[Karate Home Page](https://github.com/karatelabs/karate) | Original Google Drive [Link](https://docs.google.com/document/d/1ETTrdMVcBXaPjdKY-_67zCWBsi2Ctc5DIQUIfr02H7A/edit?usp=sharing) for this document | Please send corrections and feedback to [@ptrthomas](https://twitter.com/ptrthomas)

|  |  | **REST-assured** | **Karate** | **References / Comments** |
| --- | --- | --- | --- | --- |
| 1 | BDD Syntax | Yes | Yes |  |
| 2 | True DSL | No. Fluent Interface.  Also IDE formatting is a [challenge](https://groups.google.com/forum/#!topic/rest-assured/e6Sx_4way2M) | Yes | [DSL vs Fluent Interface.](https://ayende.com/blog/2984/dsl-vs-fluent-interface-compare-contrast)  Also see (24) and (25) |
| [3](https://ayende.com/blog/2984/dsl-vs-fluent-interface-compare-contrast) | Runs on the JVM | Yes | Yes |  |
| 4 | Implementation | Java and Groovy | Java |  |
| 5 | Code-base Size | Large.  46,000 lines of code (source: [OpenHub](https://www.openhub.net/p/rest-assured/estimated_cost)) | Large.  50,000 lines of code (source: [OpenHub](https://www.openhub.net/p/karate/estimated_cost)) | Karate has API-mocks, perf-testing and UI automation as well. |
| 6 | Mature | Yes. Inception 2010.  Lots of blog posts, tutorials and StackOverflow posts.  In the ThoughtWorks Tech Radar [at 5 years](https://www.thoughtworks.com/radar/tools/rest-assured), appeared only once. | Yes. Inception February 2017.  Clear signs of [wide and rapid adoption](https://github.com/intuit/karate/wiki/Community-News).  In the [ThoughtWorks Tech Radar](https://twitter.com/KarateDSL/status/1120985060843249664) within just 2 years and rating [upgraded within 1 year](https://twitter.com/KarateDSL/status/1262719979104817152). | Karate has 5000 GitHub “stars” in 4 years, which took more than 10 years for REST-assured.  REST-assured does not even show up on [Stack Overflow trends](https://twitter.com/KarateDSL/status/1392699279337136129). |
| 7 | JsonPath Implementation | Groovy GPath | JayWay JsonPath | GPath has some [limitations](https://github.com/rest-assured/rest-assured/issues/753) and [updates are not possible](https://groups.google.com/forum/#!topic/rest-assured/uNGxBLB5uvI) |
| 8 | XPath implementation | Groovy GPath and “XMLSlurper”. Standard XPath is also supported, but paths that return XML nodes cannot be used in assertions. Updating an XML document is not possible. | [W3C](https://www.w3.org/TR/xpath/) standard XPath using the Java built-in XML lib. You can even update XML documents using XPath. |  |
| 9 | HTTP Client | Apache 4.X, but the code depends on [deprecated APIs](https://hc.apache.org/httpcomponents-client-ga/httpclient/apidocs/deprecated-list.html).    There are [some concerns](https://github.com/rest-assured/rest-assured/issues/783) with this [design](https://groups.google.com/forum/#!topic/rest-assured/cvnBgh_ErAY).    More details in this [issue](https://github.com/rest-assured/rest-assured/issues/497). | Pluggable (future-proof), you can even implement your own.  Karate also has [minimal maven dependencies](https://gist.github.com/ptrthomas/b24b8d5c375412d1378414d824c01bb5) - and the Apache libraries are “shaded” to not cause conflicts. | Karate even has an option to [mock a servlet container](https://github.com/intuit/karate/tree/master/karate-mock-servlet) because of this abstraction. |
| 10 | Quick Start / Project Scaffolding | No | Yes (Maven Archetype).  There’s also a [standalone executable](https://github.com/intuit/karate/tree/master/karate-netty#standalone-jar) and [ZIP Release](https://github.com/intuit/karate/wiki/ZIP-Release). | Dev onboarding experience much better with Karate. Archetype Includes working example. |
| 11 | Test-Scripting Language | Java | Karate-Script (Cucumber / Gherkin)  [Java is also supported](https://twitter.com/KarateDSL/status/1353969718730788865) in version 1.0 onwards. | No Java knowledge needed for Karate |
| 12 | Test Scripts have to be compiled | Yes | No | Tests are plain-text. No IDE required for Karate |
