Karate Home Page | Original Google Drive Link for this document | Please send corrections and feedback to @ptrthomas

Mode				·	
No. State of Private State And Defendance No. State Interface			REST-assured	Karate	References / Comments
Also DE Formattings is challenge. 4. Implementation 5. Code-base Size Large. Large. Large. Large. South of the comment	1	BDD Syntax	Yes	Yes	
3	2	True DSL	No. Fluent Interface.	Yes	DSL vs Fluent Interface.
3			Also IDE formatting is a challenge		Also see (24) and (25)
Mature	3	Runs on the IVM		Yes	, , ,
Code base Size Large. As Large. As Large. As Code (source: Operation) So,000 lines of code (source: Operation) Source lines of code (source: Operation) Source: Operation	-				
6 Mature Ves. Inception 2010. Late of blog posts, tutorials and StackOwriftow posts. In the ThoughtWorks Tech Radar at Susan, appeared only once. Growy GPath and "Multi-lumps" Standard With his also supported, the paths the recent Multi-lumps and process. Mark Thought Work is also supported, the paths the recent Multi-lumps and process. Mark Thought Work is also supported, the paths the recent Multi-lumps and process. Mark Thought Standard With his also supported, the paths the recent Multi-lumps and supported, the paths the recent Multi-lumps and process. Mark Thought Standard With his also supported, the paths the recent Multi-lumps and process. Mark Thought and Standard Multi-lumps and the paths the recent Multi-lumps and process. Mark Thought and Standard Multi-lumps and the paths the recent Multi-lumps and process. Mark Thought and Standard Multi-lumps and the paths the recent Multi-lumps and process. Mark Thought and Standard Multi-lumps and the paths the recent Multi-lumps and process. Mark Thought and Standard Multi-lumps and the paths the recent Multi-lumps and process. Mark Thought and Standard Multi-lumps and the paths the	_		·		Karata haa ADI waaala
Mature Ves. Inception 2010. Ves. Inception February 2017. Karate has 5000 GitHub "start" Average with a start of bing posts, tutorials and StarkOverflow posts. Interfloop/Winkins Tech Radar all 5 years and rating userated within 1 start Interfloop/Winkins Tech Radar all 5 years and rating userated within 1 start Interfloop/Winkins Tech Radar all 5 years and rating userated within 1 start Interfloop/Winkins Tech Radar all 5 years and rating userated within 1 start Interfloop/Winkins Tech Radar all 5 years Interfloop/Winkins Tech Radar all 5 years and radar all years Interfloop/Winkins Tech Radar all 5 years	5	Code-base Size	Large.	Large.	•
Lots of blog posts, turorials and SlockOverflow posts. In the ThoughtWorks Tech Radar at Siverary appeared only once. In the ThoughtWorks Tech Radar at Siverary appeared only once. In the ThoughtWorks Tech Radar at Siverary appeared only once. In the ThoughtWorks Tech Radar at Siverary appeared only once. In the ThoughtWorks Tech Radar at Siverary appeared only once. In the ThoughtWorks Tech Radar within just 2 years and rating superaded until an appeared only once. In the ThoughtWorks Tech Radar within just 2 years and rating superaded until an appeared only once in the technique of the second on the second in appeared only once in the second once in the second only once in the second			46,000 lines of code (source: OpenHub)	50,000 lines of code (source: OpenHub)	
Lots of blog posts, tutorials and Stack-Overflow posts. In the Thought-Works Tech Radar at 5 years, appeared only once. 7 JanoiPath Implementation 8 Affective and provided and provided and state of the provided and provided	6	Mature	Yes. Inception 2010.	Yes. Inception February 2017.	
## In the Thought Works fech Radar at 5 years appeared only once. ## Japared only once. #			9, .	Clear signs of wide and rapid adoption.	than 10 years for
Registrollar March				just 2 years and rating upgraded within 1	show up on <u>Stack Overflow</u>
Start Salo supported, but paths that return XML notes cannot be used in assertions. Updating an XML document is not possible. Apache 4X, but the code depends on dependencies Apache 4X, but the code depends on the Apache ibtrares Devendencies Apache 4X, but the code of the c	7		Groovy GPath	JayWay JsonPath	
Start Salo supported, but paths that return XML notes cannot be used in assertions. Updating an XML document is not possible. Apache 4X, but the code depends on dependencies Apache 4X, but the code depends on the Apache ibtrares Devendencies Apache 4X, but the code of the c	8	XPath implementation	Groovy GPath and "XMLSlurper". Standard	W3C standard XPath using the Java	
December 2015 There are some concerns with this design, More details in this issue. More details in this issue. No Ves (Maren Archetype). Dev onboarding experience much better with Krate. Archetype includes working experience much experienc		, ,	XPath is also supported, but paths that return XML nodes cannot be used in assertions. Updating an XML document is	built-in XML lib. You can even update	
There are some concerns with this design. More details in this issue. No Quick Start / Project Scaffolding Test Scripting Language Test Scripting Language Test Scripts have to be compiled No Test Scripting Language Test Scripts have to be compiled No Test Scripts have to be compiled in version 1.0 onwards. Script have pretty good syntax coloring. Karate has a Java API option in version 1.0 onwards. Even hetter I Debug in Visual Studio Cond with a tep-back and even hot-reload (since v0.9.5). Also see (42) Code with a tep-back and even hot-reload (since v0.9.5). Also see the Dev. mode HTML reports steps, error diagnostics and HTTP logs in the law to the compile, and there is a Java API. Test Runner Any, bring your own. TestNG or JUnit will work. No (have to use TestNG or equivalent) No (have to us	9	HTTP Client	· · · · · · · · · · · · · · · · · · ·		
More details in this issue. are "shaded" to not cause conflicts.			There are some concerns with this design.		because of this abstraction.
Outsk Start / Project Scaffolding			More details in this issue.		
There's also a standalone executable and zarchetype includes working example. Partial	10	Quick Start / Project			Dev onboarding experience
Test Scripting Language Lan		Scaffolding			much better with Karate.
Test Scripting Language					
Language Language Language Language Language Language Language No Tests or plain-text. No IDE required for Karate Tests are plain-text. No IDE required for Karate Tests are plain-text. No IDE required for Karate Since you can re-use ISON purport Cucumber plugins that work well and have pretty good syntax coloring. Karate has a Java API option in version 1.0 onwards. Even better I Debug in Visual Studio Code with step-back and even hot-reload (since v0.9.5). also see the Devenode HTML report: steps, error diagnostics and HTTP logs in-line, and there is a Java API. Junit is supported and can co-exist with TestNG in the same project if needed. Test Runner Any, bring your own. TestNG or JUnit will Junit is supported and can co-exist with TestNG in the same project if needed. Testend with custom routines via Test Runner Any Chrave to use TestNG or equivalent) No (have to use TestNG or equivalent) No way to do a "deep equals" comparison for nested objects. No way to "grore" fields - for e.g. id / date / time values which are dynamic No way to "grore" fields - for e.g. id / date / time values which are dynamic No way to "grore" fields - for e.g. id / date / time values which are dynamic Ves includes RegEx validations No No No No No No No No No N				ZIP Release.	example.
