

# 1. Intercepting Network Requests (GraphQL & REST API Testing)

Cypress allows you to watch, modify, mock, or wait on network calls using `cy.intercept()`. You can control backend behavior during tests—useful for handling GraphQL queries, REST endpoints, and flaky backend responses.

---

## 2. Mocking GraphQL Queries with `cy.intercept()`

GraphQL requests usually hit the same endpoint (`/graphql`), so Cypress intercepts them by checking the **operation name**.

You can mock the response to return controlled test data, so your UI tests don't depend on the real backend.

Helps you test loading states, error states, and specific UI conditions.

---

## 3. API Testing with Cypress (`cy.request()`)

`cy.request()` is used to test APIs **independently of the UI**.

It can:

- Send GET/POST/PUT/DELETE calls
- Validate status codes, headers, and response bodies
- Set up or clean test data before UI tests

This makes Cypress a **full-stack testing tool**—UI + API.

---

## 4. Verifying Request & Response Payloads

Cypress allows you to assert both outgoing and incoming data:

- **Request payload:** Validate the body sent by the UI (use `intercept().as()` + assertions)
- **Response payload:** Ensure API returns expected objects, arrays, fields, and values
- **Headers & status codes:** Confirm correct API behavior