| 13 | IDE Support | Yes. Intelli-Sense, Auto-Complete and Refactoring work for Java and POJO-s | Partial. Eclipse and IntelliJ have Cucumber plugins that work well and have pretty good syntax coloring.  Karate has a [Java API option](https://github.com/intuit/karate/tree/develop#java-api) in version 1.0 onwards. | Since you can re-use JSON payloads across tests, the “re-factorability” aspect is covered as well. |
| 14 | Step Through / Debug-ability | Yes. Java + IDE Support. | Even better ! Debug in Visual Studio Code with [step-back and *even* hot-reload](https://twitter.com/KarateDSL/status/1167533484560142336) (since v0.9.5).  also see the [Dev-mode HTML report](https://twitter.com/KarateDSL/status/935029435140489216): steps, error diagnostics and HTTP logs in-line, and there is a Java API. | Also see (42) |
| 15 | Test Runner | Any, bring your own. TestNG or JUnit will work. | JUnit is supported and can co-exist with TestNG in the same project if needed.  Even JUnit is [optional](https://stackoverflow.com/a/65578167/143475) | And Karate’s parallel execution capability is in “core”, independent of even Maven or any unit-testing framework. |
| 16 | Tags / Groups Built In | No (have to use TestNG or equivalent) | Yes |  |
| 17 | Extend with custom routines via... | Java code | JavaScript | No need to compile, and easier for non-programmers. |
| 18 | Re-use Java code | Yes | Yes (via [JavaScript interop](https://github.com/intuit/karate#calling-java)) |  |
| 19 | Validate All Payload values in one step | Not built-in. You *need* to use external libraries such as Hamcrest.   * No easy way to do a “deep equals” comparison for nested objects. * No way to “ignore” fields - for e.g. id / date / time values which are dynamic | Karate natively supports a “*deep equals*” and “*contains in any order*” assertion for JSON, JSON arrays and XML, ***and*** lets you ignore chosen fields at the same time.  This is super-important for [GraphQL](https://stackoverflow.com/a/47469363/143475). | IMO the **biggest** limitation of REST-Assured:   * [Example 1](https://groups.google.com/forum/#!topic/rest-assured/vEXHeVQY1LQ) * [Example 2](https://groups.google.com/forum/#!topic/rest-assured/0NoxuFRhZtE) * [Example 3](https://groups.google.com/forum/#!topic/rest-assured/TwgDEKkBuEE) * [Example 4](https://groups.google.com/forum/#!topic/rest-assured/tAFv1CysNOM) |
| 20 | Built-in data-type, conditional-logic and RegEx validations | No | Yes, includes [RegEx and Macros](https://github.com/intuit/karate#ignore-or-validate) |  |
| 21 | Validate schema of all elements in a JSON array in one step | No | [Yes](https://github.com/intuit/karate#match-each) |  |
| 22 | Built-in JSON Schema and XML Schema validation support | Yes | [RegEx and Macros](https://github.com/intuit/karate#ignore-or-validate) support is sufficient (and far simpler) for most use cases. That said, users can [easily add a Java lib](https://github.com/intuit/karate/blob/master/karate-demo/src/test/java/demo/schema/schema.feature) via Karate’s Java interop - if needed. | For details on how Karate’s approach is simpler and more intuitive than JSON (or XML) Schema see [this link](https://twitter.com/KarateDSL/status/878984854012022784). |
| 23 | Native support for expressing JSON or XML in test-scripts | No  **"{ \"name\": \"Billie\" }"**  **"<cat name=\"Billie\"></cat>"** | Yes  **{ name: 'Billie' }**  **<cat name="Billie"></cat>** | No need to use double-quotes or “escape” characters.  You can also read from files and re-use. |