Legislation	11	Test-Scripting	Java	Karate-Script (Cucumber / Gherkin)	No Java knowledge needed for
12 Test Scripts have to be compiled 13 IDE Support 14 Partial. Eclipse and Intellid have Cucumber plugins that work well and have pretty good syntax coloring. Karate has a Java API option in version 1.0 onwards. 14 Step Through / Debug-ability 15 Test Runner 16 Tags / Groups Built in 17 Extend with custom rorutines via 18 Re-use Java code 19 Validate AII Payloady values in one step 19 Ratial actions and values in one step 20 Built-in data-type, conditional-logic and RegEx validations 21 Validate schema of all elements in a JSON 21 Validate schema of all elements in a JSON 21 Validate schema of all elements in a JSON 21 Validate schema of all elements in a JSON 21 Validate schema of all elements in a JSON 21 Validate schema of all elements in a JSON 21 Validate schema of all elements in a JSON 21 Validate schema of all elements in a JSON 21 Validate schema of all elements in a JSON 21 Validate schema of all elements in a JSON 21 Validate schema of all elements in a JSON 22 See, Intellid Sepace Auto-Complete and RegEx validations 24 Complete and Intellid have Cucumber plugins that work well and have pretty good syntax coloring. 25 Read of Cucumber plugins that work well and have pretty good syntax coloring. 26 Partial Eclipse and Intellid have 26 Cucumber plugins that work well and have pretty good syntax coloring. 26 Exen plunit and pall execution Code with step-back and even hot reload (since v0.9.5). 26 Even better I Debug in Visual Studio Code with step-back and even hot reload (since v0.9.5). 26 Even better I Debug in Visual Studio Code with step-back and even hot reload (since v0.9.5). 27 Even better I Debug in Visual Studio Code with fire-re-re-re-re-re-re-re-re-re-re-re-re-r		Language			Karate
Test Scripts have to be compiled Yes No Tests are plain-text. No IDE required for Karate				Java is also supported in version 1.0	
Compiled Pequive for Karate				onwards.	
Test Runner Any, bring your own. TestNG or JUnit will work. Step Through / Debug-ability Past Runner Any, bring your own. TestNG or JUnit will work. Step Junit is optional vort. Step Junit is optional vo	12		Yes	No	·
Refactoring work for Java and POJO-s Refactoring work for Java API poption in version 1.0 onwards. Refactoring work for Java and POJO-s Refactoring work for Java API poption in version 1.0 onwards. Refactoring work for Java API poption in version 1.0 onwards. Refactoring work for Java API poption in version 1.0 onwards. Refactoring work for Java API poption in version 1.0 onwards. Refactoring work for Java API poption in version 1.0 onwards. Refactoring work for Java API poption in version 1.0 onwards. Refactoring work for Java API poption in version 1.0 onwards. Revue Java API poption in version 1.1 onwards. Also see (42) Also see (42) Also see (42) And Karate pate well. And Karate's parallel execution capability is in "core", independent of even Maven or any unit-testing framework. Yes 15 Test Runner No leave to use TestNG or JUnit will work. No leave to use TestNG or JUnit will work. Pes Java API poption in version 1.0 onwards. Also see (42) And Karate's parallel execution capability is in "core", independent of even Maven or any unit-testing framework. Pes Juliate Interest on Juliate Season of a sale of even Maven or any unit-testing framework. Pes Juliate Interest on Juliate Season of even Juliate Season or any unit-testing framework. Pes Juliate Season or Season or any unit-testing framework.	12		Vos Intelli Sonso Auto Complete and	Partial Eclinea and Intellia have	
have pretty good syntax coloring. Karate has a Java API option in version 1.0 onwards. Even better! Debug in Visual Studio Code with step-back and even hot-reload (since v0.9.5). also see the Dev-mode HTML report: steps, error diagnostics and HTPT logs in-line, and there is a Java API. Test Runner Any, bring your own. TestNG or JUnit will work. Any, bring your own. TestNG or JUnit will work. TestNG in the same project if needed. Tags / Groups Built In No (have to use TestNG or equivalent) Re-use Java code Tags / Groups Built In No (have to use TestNG or equivalent) Re-use Java code Yes No need to compile, and read values in one step Ves (via JavaScript No need to compile, and routines via No way to 'group' fields - for e.g. id / date / time values which are dynamic No No way to 'group' fields - for e.g. id / date / time values which are dynamic Yes, includes RegEx and Macros No Ves, includes RegEx and Macros No Ves Ves Ves Ves Ves, includes RegEx and Macros No Ves Ves Ves Ves Ves Ves Ves, includes RegEx and Macros	13	IDL Support	· ·	•	
Karate has a java API option in version 1.0 onwards.			Relactoring work for Java and 1 030-3		
Karate has a Java API option in version 1.0 onwards.				liave pretty good syntax coloring.	
Step Through				Karata has a lava ADI ention in varsion	covered as well.
Step Through / Debug-ability Yes. Java + IDE Support. Code with step-back and even hot-reload (since v0.9.5). also see the Dev-mode HTML report: steps, error diagnostics and HTTP logs in-line, and there is a Java API.					
Debug-ability Code with step-back and even hot-reload (since v0.9.5). also see the Dev-mode HTML report: steps, error diagnostics and HTTP logs in-line, and there is a Java API. Test Runner Any, bring your own. TestNG or JUnit will work. Test Runner Any, bring your own. TestNG or JUnit will work. 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 No (have to use TestNG or equivalent) Yes Test Runner And Karate's parallel execution capability is in "core", independent of even Maven or any unit-testing framework. Peven JUnit is optional No need to compile, and easier for non-programmers. Yes (via JavaScript interop) No need to compile, and easier for non-programmers. Yes (via JavaScript interop) 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. No way to "ignore" fields - for e.g., id / date / time values which are dynamic Do atte / time values which are dynamic No Yes, includes RegEx and Macros Yes, includes RegEx and Macros	4.4	C: T! ! /	V 1 1050		41 (42)
also see the <u>Dev-mode HTML report:</u> steps, error diagnostics and HTTP logs in-line, and there is a Java API. Any, bring your own. TestNG or JUnit will work. 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) Fexen JUnit is optional No need to compile, and easier for non-programmers. Re-use Java code Ves Ves (via JavaScript interop) And Karate's parallel execution capability is in "core", independent of even Maven or any unit-testing framework. No need to compile, and easier for non-programmers. Re-use Java code Validate All Payload values in one step Ves (via JavaScript interop) No No easy way to do 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. No way to "ignore" fields - for e.g. id / date / time values which are dynamic No Built-in data-type, conditional-logic and RegEx validations Result in data-type, conditional-logic and RegEx validations No Yes, includes RegEx and Macros Yes, includes RegEx and Macros	14		Yes. Java + IDE Support.	_	Also see (42)
Steps, error diagnostics and HTTP logs in-line, and there is a Java API.				(since v0.9.5).	
