| **Feature** | **RestTemplate** | **Java 11+ HttpClient** |
| --- | --- | --- |
| **Threading** | Blocking/synchronous | Supports both blocking and **non-blocking** async via sendAsync()  |
| **HTTP/2**  | HTTP/1.1 only | **Native HTTP/2 support** for better performance  |
| **API Design** | Verbose, limited builder patterns | **Fluent builder**, clean separation of request and response  |
| **Dependency Footprint** | Requires Spring context | **Built-in** to JDK—no external dependencies  |
| **Control** | Higher-level convenience methods | Low-level: full control over headers, connection, etc.  |

*public class* HttpClientTest {
 *public void* makeHttpRequests() {

 *// Blocks the calling thread; (Blocking, Synchronous)
 // Uses HTTP/1.1* RestTemplate rt = *new* RestTemplate();
 String body = rt.getForObject("https://api.example.com", String.*class*);
 System.*out*.println(body);

 *// Java 11 HttpClient (Async, Non-blocking)
 // Uses HTTP/2* HttpClient client = HttpClient.*newHttpClient*();

 HttpRequest request = HttpRequest.*newBuilder*()
 .uri(URI.*create*("https://api.example.com"))
 .build();

 client.sendAsync(request, *HttpResponse*.BodyHandlers.*ofString*())
 .thenApply(*HttpResponse*::body)
 .thenAccept(System.*out*::println);
 }
}