| 24 | Example – JSON assertions | **@Test public void**  **lotto\_resource\_returns\_200\_with\_expected\_id\_and\_winners() {**  **when().**  **get("/lotto/{id}", 5).**  **then().**  **statusCode(200).**  **body("lotto.lottoId", equalTo(5),**  **"lotto.winners.winnerId", containsOnly(23, 54));**  **}** | **Scenario: lotto resource returns 200 with expected id and winners**  **Given path ‘lotto’, 5**  **When method get**  **Then status 200**  **And match $.lotto.lottoId == 5**  **And match $.lotto.winners[\*].winnerId contains only [23, 54]** | Matching built-in, and more readable syntax. Note the simpler way to specify path parameters without placeholders.  For REST-assured, IDE formatting is a known [challenge](https://groups.google.com/forum/#!topic/rest-assured/e6Sx_4way2M). |
| 25 | Example - GET with params | **given().**  **param("key1", "value1").**  **param("key2", "value2").**  **when().**  **get("/somewhere").**  **then().**  **body(containsString("OK"));** | **Given param key1 = ‘value1’**  **And param key2 = ‘value2’**  **And path ‘somewhere’**  **When method get**  **Then match response contains ‘OK’** | Karate is a [true DSL](https://ayende.com/blog/2984/dsl-vs-fluent-interface-compare-contrast). No syntax “noise”, no unnecessary symbols or punctuation. No need to worry about indenting a giant “one liner” of Java code. |
| 26 | Extracting multiple data-elements for reuse in subsequent HTTP calls | Convoluted.  The Fluent Interface which is supposed to be the main highlight of REST-Assured actually [gets in the way](https://github.com/rest-assured/rest-assured/wiki/usage#extracting-values-from-the-response-after-validation) here. More [examples](http://stackoverflow.com/questions/21166137/rest-assured-is-it-possible-to-extract-value-from-request-json). | Easy. You can even use JsonPath to extract JSON chunks or arrays and save them to variables for use in later steps.  For XML, XPath does the same. | Some of the [quirks](https://stackoverflow.com/questions/45112425/how-do-i-get-the-value-of-a-key-if-it-contains-a-space-in-rest-assured-serenit) of the REST-assured JsonPath implementation get in the way as well. |
| 27 | Can update a given JSON or XML using a path expression | [No](https://groups.google.com/forum/#!topic/rest-assured/uNGxBLB5uvI).  Especially for data-driven tests, updating nested JSON is [near impossible](https://stackoverflow.com/questions/22469280/how-do-i-send-jsonobject-with-nested-values-as-post-request-in-rest-assured). | Yes. There are actually multiple ways to update payloads: a) by path b) using embedded expressions and c) via a built-in string replacement keyword. | You can even modify a response and re-use it ‘as-is’ as the next request. |
| 28 | Data Driven Testing | No (have to use TestNG or equivalent)  [REST-Assured Example](https://github.com/basdijkstra/rest-assured-workshop/blob/d9734da98bfcd8087055bdcd78545581dd23cb77/src/test/java/answers/RestAssuredAnswers2Test.java) | Yes. Can even use dynamic JSON or even CSV as a data-source.  [Karate Example](https://gist.github.com/ptrthomas/0f5c00582b345a74d20261fb346225d0) |  |
| [29](https://gist.github.com/ptrthomas/0f5c00582b345a74d20261fb346225d0) | SOAP support | [No](https://groups.google.com/forum/#!msg/rest-assured/y5IBklgfY88/frLJT_gE954J) | Yes | Plus, Karate’s XML support is far more [flexible and easier to use](https://github.com/intuit/karate/blob/master/karate-junit4/src/test/java/com/intuit/karate/junit4/xml/xml.feature). |
| 30 | HTTPS / SSL without certificates | Although there is “[relaxed](http://stackoverflow.com/questions/30423660/rest-assured-userelaxedhttpsvalidation-and-java-6-issues)” HTTPS, a certificate is [needed](http://static.javadoc.io/com.jayway.restassured/rest-assured/2.6.0/com/jayway/restassured/config/SSLConfig.html) in some cases | Yes |  |
| 31 | Built-in support for switching environment config | No  Also config is somewhat [convoluted](https://github.com/rest-assured/rest-assured/issues/239) in REST-Assured | Yes.  Adding a new variable to a test is just one step: edit **karate-config.js** | In REST-assured, you have to use something like a dependency-injection framework (or roll your own) to read properties files. |