Steps, error diagnostics and HTTP logs in-line, and there is a Java API.					
in-line, and there is a Java API. Any, bring your own. TestNG or JUnit will work. Any, bring your own. TestNG or JUnit will work. TestNG in the same project if needed. Tags / Groups Built In No (have to use TestNG or equivalent) Pes Textend with custom routines via Re-use Java code Validate All Payload values in one step Not built-in. You need to use external libraries such as Hamcrest. No way to "ignore" fields - for e.g. id / date / time values which are dynamic No Way to "ignore" fields - for e.g. id / date / time values which are dynamic No Validate Schema of all elements in a JSON In-line, and there is a Java API. Junit is supported and can co-exist with TestNG in the same project if needed. Junit is supported and can co-exist with TestNG in the same project if needed. Junit is supported and can co-exist with TestNG in the same project if needed. JavaScript No need to compile, and easier for non-programmers. 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. No way to "ignore" fields - for e.g. id / date / time values which are dynamic No Yes, includes RegEx and Macros Yes, includes RegEx and Macros					
Test Runner Any, bring your own. TestNG or JUnit will work. Any, bring your own. TestNG or JUnit will work. Item JUnit is supported and can co-exist with TestNG in the same project if needed. Even JUnit is optional Yes Tags / Groups Built In No (have to use TestNG or equivalent) Pess Item Junit is optional Yes Item Junit is optional No need to compile, and easier for non-programmers. Pess Java code Yes Validate All Payload values in one step 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 District in data-type, conditional-logic and RegEx validations No Validate schema of all elements in a JSON No No No No No Yes Junit is supported and can co-exist with TestNG in the same project if needed. Even JUnit is optional Yes No need to compile, and easier for non-programmers. No need to compile, and easier for non-programmers. No need to compile, and easier for non-programmers. 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. Example 1 Example 1 Example 2 Example 3 Example 4					
TestNG in the same project if needed. Tags / Groups Built In No (have to use TestNG or equivalent) No (have to use TestNG or equivalent) TestNG in the same project if needed. Even JUnit is optional No (have to use TestNG or equivalent) Yes TestNG in the same project if needed. Even JUnit is optional No need to compile, and easier for non-programmers. No need to compile, and easier for non-programmers. No need to compile, and easier for non-programmers. Re-use Java code Yes Not built-in. You need to use external libraries such as Hamcrest. No easy way to do 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. No way to "ignore" fields - for e.g. id / date / time values which are dynamic No Yes, includes RegEx and Macros Yes, includes RegEx and Macros Yes, includes RegEx and Macros				·	
Tags / Groups Built In No (have to use TestNG or equivalent) Yes	15	Test Runner			
Even JUnit is optional any unit-testing framework. Tags / Groups Built In No (have to use TestNG or equivalent) Extend with custom routines via Re-use Java code Ves No need to compile, and easier for non-programmers. Yes (via JavaScript interop) 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 No Built-in data-type, conditional-logic and RegEx validations Validate schema of all elements in a JSON Even JUnit is optional any unit-testing framework. Yes Valva Script No need to compile, and easier for non-programmers. 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. Example 1 Example 2 Example 2 Example 3 Example 4 Yes, includes RegEx and Macros			WOTA.	lesavo in the same project ii needed.	
Tags / Groups Built In No (have to use TestNG or equivalent) Extend with custom routines via Re-use Java code Yes 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 Built-in data-type, conditional-logic and RegEx validations Validate schema of all elements in a JSON No (have to use TestNG or equivalent) Yes Yes Yes Yes (via JavaScript interop) 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. Example 1 Example 1 Example 2 Example 3 Example 4				Even II Init is optional	
17 Extend with custom routines via 18 Re-use Java code 19 Validate All Payload values in one step 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 Ves, includes RegEx and Macros Ves (via JavaScript interop) No easy way to do 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. Example 1 Example 2	16	Tags / Groups Built In	No (have to use TestNG or equivalent)		any unit-testing namework.
routines via Re-use Java code Yes 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 Built-in data-type, conditional-logic and RegEx validations Validate schema of all elements in a JSON Reservance Yes (via JavaScript interop) 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. Yes, includes RegEx and Macros Yes Yes Yes Yes Yes		20-7 -10 apo 2 une 111	, and the second of equivalently		
18 Re-use Java code 19 Validate All Payload values in one step 19 Validate All Payload values in one step 20 Built-in data-type, conditional-logic and RegEx validations 21 Validate Schema of all elements in a JSON 2 Validate All Payload values in one step 2 Validate All Payload values in one step 2 Validate All Payload values in one step 3 Not built-in. You need to use external libraries such as Hamcrest. 4 Validate schema of all elements in a JSON 4 Validate Schema of all elements in a JSON 4 Validate schema of all elements in a JSON 4 Validate schema of all elements in a JSON 4 Validate schema of all elements in a JSON 4 Validate schema of all elements in a JSON 4 Validate schema of all elements in a JSON 4 Validate schema of all elements in a JSON 4 Validate schema of all elements in a JSON 5 Validat	17		Java code	JavaScript	
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 20 Built-in data-type, conditional-logic and RegEx validations 21 Validate schema of all elements in a JSON No built-in. You need to use external libraries such as Hamcrest. 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 GraphOL. Yes, includes RegEx and Macros Yes, includes RegEx and Macros		routines via			easier for non-programmers.
values in one step Values in one order" assertion for JSON, JSON arrays and XML, and lets you ignore chosen fields at the same time. Values in one step Values in one order" assertion for JSON, JSON arrays and XML, and lets you ignore chosen fields at the same time. Values in one order" assertion for JSON, JSON arrays and XML, and lets you ignore chosen fields at the same time. Values in one order" assertion for JSON, JSON arrays and XML, and lets you ignore chosen fields at the same time. Values in one order in assertion for JSON, JSON arrays and XML, and lets you ignore chosen fields at the same time. Values in one order in assertion for JSON, JSON arrays and XML, and lets you ignore chosen fields at the same time. Values in one order in assertion for JSON, JSON arrays and XML, and lets you ignore chosen fields at the same time. Values in one order in assertion for GraphOL. Ves, includes RegEx and Macros Ves, includes RegEx and Macros Ves, includes RegEx and Macros Ves, includes in assertion for GraphOL.	18	Re-use Java code	Yes	Yes (via <u>JavaScript interop</u>)	
values in one step Values in one order" assertion for JSON, JSON arrays and XML, and lets you ignore chosen fields at the same time. Values in one step Values in one order" assertion for JSON, JSON arrays and XML, and lets you ignore chosen fields at the same time. Values in one order" assertion for JSON, JSON arrays and XML, and lets you ignore chosen fields at the same time. Values in one order" assertion for JSON, JSON arrays and XML, and lets you ignore chosen fields at the same time. Values in one order in assertion for JSON, JSON arrays and XML, and lets you ignore chosen fields at the same time. Values in one order in assertion for JSON, JSON arrays and XML, and lets you ignore chosen fields at the same time. Values in one order in assertion for JSON, JSON arrays and XML, and lets you ignore chosen fields at the same time. Values in one order in assertion for GraphOL. Ves, includes RegEx and Macros Ves, includes RegEx and Macros Ves, includes RegEx and Macros Ves, includes in assertion for GraphOL.	19	Validate All Payload	Not built-in. You <i>need</i> to use external	Karate natively supports a "deep equals"	IMO the biggest limitation of
Son, JSON arrays and XML, and lets you ignore chosen fields at the same time. 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 Built-in data-type, conditional-logic and RegEx validations Validate schema of all elements in a JSON No No No No No Yes JSON, JSON arrays and XML, and lets you ignore chosen fields at the same time. Example 1 Example 2 Example 2 Example 4 Yes, includes RegEx and Macros		•			
 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 Built-in data-type, conditional-logic and RegEx validations Validate schema of all elements in a JSON No easy way to do a "deep equals" ignore chosen fields at the same time. Example 2 Example 3 Example 4 This is super-important for GraphOL. Yes, includes RegEx and Macros Yes 				•	
comparison for nested objects. No way to "ignore" fields - for e.g. id / date / time values which are dynamic Built-in data-type, conditional-logic and RegEx validations Validate schema of all elements in a JSON Comparison for nested objects. This is super-important for GraphOL. Yes, includes RegEx and Macros Yes, includes RegEx and Macros			No easy way to do a "deep equals"		
No way to "ignore" fields - for e.g. id / date / time values which are dynamic Built-in data-type, conditional-logic and RegEx validations Validate schema of all elements in a JSON No way to "ignore" fields - for e.g. id / date / time values which are dynamic Yes, includes RegEx and Macros Yes, includes RegEx and Macros					
20 Built-in data-type, conditional-logic and RegEx validations 21 Validate schema of all elements in a JSON date / time values which are dynamic Yes, includes RegEx and Macros Yes, includes RegEx and Macros Yes, includes RegEx and Macros			·	This is super-important for GraphQL.	
