Katalon Postman SoapUI Rest-Assured CITRUS Karate Immeter Apigee API Terminologies API Application Programming interface (API) is software that acts as an intermediary for two apps to communicate with each other. HTTP Hypertext Transfer Protocol is the collection of rules for the transmission of data on the World Wide Web, like graphic images, text, video, sound, and other multimedia HTTPS The S in HTTPS stands for "secure." HTTPS uses TLS (or SSL) to encrypt HTTP requests and responses URI Uniform Resource Identifier is a string identifier that refers to a resource on the internet. It is a string of characters that is used to identify any resource on the internet using location, name, or both. URL Uniform Resource Locator is used to find the location of the resource on the web. It is a reference for a resource and a way to access that resource. A URL always shows a unique resource, and it can be an HTML page, a CSS document, an image, etc. Layers of API Testing Three separate layers Presentation (or user interface) layer, the business layer, and the database layer for modeling and manipulating data. API Test Actions Verify correct HTTP status requests should return 403 FORBIDDEN, etc. Verify response payload Check valid JSON body and correct field names, types, and values — including in error responses. Verify correct application State Verify correct application This is optional and applies mainly to manual testing, or when a UI or another interface can be easily inspected. Verify testing ereformance sanity This is optional and applies mainly to manual testing, or when a UI or another interface can be easily inspected. API Test Scenario Categories 1 Basic positive tests (happy paths) Extended positive testing with optional parameters 3 Destructive testing with valid input API Example with Test Matrix API Call Action GET Jusers/(id) Get Jusers by UB Get all configurations for user	Top API Testing Tools	
CITRUS Karate Juneter apige		Doctman
API reminologies API meter apigee API meter apigee API meter apigee API province meter apigee API province meter apigee API province meters apigee API province managee API province meters managee API province meters manageee API province meters manageeeeeeeeeeee		
API Test Actions Presentation (or user interface) layer, the business layer, and the database layer for modeling and manipulating data. API Test Actions Verify correct HTTP status code Verify response payload Check valid JSON body and correct field names, types, and values — including in error responses Verify correct application This is optional and applies mainly to manual testing, or when a UI or another interface can be easily inspected. Verify correct application This is optional and applies mainly to manual testing, or when a UI or another interface can be easily inspected. Verify to performance Sacious API Test Scenario Categories This is optional and applies mainly to manual testing, or when a UI or another interface can be easily inspected. Verify test Scenario Categories 1 Sacious API Test Scenario Categories API Test Scenario Categories This is optional and applies mainly to manual testing, or when a UI or another interface can be easily inspected. Verify test Scenario Categories 4 Security, authorization, and permission tests (which are out of the scope of this post) Negative testing 4 Security and persisting with valid input API Est Scenario Categories 4 Security, authorization, and permission tests (which are out of the scope of this post) Negative testing with valid input API Est Scenario Categories 4 Security, authorization, and permission tests (which are out of the scope of this post) Negative testing with valid input API Est Scenario Categories 4 Security, authorization, and permission tests (which are out of the scope of this post) Negative testing with valid input API Est Scenario Categories 4 Security, authorization, and permission tests (which are out of the scope of this post) Negative testing with valid input API Est Scenario Categories 5 Negative testing with valid input API Est ample with Test Matrix API Test Scenario Categories GET /users /[id] Get user by ID Get all configurations for user Create a new configuration for user		
API Terminologies Application Programming Interface (API) is software that acts as an intermediary for two apps to communicate with each other. HTTP Hypertext Transfer Protocol is the collection of rules for the transmission of data on the World Wide Week, like graphic images, text, video, sound, and other multimedia and responses URI Uniform Resource Identifier is a string identifier that refers to a resource on the internet. It is a string of characters that is used to identify any resource on the internet using location, name, or both. URI Uniform Resource Locator is used to find the location of the resource on the web. It is a reference for a resource and a way to access that resource. A URL always shows a unique resource, and it can be an HTML page, a CSS document, an image, etc. Layers of API Testing Three separate layers Presentation (or user interface) layer, the business layer, and the database layer for modeling and manipulating data. API Test Actions Persentation (or user interface) layer, the business layer, and the database layer for modeling and manipulating