| 32 | File Upload / Multipart Support | Partial / Buggy  [Libraries](http://stackoverflow.com/questions/32722271/unable-to-perform-multipart-form-upload-using-rest-assured) | [Content-Type](https://github.com/rest-assured/rest-assured/issues/377) | [Dependencies](https://github.com/rest-assured/rest-assured/issues/585) | ‘multipart/related’ [not supported](https://github.com/rest-assured/rest-assured/issues/623) | [questions](https://github.com/rest-assured/rest-assured/issues/374) on ‘multipart/mixed’ | Yes |  |
| 33 | URL encoded HTML Form data | Yes | Yes |  |
| 34 | Cookies | [Some](https://groups.google.com/forum/#!topic/rest-assured/mb823qPuzNs) [Limitations](https://groups.google.com/forum/#!topic/rest-assured/EVmxoA5UMYo) | Yes | Can be [configured one-time](https://github.com/intuit/karate#configure) for all subsequent requests, just like headers. |
| 35 | Auth Schemes out of the box | Yes | No (but [easily pluggable](https://github.com/intuit/karate#http-basic-authentication-example) via re-usable scripts or JavaScript without needing to write Java code) |  |
| 36 | Custom Auth | Java code (needs compilation). Existing mechanism is [not extensible](http://stackoverflow.com/questions/42340906/unable-to-login-through-application-when-testing-using-restassured). | Unified plug-in system via JavaScript (no compilation needed) |  |
| 37 | Parallel Execution of Tests | No. While [some teams](https://groups.google.com/d/msg/rest-assured/eDy1FFEiY5s/PsFBfCrHZN0J) seem to have had success running REST-assured in parallel, there are [some cases](https://github.com/rest-assured/rest-assured/issues/534) in which multi-threading is not supported. Also see [this thread](https://github.com/rest-assured/rest-assured/pull/851). The creator has also confirmed that “[REST Assured is not 100% safe to use in multi-threaded scenarios](https://groups.google.com/d/msg/rest-assured/NEhrDwIYQag/c6KYLTiSBgAJ)”. | [Yes](https://github.com/intuit/karate#parallel-execution)  Even if you run tests in parallel, reports and logs are collected per test and HTML reporting works as you would expect. | This is a **critical** requirement for HTTP integration tests which typically take a much longer time than unit tests. |
| 38 | Floating-point precision | All numbers are [converted to float](http://www.baeldung.com/rest-assured-tutorial) and you shouldn’t forget to use floats (not the default double) in assertions.  **get("/odd")**  **.then().assertThat()**  **.body("odd.ck", equalTo(12.2f));** | Numeric assertions work just as you expect and even auto-conversion to BigDecimal happens if needed.  **Given path ‘odd’**  **When method get**  **Then $.odd contains { ck: 12.2 }** | Even this works:  **And $.odd.ck == 12.2000000000000** |
| 39 | Lines of Code Needed to express a test | More. By nature, Java is verbose and especially if you depend on POJO representations of payloads - you need more Java code in place. | Less.  [This particular comparison](https://twitter.com/KarateDSL/status/873035687817117696) shows a dramatic difference, 431 lines of code reduced to 67 | Another example of how Java “gets in the way” - the [contortions you need to do](https://stackoverflow.com/questions/15531767/rest-assured-generic-list-deserialization) to handle JSON arrays in REST-assured. |
| 40 | Test Reports Built-in | No, you have to use JUnit, TestNG or equivalent for test reporting. | Karate has text and HTML reports out of the box and you get the option of choosing from the Cucumber ecosystem of 3rd party reports. | [Here is an example](https://github.com/intuit/karate/tree/master/karate-demo#generate-report) of the very nice-looking reports you can get by using the [cucumber-reporting](https://github.com/damianszczepanik/cucumber-reporting) library. |