Built-in data-type, conditional-logic and RegEx validations Validate schema of all elements in a JSON No Yes, includes RegEx and Macros Yes, includes RegEx and Macros				Side in the second seco	
conditional-logic and RegEx validations 21 Validate schema of all elements in a JSON Ves	20	Built-in data-type		Yes, includes RegEx and Macros	
RegEx validations 21 Validate schema of all elements in a JSON No Yes					
21 Validate schema of all elements in a JSON No Yes		_			
elements in a JSON	21		No	Ves	
	-1			100	
array in one step					
		urray iii Olie step			

	I			I
22	Built-in JSON Schema	Yes	RegEx and Macros support is sufficient	For details on how Karate's
	and XML Schema		(and far simpler) for most use cases. That	approach is simpler and more
	validation support		said, users can easily add a Java lib via	intuitive than JSON (or XML)
			Karate's Java interop - if needed.	Schema see this link.
22	Native superant for	Ne	·	No need to use double-quotes
23	Native support for	No	Yes	
	expressing JSON or			or "escape" characters.
	XML in test-scripts	"{ \"name\": \"Billie\" }"	{ name: 'Billie' }	
		" <cat name='\"Billie\"'></cat> "	<cat name="Billie"></cat>	You can also read from files
		0		and re-use.
24	Example – JSON	@Test public void lotto resource returns 200 with expecte	Scenario: lotto resource returns 200 with expected id and winners	Matching built-in, and more
	assertions	d id and winners() {	with expected id and winners	readable syntax. Note the
		<u> </u>	Given path 'lotto', 5	simpler way to specify path
		when ().	When method get	parameters without
		get("/lotto/{id}", 5).	Then status 200	placeholders.
		then().	And match \$.lotto.lottoId == 5	placefloiders.
		statusCode(200).	And match \$.lotto.winners[*].winnerId	
		<pre>body("lotto.lottoId", equalTo(5),</pre>	contains only [23, 54]	For REST-assured, IDE
		"lotto.winners.winnerId",		formatting is a known
		containsOnly(23, 54));		challenge.
		}		
25	Example - GET with	given().	Given param key1 = 'value1'	Karate is a <u>true DSL</u> . No syntax
	params	<pre>param("key1", "value1"). param("key2", "value2").</pre>	And param key2 = 'value2' And path 'somewhere'	"noise", no unnecessary
		when().	When method get	symbols or punctuation. No
		get("/somewhere").	Then match response contains 'OK'	need to worry about indenting
		then().	material response consulting the	, · · · · · · · · · · · · · · · · · · ·
		<pre>body(containsString("OK"));</pre>		a giant "one liner" of Java
				code.
26	Extracting multiple	Convoluted.	Easy. You can even use JsonPath to	Some of the <u>quirks</u> of the
	data-elements for		extract JSON chunks or arrays and save	REST-assured JsonPath
	reuse in subsequent	The Fluent Interface which is supposed to	them to variables for use in later steps.	implementation get in the way
	HTTP calls	be the main highlight of REST-Assured	The state of the s	as well.
	HTTP Calls	5 5	5 70 11 70 71 71	as well.
		actually gets in the way here. More	For XML, XPath does the same.	1
		<u>examples.</u>		
27	Can update a given	No.	Yes. There are actually multiple ways to	You can even modify a
	JSON or XML using a		update payloads: a) by path b) using	response and re-use it 'as-is'
	path expression	Especially for data-driven tests, updating	embedded expressions and c) via a	as the next request.
	patri expression		·	as the flext request.
		nested JSON is <u>near impossible</u> .	built-in string replacement keyword.	
28	Data Driven Testing	No (have to use TestNG or equivalent)	Yes. Can even use dynamic JSON or even	
			CSV as a data-source.	
		REST-Assured Example	Karate Example	
29	SOAP support	No	Yes	Plus, Karate's XML support is
23	SOAI Support	NO	163	far more <u>flexible</u> and easier to
				use.
30	HTTPS / SSL without	Although there is " <u>relaxed</u> " HTTPS, a	Yes	
	certificates	certificate is <u>needed</u> in some cases		
31	Built-in support for	No	Yes.	In REST-assured, you have to
	switching			use something like a
	environment config	Also config is somewhat convoluted in	Adding a new variable to a test is just	dependency-injection
	environment comig			
		REST-Assured	one step: edit karate-config.js	framework (or roll your own)
				to read properties files.
32	File Upload /	Partial / Buggy	Yes	
	Multipart Support	<u>Libraries</u> Content-Type Dependencies		
		'multipart/related' not supported		I
		questions on 'multipart/mixed'		1
22	LIDI and a deduction to		. Van	<u> </u>
33	URL encoded HTML	Yes	Yes	1
	Form data			
34	Cookies	Some Limitations	Yes	Can be configured one-time
				for all subsequent requests,
				just like headers.
