

WATADCASH

Payment API Solutions: The Integration Brief

Prepare a practical integration brief covering the customer journey, activation, payment verification and support responsibilities.



Start with the outcome you need

A payment integration often starts with a developer asking for an API key. That is a technical requirement, but it is not yet a complete business brief. The team still needs to know what customers will buy, which payment route is intended, when an order becomes deliverable and who resolves exceptions. WATADCASH's gateway guidance asks businesses to consider activity, customer markets and the payment flow. Writing those answers first can save repeated redesign later.

Describe a real customer journey

Use one representative order to explain the intended flow. State what is sold, when the amount becomes final and what the customer receives after successful payment. A digital service activated immediately has different fulfilment needs from a physical order reviewed before shipment.

Then name the payment route you want to support. WATADCASH's public gateway material describes USDT on TRON TRC20 and payment from the WATADCASH wallet. Virtual card issuance is a separate product; it should not be interpreted as card-acquiring capability for your checkout. A brief that makes this distinction prevents a technical team from designing around a service that was never part of the offer.

Resolve activation and permissions

Opening an account and being authorised to collect payments are separate steps. WATADCASH's public pages describe service activation, account eligibility and keys with the required permissions. Put those dependencies into the project brief so that a working interface is not mistaken for an approved live payment service.

Identify the person responsible for confirming activation terms and the developer responsible for credential handling. Keep commercial approval details separate from secret values. The brief should explain which service is approved and who owns it, while the actual key belongs in the organisation's approved server-side secret storage process.

Agree on what paid means

Write a plain-language rule for fulfilment. Which verified provider state allows the order to move forward? What should the customer see while payment is pending? Who can review a discrepancy? WATADCASH's guidance says that screenshots, browser redirects and invoice-creation responses are not proof of payment.

That rule belongs in the business brief, not only in code. Support staff should know why a customer may have completed a browser step while the order still awaits confirmation. When everyone uses the same definition, the developer can build a state transition that the operations team understands and can explain consistently.

Plan references and exception ownership

Decide which references will connect the customer order, payment request and any support case. A clear identifier lets a support agent ask about the correct transaction without requesting passwords or unrelated financial records. Store the provider's identifier beside your own order reference when it becomes available.

Also assign ownership of exceptional cases. A developer may investigate a signature failure, while a business operator may review an unresolved order or ask the provider about a transfer status. Without that distinction, every unusual payment becomes an urgent technical incident even when the actual question is a commercial or customer-service decision.

Define the first release narrowly

A useful first release can cover one enabled payment route and one well-understood order type. Confirm the successful journey, then test pending states, duplicate notifications and interrupted responses. Add further services once the team can operate the first one confidently.

WATADCASH's documentation covers authentication, payment requests, cards and webhooks, but a broad documentation set is not a requirement to enable every feature at once. The brief should name the functions your business needs now and leave later capabilities as explicit future decisions. This produces a smaller, testable integration and a clearer experience for the first real customer.

What should the brief say about payment methods?

Name the exact route, currency and customer use you intend to support. Do not use a broad label such as “all online payments” when the approved service is a particular USDT or wallet payment flow.

Who should hold the API key?

The key should remain on the server, following the provider's documentation. The brief can identify the system and responsible team, but it should not contain the secret itself or distribute it to customer-facing code.

What is a sensible first milestone?

An enabled service with server-side authentication, a linked order and payment request, a verified paid-state transition and an understood exception path. A visually complete checkout without those elements is not the same milestone.

Sources and scope

References reviewed by the WATADCASH editorial team on 18 October 2026. Availability and pricing depend on the actual request and current terms; this guide is not a promise of universal access.

1. WATADCASH gateway workflow guide

<https://watadcash.com/gateway.php>

2. WATADCASH developer documentation

<https://watadcash.com/documentation/>

Review the decision before confirmation

What outcome do you need?

Who is responsible for delivery?

What are the total cost and net amount?

What proves completion?

How will you follow an exception?

<https://watadcash.com/journal/en/payment-integration-kickoff-brief/>