| 41 | Test any Java servlet or HTTP resource without a container | REST-assured has support for “out-of-container” testing of specifically [Spring-MVC](https://github.com/rest-assured/rest-assured/wiki/Usage#spring-mock-mvc-module) but your tests will be “hard-coded” in this mode.  There is no support for things like JAX-RS or custom servlets or controllers - and for these you have to deploy to an app-server. | Karate v0.5.0 onwards has support for testing **any** servlet by providing extension points for teams to write an adapter.  The huge advantage of Karate’s approach is that the same test-script can be re-used for http-integration tests **without** changes. | This is possible because of Karate’s pluggable abstraction of the HTTP Client. Refer to [the documentation](https://github.com/intuit/karate/tree/master/karate-mock-servlet) for more details.  You will be able to quickly implement a custom adapter for any Java server-side stack in a similar way. |
| 42 | Report includes HTTP request and response logs in-line | No. | Karate includes [HTTP request and response logs](https://github.com/intuit/karate/tree/master/karate-demo#generate-report) in the JSON report output. If you use the **print** keyword, the console output appears in the report as well, which is great for troubleshooting. All this works even when tests are run in parallel. |  |
| 43 | Construct JSON or XML from scratch using just path expressions | No. | Best explained via [some examples](https://gist.github.com/ptrthomas/d6beb17e92a43220d254af942e3ed3d9). |  |
| 44 | Test Doubles or Mocks built-in | No.  You have to use 3rd party frameworks such as [Wiremock](http://wiremock.org) | Karate 0.7.0 onwards has support for [creating API mocks](https://github.com/intuit/karate/tree/master/karate-netty) that can fully simulate stateful CRUD. Having both a client and server in the same unified framework keeps things simple and you can move fast. | Karate is a superior alternative to Wiremock, here’s a [comparison](https://twitter.com/ptrthomas/status/945688762004119552).  Also see this [perf benchmark](https://twitter.com/KarateDSL/status/1083775218873581571). |
| 45 | Performance Testing | No.  Also see [37] | You can re-use Karate tests as [Gatling performance tests](https://twitter.com/ptrthomas/status/986463717465391104).  You can compose multiple Karate feature files or “work-flows” into a single performance-test and use Gatling to define the load-model (ramp-up, concurrent users, etc) | This alone is reason to choose Karate and no other open-source test-automation framework has this option.  In 0.9.0 onwards you can test [any Java code](https://twitter.com/KarateDSL/status/1042055022052179968), not just HTTP |
| 46 | Websockets support | [No](https://github.com/rest-assured/rest-assured/issues/850). | Karate 0.9.0 onwards has [support for websockets](https://github.com/intuit/karate/issues/395#issuecomment-434745214) and even generic async / await support (see below) |  |
| 47 | Async Support | No, you have to use Java code or a library like [Awaitility](https://github.com/awaitility/awaitility/wiki/Usage) | Karate 0.9.0 has built-in support via a very [elegant API](https://github.com/intuit/karate/issues/395#issuecomment-434745214). | Also see <https://twitter.com/KarateDSL/status/1417023536082812935> |
| 48 | Retry Support | No, you have to write custom Java logic. | Karate 0.9.0 onwards has this built-in - where you [only specify the condition](https://twitter.com/KarateDSL/status/1063415110197080065) to be polled for. |  |
| 49 | CSV file support | No, you have to use Java code or a library. | In Karate 0.9.0 you can read a CSV file and it will be auto-converted to JSON | Karate also has YAML file support out of the box. |
| 50 | Web-UI Automation | No. | In Karate 0.9.5 onwards [Web-UI automation](https://twitter.com/KarateDSL/status/1163296799999705095) is possible. | Chrome / CDP, WebDriver, Appium and Playwright supported. |
| 51 | Distributed Testing | No. | In Karate 0.9.5 onwards, there is [built-in support](https://github.com/intuit/karate/wiki/Distributed-Testing) to split a test-suite and orchestrate the execution across multiple cloud or hardware / cluster nodes |  |

### 