

Groovy HTTP Client — Modern, Secure, Multipart Ready A modern, Groovy friendly HTTP client built on **Java 21 virtual threads** and the standard JDK `HttpClient`. This version includes **fully redesigned multipart upload support**, a **new SecurityConfig aware constructor**, a **clean DSL for multipart**, and **enhanced test friendly behavior** with the project's `MockHttpServer`. --- ## ■ Major Enhancements (2025) ### ■ **1. Full Multipart Support** The client now supports: - **DSL style multipart** - **Static list based multipart** - **Automatic multipart boundary generation** - **Mixing multipart parts + custom headers** - **Text or file parts** - **Binary or JSON parts** - **Multipart builder exposed to user closures** ### DSL Example `groovy def resp = client.postMultipartSync("/upload") { b -> b.part { name "file1" filename "readme.txt" content "hello world" } b.part { name "meta" contentType "application/json" content '{"x":1}' } b.header("X-Test", "dsl") }` --- ### ■ **2. MultipartPart — Unified Representation** `groovy MultipartPart.text(name, stringContent) MultipartPart.bytes(name, byteArray, contentType) MultipartPart.file(name, file, contentType)` These work in: `groovy def parts = [MultipartPart.text("file1", "hello"), MultipartPart.text("file2", "world")] client.postMultipartSync("/upload", parts)` --- ### ■ **3. New Constructor: `GroovyHttpClient(SecurityConfig)` The client now reads all security settings directly from `SecurityConfig`, including: | Field | Effect | |-----|-----| | `insecureAllowed` | Enables trust | | `allowAbsoluteUrls` | Whether absolute URLs are allowed | | `allowedHosts` | SSRF protection | | `connectTimeout` / `requestTimeout` | Default request timeouts | | `cookiePolicy` | How cookies are accepted/stored | | `enableLogging` | Print request/response warnings | | `maxResponseBytes` | Response body guard | | `failureThreshold`, `resetTimeoutMs` | Circuit breaker | **Example** `groovy def cfg = SecurityConfig.testing("http://localhost:8080") def client = new GroovyHttpClient(cfg)` --- ### ■ **4. More Groovy Friendly Request Builder The DSL safe wrapper now supports: `groovy builder.header("X", "123") builder.cookie("session", "abc") builder.cookies([a:"1", b:"2"]) builder.timeout(Duration.ofSeconds(3))` And you can mix it with multipart: `groovy client.postMultipartSync("/u") { b -> b.header("X-Mode", "test") }` --- ### ■ **5. New File Download API `groovy byte[] bytes = client.downloadBytesSync("/download")` Uses cookies and headers automatically. --- ### ■ **6. Circuit Breaker Included by Default | Parameter | Default | |-----|-----| | `failureThreshold` | 5 | | `resetTimeoutMs` | 30,000 ms | Thrown exception: `CircuitOpenException` --- ### ■ **7. Stronger SSRF Protection The client now prevents: - Absolute URLs (unless allowed) - Requests to unknown hosts - Improper host rewriting --- ### ■ **8. Greatly Enhanced MockHttpServer Integration `MockHttpServer` now supports: - Multipart request validation - Expected headers - Expected cookies - Response cookies - Delayed responses - Binary body responses - Cleaner diagnostics --- ## ■ Complete Multipart API Overview ### DSL Variant `groovy client.postMultipartSync("/upload") { b -> b.part { name "avatar" filename "me.png" contentType "image/png" content bytes } }` --- ### Static List Variant `groovy def parts = [MultipartPart.bytes("raw", [1,2,3] as byte[], "application/octet-stream")] client.postMultipartSync("/upload", parts)` --- ## ■ All Constructors ### 1. Legacy `groovy new GroovyHttpClient("https://api.example.com")` --- ### 2. Positional Advanced `groovy new GroovyHttpClient("https://api", Duration.ofSeconds(2), Duration.ofSeconds(10), 5, 30000L)` --- ### 3. **SecurityConfig Constructor (recommended)** `groovy def cfg = SecurityConfig.production("https://api") def client = new GroovyHttpClient(cfg)` --- ## ■ Testing With MockHttpServer The multipart tests now pass with: `groovy server.addRequestCheck("POST", "/upload", 200).withResponseBody("OK")` Plus: - `.withExpectedHeaders([:])` - `.withExpectedCookies([:])` - multipart validation hooks --- ## ■ Download Example `groovy byte[] bytes = client.downloadBytesSync("/download") assert new String(bytes) == "hello"` --- ## ■ Troubleshooting | Symptom | Cause | |-----|-----| | `MissingMethodException: part()` | Occurs when DSL incorrectly scoped | | Multipart empty? | Closure using delegate incorrectly | | Cookie mismatch | `MockHttpServer` expected cookies missing | | Circuit open | Too many test failures | --- ## ■ Summary `GroovyHttpClient` 2025 Edition includes: - ✓ Full multipart support (DSL + static) - ✓ Strong `SecurityConfig` constructor - ✓ Virtual thread core - ✓ Cookie management - ✓ Circuit breaker - ✓ SSRF protections - ✓ Highly testable with `MockHttpServer` - ✓ Production ready hardening defaults ---