation tool |  |

### 3.5 Production payment validation by Digital River

During this phase, Digital River tests the implementation in our production environment using live payment methods. This leverages the commerce site created by the system integrator and points to Digital River’s production environment.

|  |  |  |
| --- | --- | --- |
|  | **Ecosystem partner** | **Digital River** |
| **Major tasks** | * Updates the production keys in the commerce ecosystem * Completes security reviews and resolves any outstanding issues |  |
| **Exit criteria and deliverables** | * All end-to-end tests planned by Digital River are executed * Security reviews are complete * No open issues exist | The internal stakeholder review of the quality certification documents is complete |

# Issues and prioritization

## Issue triage process

Regular project meetings are held to discuss open issues. This ensures these issues are addressed by either the commerce partner or Digital River. During these meetings, Digital River reviews all open issues that we have documented.

The triage process include representatives from project management, product, engineering, and, as needed, security.

## Defect severity matrix

|  |  |  |
| --- | --- | --- |
| **Severity Level** | **Definition** | **Examples** |
| **1 (critical)** | These defects result in a crash, hang, serious data corruption/loss, or pose a critical security risk. Because these defects do not have a feasible workaround, customers are negatively impacted. And since these defects have no workaround, they result in a halt to the commerce flow. | * Unrecoverable crashes and hangs * Serious data corruption or loss * Fatal assertion failures **Note**: A non-fatal assertion that does not cause exception after ignoring is not a severity level one defect * High risk security issues * Unrecoverable errors * Severe functionality problems that make it impossible to use a device |
| **2 (major)** | These defects cause a frequently used feature to operate contrary to its specification. They affect application flow but have an existing workaround. | * Non-fatal assertion failures * Major functionality issues * Memory leaks that result in device lockup * Security bugs without significant vulnerability * Frequent deadlocks * Occasional crashes that are recoverable on restart * Government regulatory or legal issues |
| **3 (average)** | These defects impair system usability, but all main functions remain available. This would be a defect or design change request to meet a mandatory requirement. | * Minor functionality issues * Major UI issues * Mandatory requirements * Feature that enables important new customer functionality |
| **4 (minor)** | These defects are primarily small, cosmetic distractions that should be corrected.  They do not affect any of the application’s main functions and generally have very low customer impact. | * Spelling or localization errors * Poorly worded phrases |

## Priority matrix

|  |  |  |
| --- | --- | --- |
| **Priority Level** | **Definition** | **Examples** |
| **1 (high)** | The issue must be fixed prior to the next phase of testing or launch. These issues are launch blockers and affect critical functionality. | * Major tasks or exit criteria of a phase cannot be verified by Digital River * Unresolved functional defects (severity level 1 or 2) during a phase |
| **2 (medium)** | The issue is not progress blocking but fixing it is highly desirable. Eventually the issue should be addressed but can be delayed until a later phase. | Unresolved functional defects (severity level 3) during a phase |
| **3 (low)** | The issue is not scheduled to be fixed and should be corrected when time and resources allow. | Unresolved functional defects (severity level 4) during a phase |

# Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Application Lifecycle Management (ALM) | This is a test case, requirements, test execution, and defect repository. The defects integrate with Jira so all project members can have visibility. |
| Production Trial Environment (PTE) | Digital River’s environment for partners and clients to try the APIs. |
| Ecosystem partner | The partner implementing Digital River’s API libraries using the build standards and the ecommerce front-end provider. |