35	Auth Schemes out of	Yes	No (but <u>easily pluggable</u> via re-usable	
33		100		I
	the box		scripts or JavaScript without needing to	1
			write Java code)	
36	Custom Auth	Java code (needs compilation). Existing	Unified plug-in system via JavaScript (no	
		mechanism is <u>not extensible</u> .	compilation needed)	<u></u>
37		At a Mile the second transfer of the first transfer of	<u>Yes</u>	This is a critical requirement
3/	Parallel Execution of	No. While some teams seem to have had		•
37				I for HTTP integration tests
37	Parallel Execution of Tests	success running REST-assured in parallel,	Even if you run tosts in parallal reports	for HTTP integration tests
37		success running REST-assured in parallel, there are some cases in which	Even if you run tests in parallel, reports	which typically take a much
3/		success running REST-assured in parallel, there are <u>some cases</u> in which multi-threading is not supported. Also see	and logs are collected per test and HTML	_
37		success running REST-assured in parallel, there are some cases in which		which typically take a much
37		success running REST-assured in parallel, there are <u>some cases</u> in which multi-threading is not supported. Also see	and logs are collected per test and HTML	which typically take a much
3/		success running REST-assured in parallel, there are <u>some cases</u> in which multi-threading is not supported. Also see <u>this thread</u> . The creator has also confirmed that "REST Assured is not 100% safe to use	and logs are collected per test and HTML	which typically take a much
	Tests	success running REST-assured in parallel, there are <u>some cases</u> in which multi-threading is not supported. Also see <u>this thread</u> . The creator has also confirmed that "REST Assured is not 100% safe to use <u>in multi-threaded scenarios</u> ".	and logs are collected per test and HTML reporting works as you would expect.	which typically take a much
38	Tests Floating-point	success running REST-assured in parallel, there are some cases in which multi-threading is not supported. Also see this thread. The creator has also confirmed that "REST Assured is not 100% safe to use in multi-threaded scenarios". All numbers are converted to float and you	and logs are collected per test and HTML reporting works as you would expect. Numeric assertions work just as you	which typically take a much longer time than unit tests.
	Tests	success running REST-assured in parallel, there are some cases in which multi-threading is not supported. Also see this thread. The creator has also confirmed that "REST Assured is not 100% safe to use in multi-threaded scenarios". All numbers are converted to float and you shouldn't forget to use floats (not the	and logs are collected per test and HTML reporting works as you would expect. Numeric assertions work just as you expect and even auto-conversion to	which typically take a much
	Tests Floating-point	success running REST-assured in parallel, there are some cases in which multi-threading is not supported. Also see this thread. The creator has also confirmed that "REST Assured is not 100% safe to use in multi-threaded scenarios". All numbers are converted to float and you	and logs are collected per test and HTML reporting works as you would expect. Numeric assertions work just as you	which typically take a much longer time than unit tests. Even this works:
	Tests Floating-point	success running REST-assured in parallel, there are some cases in which multi-threading is not supported. Also see this thread. The creator has also confirmed that "REST Assured is not 100% safe to use in multi-threaded scenarios". All numbers are converted to float and you shouldn't forget to use floats (not the	and logs are collected per test and HTML reporting works as you would expect. Numeric assertions work just as you expect and even auto-conversion to BigDecimal happens if needed.	which typically take a much longer time than unit tests. Even this works: And \$.odd.ck ==
	Tests Floating-point	success running REST-assured in parallel, there are some cases in which multi-threading is not supported. Also see this thread. The creator has also confirmed that "REST Assured is not 100% safe to use in multi-threaded scenarios". All numbers are converted to float and you shouldn't forget to use floats (not the	and logs are collected per test and HTML reporting works as you would expect. Numeric assertions work just as you expect and even auto-conversion to BigDecimal happens if needed. Given path 'odd'	which typically take a much longer time than unit tests. Even this works:
	Tests Floating-point	success running REST-assured in parallel, there are some cases in which multi-threading is not supported. Also see this thread. The creator has also confirmed that "REST Assured is not 100% safe to use in multi-threaded scenarios". All numbers are converted to float and you shouldn't forget to use floats (not the default double) in assertions. get ("/odd") .then().assertThat()	and logs are collected per test and HTML reporting works as you would expect. Numeric assertions work just as you expect and even auto-conversion to BigDecimal happens if needed. Given path 'odd' when method get	which typically take a much longer time than unit tests. Even this works: And \$.odd.ck ==
38	Floating-point precision	success running REST-assured in parallel, there are some cases in which multi-threading is not supported. Also see this thread. The creator has also confirmed that "REST Assured is not 100% safe to use in multi-threaded scenarios". All numbers are converted to float and you shouldn't forget to use floats (not the default double) in assertions. get ("/odd") .then().assertThat() .body("odd.ck", equalTo(12.2f));	and logs are collected per test and HTML reporting works as you would expect. Numeric assertions work just as you expect and even auto-conversion to BigDecimal happens if needed. Given path 'odd'	which typically take a much longer time than unit tests. Even this works: And \$.odd.ck == 12.200000000000000000000000000000000000
	Tests Floating-point	success running REST-assured in parallel, there are some cases in which multi-threading is not supported. Also see this thread. The creator has also confirmed that "REST Assured is not 100% safe to use in multi-threaded scenarios". All numbers are converted to float and you shouldn't forget to use floats (not the default double) in assertions. get ("/odd") .then().assertThat()	and logs are collected per test and HTML reporting works as you would expect. Numeric assertions work just as you expect and even auto-conversion to BigDecimal happens if needed. Given path 'odd' when method get	which typically take a much longer time than unit tests. Even this works: And \$.odd.ck ==

Part		ı			,
Report includes HTP request and response logs in line Section			representations of payloads - you need	This particular comparison shows a	contortions you need to do to
Test any Java service			more Java code in place.		•
East any Java service to History and State (1997) History and State (1997) History and History a					
Act Construct ISON or No. Seate of plants No	40	Test Reports Built-in		•	,
A			equivalent for test reporting.		
Test any Java servict REST-assured has support for without a container without his pass and coded in list im made. There is no support for things like JAX-RS or custom services or controllers - and fort the test services or controllers - and for the execution points for teams to write an adapter. The huge advantage of Korate's approach is that the same test-script can be resued for http-integration tests without changes. Account Acco				,	1 5 7 5
without a container without and without a container without and without a container without the same test-script can be re-used for hitp-integration tests without changes. Wou the separation is sail the report a well within it parallel. Warate 0.7.0 onwards has support for container and with a carnfully simulate stateful CRUD. Having both a client and server in the same unified framework keeps things simple and you can more fast. Wou can re-use Karate tests as Galling performance tests you can compose multiple Karate feature files or "work-flows" into a single performance tests you can compose multiple Karate featur					
without a container without a container hard-coded in this mode. There is no support for things like JAX-RS or custom services or controllers - and for these you have to deploy to an app-server. The huge advantage of Karate's approach is that the same test-script can be required the support of these you have to deploy to an app-server. Ratate includes HTIP request and response logs in-line Ratate includes HTIP request and response logs in-line loss in the report as well, which is great for troubleshooting. All this works even when tests are run in parallel. Best explained via some examples. Ratate includes HTIP request and response logs in-line SDN report output. If you use the principle sponse in the PEON Report output, in similar way. Ratate includes HTIP request and response logs in-line SDN report output, in similar way. Ratate includes HTIP request and response logs in-line SDN report output, in similar way. Ratate includes HTIP request and response logs in the SDN report as without changes. Ratate includes HTIP request and response logs in the SDN report output, in similar way. Ratate 0.7.0 onwards has support for requested the same unfield framework keeps things simple and you can response multiple Karate feature files or "work-flows" into a single performance-test and use Galling to define the load model (ramps up.) Ratate 0.7.0 onwards has support for requested feature files or "work-flows" into a single representation of the documental single may be support for well-flowers in the spent and use Galling to define the load model (ramps up.) Ratate 0.9.0 onwards has support for work-flows and even generic async / work flows in the spent and promote single has support for work-flows and even generic async / work flo	41	Test any Java servlet	· ·		· ·
## decounteration for more details. There is no support for things like JAX. R5 or used to the search of these you have to deploy to an app-server. ## decounteration for more details. ## decounteration for more details. ## details. ## details. ## decounteration for more details. ## details. ## details. ## decounteration for more details. ## det		or HTTP resource	"out-of-container" testing of specifically	testing any servlet by providing	
there is no support for things like JAX-R5 or custom servlets or controllers - and for these you have to deploy to an app-server. ### Report includes HTTP request and response logs in-line ### Report and response logs in-line ### Report and response logs in-line ### No. ### Construct ISON or Will, from scratch using just path expressions ### No. ### Responsions or Mocks built-in ### No. ### No. ### Responsions or Mocks built-in ### No. ### No. ### Responsions or Mocks built-in ### No. ### No		without a container		extension points for teams to write an	
There is no support for things like JAX-R5 or these you have to deploy to an app-server. The huge advantage of Karate's approach is custom servictes or controllers—and for or these you have to deploy to an app-server. Report includes HTTP request and response logs in-line No. Report includes HTTP request and response logs in-line No. Rarate includes HTTP request and response logs in-line No. Rarate includes HTTP request and response logs in-line No. Rarate includes HTTP request and response logs in-line No. Rarate includes HTTP request and response logs in-line or the report as well, which is great for troubleshooting. All this works even when tests are run in parallel. Rest explained via some examples. No. Wou have to use 3rd party frameworks such as Witemook			"hard-coded" in this mode.	adapter.	
custom services or controllers - and for these you have to deploy to an app-server.					details.
these you have to deploy to an app-server. re-used for http-integration tests without changes. Report includes HTTP request and response logs in the JSON report output. If you use the prains 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. Construct JSON or XML from scratch using just path expressions No. Sest explained via zome examples. Karate 0.7.0 onwards has support for reacting API intests that can fully simulate stateful CRUD. Having both a comparison: Also see [37] No. Karate 0.9.0 onwards has support for reacting API intests that can fully simulate stateful CRUD. Having both a claim and severe in the same unified framework keeps things simple and you. Also see [37] Nou can reuse Karate tests as Gailing to define the load-model (ramp-up, counterent users, etc.) Also see this perf benchmark. Any output the service of the properties and see a single performance tests, etc.) Also see this performance tests and use Gating to define the load-model (ramp-up, counterent users, etc.) Also see this performance tests and see and see and use Gating to define the load-model (ramp-up, counterent users, etc.) Also see this performance tests and see and use Gating to define the load-model (ramp-up, counterent users, etc.) Also see this performance tests and see and use Gating to define the load-model (ramp-up, counterent users, etc.) Also see this performance tests and see and use Gating to define the load-model (ramp-up, counterent users, etc.) Also see this performance tests and expression the service and test and see and the service and test and see and the service and test and test and					
### Report includes HTTP request and response logs in His ISON report output. If you use the pastant keyword, the console output appears in the FON report output. If you use the pastant keyword, the console output appears in the FON report output. If you use the pastant 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. ### Construct ISON or XML from scratch using just path expressions ### No. ### Performance Testing				·	
Async Support No. No. No. No. No. No. Rarate includes HTTP request and response logs in line No. No. No. No. No. Response logs in the ISON report output. If you use the parameter seventh, the tast are run in parallel. No. No. Rest explained via some examples. No. No			these you have to deploy to an app-server.	, ,	I
Report includes HTTP request and response logs in-line No. Rarate includes HTTP request and response logs in the particle spin in the pSON 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.				without changes.	· · ·
request and response logs in-line request and response logs in the ISON 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. Rest explained via some examples. No. Rest explained via some examples. No. Test Doubles or Mocks built-in Vou have to use 3rd party frameworks such as Wiremock Test Doubles or Mocks built-in Vou have to use 3rd party frameworks such as Wiremock No. Also see [37] No. Also see [37] No. Also see [37] No. Websockets support No. No. No. No. No. Also see [37] No. Also see [37] No. Also see [37] No. Rest explained via some examples. No. Also see this performance itests. You can re-use Karate tests as Gattling to define the load-model (ramp-up. concurrent users, etc) Arate and no other open-source test-automation framework has this option. This alone is reason to choose Karate and one of the companies					in a similar way.
International Construct ISON or XML from scratch using just path expressions No. Best explained via some examples.	42	Report includes HTTP	No.	Karate includes <u>HTTP request and</u>	
console output appears in the report as well, which is great for troubleshooting. All this works even when tests are run in parallel. 8est explained via some examples. 8est explained via substance vib to Wiremock feet explained via via via via substance via v		request and response			
Well, which is great for troubleshooting, All this works even when tests are run in parallel.		logs in-line		If you use the print keyword, the	
All this works even when tests are run in parallel. Construct ISON or XML from scratch using just path expressions All Test Doubles or Mocks built-in All Test Doubles or Mocks built-in Also see [37] Performance Testing Websockets support No. Also see [37] Also see [37] Websockets support No. you have to use Java code or a library. Async Support No. you have to use Java code or a library. Retry Support No. you have to use Java code or a library. Retry Support No. you have to use Java code or a library. Also see No. you have to use Java code or a library. Retry Support No. you have to use Java code or a library. No. Web-UI Automation No. Also see No. you have to use Java code or a library. In Karate 0.9.0 nowards has support for websockets and even generic a sync / await support (see below) Also see Hittory Inc. Karate 0.9.0 nowards has support for websockets and even generic a sync / await support (see below) Also see Hittory Inc. Also see Hitt				console output appears in the report as	
Async Support No, you have to use Java code or a library Async Support No, you have to use Java code or a library Retry Support No, you have to use Java code or a library Retry Support No, you have to use Java code or a library				well, which is great for troubleshooting.	
Async Support No. you have to use Java code or a library Async Support No. you have to use Java code or a library. Async Support No. you have to use Java code or a library. Async Support No. you have to use Java code or a library. No. you have to use Java code or a library. No. you have to use Java code or a library. No. you have to use Java code or a library. No. you have to use Java code or a library. No. you have to use Java code or a library. No. you have to use Java code or a library. No. you have to use Java code or a library. No. you have to use Java code or a library. No. Jin Karate 0.9.0 onwards has support for where you gonk specify the condition to be polled for. No. you have to use Java code or a library. No. Jin Karate 0.9.0 onwards has bill-tin where you gonk specify the condition to be polled for. No. Jin Karate 0.9.5 onwards has bill-tin support out of the box. No. Jin Karate 0.9.5 onwards has bill-tin to be polled for. No. Jin Karate 0.9.5 onwards there is built-in support out of the box. No. Jin Karate 0.9.5 onwards there is built-in support out of the box. No. Jin Karate 0.9.5 onwards, there is built-in support out of the box. No. Jin Karate 0.9.5 onwards, there is built-in support out of the box. No. Jin Karate 0.9.5 onwards, there is built-in support out of the box. No. Jin Karate 0.9.5 onwards, there is built-in support out of the box. No. Jin Karate 0.9.5 onwards, there is built-in support out of the box. No. Jin Karate 0.9.5 onwards, there is built-in support out of the box. No. Jin Karate 0.9.5 onwards, there is built-in support out of the box. No. Jin Karate 0.9.5 onwards, there is built-in support out of the box. No. Jin Karate 0.9.5 onwards, there is built-in support out of the box. No. Jin Karate 0.9.5 onwards, there is built-in support out of the box.				All this works even when tests are run in	
XML from scratch using just path expressions At Test Doubles or Mocks built-in You have to use 3rd party frameworks such as Wiremock As Wiremock As Wiremock As Wiremock As Wiremock As Performance Testing Also see [37] No. Also see [37]				parallel.	
Using just path expressions Using just path expressions No.	43	Construct JSON or	No.	Best explained via some examples.	
Expressions No. No		XML from scratch			
Test Doubles or Mocks built-in Vou have to use 3rd party frameworks such as Wiremock Vou have to use 3rd party frameworks such as Wiremock Such as Wiremock Vou have to use 3rd party frameworks such as Wiremock Such as Wiremock Vou can revue Karate tests as Satting performance Testing Also see [37] Vou can revue Karate tests as Satting performance tests. Vou can revue Karate tests as Satting performance tests. Vou can compose multiple Karate feature files or "work-flows" into a single performance-tests and use Gatling to define the load-model (ramp-up, concurrent users, easync / await support (see below) Karate 0.9.0 onwards has support for websockets and even generic async / await support (see below) Karate 0.9.0 onwards has support to websockets and even generic async / await support (see below) Karate 0.9.0 onwards has this built-in where you only specify the condition to be polled for. No, you have to use Java code or a library like Awaitility Ratate 0.9.0 onwards has this built-in where you only specify the condition to be polled for. In Karate 0.9.0 onwards has this built-in support out of the box. Chrome / CDP, WebDriver, Appium and Playwright support (see Subject to Subject the execution across multiple Chrome / CDP, WebDriver, Appium and Playwright support to spit a test-sucte and orchestrate the execution across multiple Chrome / CDP, WebDriver, Appium and Playwright support to spit a test-sucte and orchestrate the execution across multiple Chrome / CDP, WebDriver, Appium and Playwright support to spit a test-sucte and orchestrate the execution across multiple Chrome / CDP, WebDriver, Appium and Playwright support to spit a test-sucte and orchestrate the execution across multiple Chrome / CDP, WebDriver, Appium and Playwright support to spit a test-sucte and orchestrate the execution across multiple Chrome / CDP, WebDriver, Appium and Playwright support to spit a test-sucte and orchestrate the execution across multiple Chrome / CDP, WebDriver, Appium an		using just path			
built-in You have to use 3rd party frameworks such as Wiremock No. Performance Testing No. Also see [37] No. Also see [37] You can compose multiple Karate feature files or "workflows" into a single performance-tests and use Gatting to define the load-model (ramp-up, concurrent users, etc) Websockets support No, you have to use Java code or a library like Awaitility No, you have to use Java code or a library. Rarate 0.9.0 nowards has this built-in where you only specify the condition to be polled for. No, you have to use Java code or a library. No, you have to use Java code or a library. No, you have to use Java code or a library. No, you have to use Java code or a library. No, you have to use Java code or a library. No, you have to use Java code or a library. No, you have to use Java code or a library. No, you have to use Java code or a library. No, you have to use Java code or a library. No, you have to use Java code or a library. No, you have to use Java code or a library. No, you have to use Java code or a library. In Karate 0.9.0 nowards has this built-in where you only specify the condition to be polled for. No, you have to use Java code or a library. In Karate 0.9.0 you can read a CSV file support out of the box. Chrome / CDP, WebDriver, Appium and Playwright support to split a test-suite and orchestrate the execution across multiple		expressions			
Some and the comparison	44	Test Doubles or Mocks	No.	Karate 0.7.0 onwards has support for	Karate is a superior alternative
Also see this perf benchmark.		built-in		creating API mocks that can fully	to Wiremock, here's a
Also see this perf benchmark. You can re-use Karate tests as Gatling performance tests. 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) In 0.9.0 onwards you can test any Java code, not just HTTP Async Support No. Websockets support No. Websockets and even generic async / await support (see below) Also see https://www.persockets.and.even.generic.async / await support (see below) Also see https://www.persockets.and.even.generic.async.gen			You have to use 3rd party frameworks such	simulate stateful CRUD. Having both a	comparison.
Performance Testing			as <u>Wiremock</u>	client and server in the same unified	
Performance Testing No. You can re-use Karate tests as Gatling performance tests. 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) In 0.9.0 onwards you can test any Java code, not just HTTP				framework keeps things simple and you	Also see this perf benchmark.
Also see [37] Also s				can move fast.	
Also see [37] Also s	45	Performance Testing	No.	You can re-use Karate tests as Gatling	This alone is reason to choose
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) In 0.9.0 onwards you can test any. Java code, not just HTTP					Karate and no other
files or "work-flows" into a single performance-test and use Gatling to define the load-model (ramp-up, concurrent users, etc) 46 Websockets support No. Karate 0.9.0 onwards has support for websockets and even generic async / await support (see below) Async Support No, you have to use Java code or a library like Awaitility Rarate 0.9.0 has built-in support via a very elegant API. Karate 0.9.0 onwards has this built-in support via a very elegant API. Karate 0.9.0 onwards has this built-in where you only specify the condition to be polled for. Karate 0.9.0 you can read a CSV file and it will be auto-converted to JSON Web-UI Automation No. In Karate 0.9.5 onwards, web-UI automation is possible. In Karate 0.9.5 onwards, there is built-in support do supported. In Karate 0.9.5 onwards, there is built-in supported.			Also see [37]		open-source test-automation
performance-test and use Gatling to define the load-model (ramp-up, concurrent users, etc) Websockets support No. No. No. Async Support No, you have to use Java code or a library like Awaitility No, you have to write custom Java logic. No, you have to write custom Java logic. No, you have to use Java code or a library. No, you have to write custom Java logic. Web-UI Automation No. No. No. Distributed Testing No. No. No. Performance-test and use Gatling to define the load-model (ramp-up, concurrent users, etc) No, Java code, not just HTTP Nary Java code, not just HTTP Karate 0.9.0 onwards support for websockets and even generic async / await support (see below) Also see https://twitter.com/KarateDSL / status/141702353608281293 S In Karate 0.9.0 onwards has this built-in - where you only specify the condition to be polled for. In Karate 0.9.0 you can read a CSV file support out of the box. Chrome / CDP, WebDriver, Appium and Playwright support to split a test-suite and orchestrate the execution across multiple				You can compose multiple Karate feature	framework has this option.
define the load-model (ramp-up, concurrent users, etc) No. Karate 0.9.0 onwards has support for websockets and even generic async / await support (see below) Async Support No, you have to use Java code or a library like Awaitility Retry Support No, you have to write custom Java logic. Karate 0.9.0 onwards has this built-in support via a very elegant API. Karate 0.9.0 onwards has this built-in where you only specify the condition to be polled for. 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 No. In Karate 0.9.5 onwards Web-UI automation No. In Karate 0.9.5 onwards, there is built-in support out of the box. Chrome / CDP, WebDriver, Appium and Playwright support to split a test-suite and orchestrate the execution across multiple				files or "work-flows" into a single	
Concurrent users, etc) Websockets support No. Async Support No, you have to use Java code or a library like Awaitility Rarate 0.9.0 onwards has support for websockets and even generic async / await support (see below) Karate 0.9.0 has built-in support via a very elegant API. Karate 0.9.0 onwards has this built-in where you only specify the condition to be polled for. No, you have to use Java code or a library. In Karate 0.9.0 you can read a CSV file support out of the box. No. Web-UI Automation No. In Karate 0.9.5 onwards Web-UI automation is possible. In Karate 0.9.5 onwards, there is built-in support of the box. In Karate 0.9.5 onwards, there is built-in support to split a test-suite and orchestrate the execution across multiple				performance-test and use Gatling to	In 0.9.0 onwards you can test
Websockets support No. Karate 0.9.0 onwards has support for websockets and even generic async / await support (see below)				define the load-model (ramp-up,	any Java code, not just HTTP
websockets and even generic async / await support (see below) Async Support No, you have to use Java code or a library like Awaitility Retry Support No, you have to write custom Java logic. Retry Support No, you have to write custom Java logic. Web-UI Automation No. No. In Karate 0.9.5 onwards web-UI automation is possible. No. In Karate 0.9.5 onwards web-UI automation is possible. In Karate 0.9.5 onwards, there is built-in supported. No. In Karate 0.9.5 onwards, there is built-in supported. No. In Karate 0.9.5 onwards, there is built-in supported.				concurrent users, etc)	
Async Support No, you have to use Java code or a library like Awaitility Async Support No, you have to use Java code or a library like Awaitility Async Support No, you have to write custom Java logic. Karate 0.9.0 navards has this built-in - where you only specify the condition to be polled for. 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 Support out of the box. In Karate 0.9.5 onwards Web-UI automation No. In Karate 0.9.5 onwards Web-UI automation Support out of the support out of the condition to be polled for. In Karate 0.9.5 onwards Web-UI automation Support out of the box or chestrate the execution across multiple No. In Karate 0.9.5 onwards, there is built-in support to split a test-suite and orchestrate the execution across multiple No.	46	Websockets support	No.	Karate 0.9.0 onwards has support for	
Async Support No, you have to use Java code or a library like Awaitility Retry Support No, you have to write custom Java logic. Retry Support No, you have to write custom Java logic. Retry Support No, you have to write custom Java logic. No, you have to write custom Java logic. Retry Support No, you have to write custom Java logic. In Karate 0.9.0 onwards has this built-in where you only specify the condition to be polled for. In Karate 0.9.0 you can read a CSV file support out of the box. No. No. In Karate 0.9.5 onwards Web-Ul automation is possible. No. In Karate 0.9.5 onwards, there is built-in support via a very elegant API. Karate also has YAML file support out of the box. Chrome / CDP, WebDriver, Appium and Playwright supported.		· · ·	_		
Async Support No, you have to use Java code or a library like Awaitility Retry Support No, you have to write custom Java logic. No, you have to write custom Java logic. No, you have to write custom Java logic. No, you have to use Java code or a library. No, you have to use Java code or a library. In Karate 0.9.0 onwards has this built-in - where you only specify the condition to be polled for. No, you have to use Java code or a library. In Karate 0.9.0 you can read a CSV file support out of the box. No. No. In Karate 0.9.5 onwards Web-UI automation is possible. Distributed Testing No. In Karate 0.9.5 onwards, there is built-in support via a very elegant API. Also see https://twitter.com/KarateDSI //status/141702353608281293 Salva Salv					
Ilike Awaitility very elegant API. https://twitter.com/KarateDSI /status/141702353608281293	47	Async Support	No. you have to use Java code or a library		Also see
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 to be polled for. 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 Web-UI Automation No. In Karate 0.9.5 onwards Web-UI automation is possible. Chrome / CDP, WebDriver, Appium and Playwright support do split a test-suite and orchestrate the execution across multiple		.,,		• •	1
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 to be polled for. Vo, 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 Web-UI Automation No. In Karate 0.9.5 onwards Web-UI automation is possible. Chrome / CDP, WebDriver, Appium and Playwright support do split a test-suite and orchestrate the execution across multiple					
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 to be polled for. 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 No. No. In Karate 0.9.5 onwards Web-UI automation is possible. Chrome / CDP, WebDriver, Appium and Playwright support do split a test-suite and orchestrate the execution across multiple					
where you only specify the condition 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 No. In Karate 0.9.5 onwards Web-UI automation No. In Karate 0.9.5 onwards Web-UI automation is possible. Chrome / CDP, WebDriver, Appium and Playwright supported. In Karate 0.9.5 onwards, there is built-in support to split a test-suite and orchestrate the execution across multiple	48	Retry Support	No. you have to write custom Java logic	Karate 0.9.0 onwards has this built-in -	_
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 support out of the box. No. In Karate 0.9.5 onwards Web-UI automation No. In Karate 0.9.5 onwards Web-UI automation is possible. Chrome / CDP, WebDriver, Appium and Playwright support do split a test-suite and orchestrate the execution across multiple		, -spport	, , ,		
49 CSV file support No, you have to use Java code or a library. 10 Karate 0.9.0 you can read a CSV file and it will be auto-converted to JSON No. 10 Karate 0.9.5 onwards Web-UI automation No. 10 Karate 0.9.5 onwards Web-UI automation is possible. 10 Distributed Testing No. 10 Karate 0.9.5 onwards, there is built-in support to split a test-suite and orchestrate the execution across multiple					
and it will be auto-converted to JSON support out of the box. No. In Karate 0.9.5 onwards Web-UI Appium and Playwright supported. Distributed Testing No. In Karate 0.9.5 onwards, there is built-in support to split a test-suite and orchestrate the execution across multiple	49	CSV file support	No you have to use lava code or a library		Karate also has VAMI file
Web-UI Automation No. In Karate 0.9.5 onwards Chrome / CDP, WebDriver, Appium and Playwright supported.	73	Cov me support	110, you have to use sava code of a library.		
automation is possible. Appium and Playwright supported. Distributed Testing No. In Karate 0.9.5 onwards, there is built-in support to split a test-suite and orchestrate the execution across multiple	50	Web-III Automation	No		
51 Distributed Testing No. In Karate 0.9.5 onwards, there is built-in support to split a test-suite and orchestrate the execution across multiple	30	Web-of Automation	INU.		
Distributed Testing No. In Karate 0.9.5 onwards, there is built-in support to split a test-suite and orchestrate the execution across multiple				automation is possible.	l ''' '
support to split a test-suite and orchestrate the execution across multiple	F1	Diotribute d Testing	No	In Koroto O O C amusuale the control of the	supporteu.
orchestrate the execution across multiple	21	Distributed lesting	INU.		
cloud or nardware / cluster nodes				·	
				cloud or nardware / cluster nodes	