data. API Test Actions Presentation (or user interface) layer, the business layer, and the database layer for modeling and manipulating data. API Test Actions Persentation (or user interface) layer, the business layer, and the database layer for modeling and manipulating data. API Test Actions For example, creating a resource should return 201 CREATED and unpermitted requests should return 403 FORBIDDEN, etc. Verify response payload Check valid JSON body and correct field names, types, and values — including in error responses. Verify correct application This is optional and applies mainly to manual testing, or when a UI or another interface can be easily inspected. Ner a security, audional and applies mainly to manual testing, or when a UI or another interface can be easily inspected. Ner a security, authorization, and permission tests (which are out of the scope of this post). Responses. Pestended positive testing w		
Application Programming interface (API) is software that acts as an intermediary for two apps to communicate with each other. HTTP Hypertext Transfer Protocol is the collection of rules for the transmission of data on the World Wide Web, like graphic images, text, video, sound, and other multimedia the World Wide Web, like graphic images, text, video, sound, and other multimedia the World Wide Web, like graphic images, text, video, sound, and other multimedia the World Wide Web, like graphic images, text, video, sound, and other multimedia the World Wide Web, like graphic images, text, video, sound, and other multimedia the World Wide Web, like graphic images, text, video, sound, and other multimedia the World Wide Web, like graphic images, text, video, sound, and other multimedia the World Wide Web, like graphic images, text, video, sound, and other multimedia the World Wide Web, like graphic images, text, video, sound, and other multimedia the World Wide Web, like graphic images, text, video, sound, and other multimedia the World Wide Web, like graphic images, text, video, sound, and other multimedia the World Wide Web, like graphic images, text, video, sound, and other multimedia the World Wide Web, like graphic images, text, video, sound, and other multimedia the World Wide Web, like graphic images, text, video, sound, and dreptimedia to a strip interface to a resource on the winterface on the interface on the interface of resource on the interface on the interface of resource and a way to access that resource on the interface on the web. It is a reference for a resource and a way to access that resource on the web. It is a reference for a resource and a way to access that resource. A URA always shows a unique resource and a way to access that resource. A URA always shows a unique resource and a way to access that resource. A URA always shows a unique resource and a way to access that resource. A URA always shows a unique resource and a way to access that resource and the database layer for modeling and ma		apigee
two apps to communicate with each other. Hypertext Transfer Protocol is the collection of rules for the transmission of data on the World Web, like graphic images, text, video, sound, and other multimedia The S in HTTPS stands for "secure." HTTPS uses TLS (or SSL) to encrypt HTTP requests and responses URI Uniform Resource Identifier is a string identifier that refers to a resource on the internet. It is a string of characters that is used to identify any resource on the internet using location, name, or both. URL Uniform Resource Locator is used to find the location of the resource on the web. It is a reference for a resource and a way to access that resource. A URL always shows a unique resource, and it can be an HTML page, a CSS document, an image, etc. Layers of API Testing Three separate layers Presentation (or user interface) layer, the business layer, and the database layer for modeling and manipulating data. API Test Actions Verify correct HTTP status code requests should return 403 FORBIDDEN, etc. Verify response payload Check valid JSON body and correct field names, types, and values — including in error responses. Verify response headers HTTP server headers have implications on both security and performance. Verify basic performance In case an operation was completed successfully but took an unreasonable amount of time, the test fails. API Test Scenario Categories 1 Basic positive tests (happy paths) 2 Extended positive testing with optional parameters 3 Destructive testing 4 Security, authorization, and permission tests (which are out of the scope of this post) 5 Negative testing with valid input API Example with Test Matrix API Call Action GET /users Get user by username GET /users/(id) Get user by ID Get all configurations for user Create a new configuration for user	API Terminologies	
HTTP Hypertext Transfer Protocol is the collection of rules for the transmission of data on the World Wide Web, like graphic images, text, video, sound, and other multimedia The Sin HTTPS stands for "secure." HTTPS uses TLS (or SSL) to encrypt HTTP requests and responses URI Uniform Resource Identifier" is a string identifier that refers to a resource on the internet. It is a string of characters that is used to identify any resource on the internet using location, name, or both. URI Uniform Resource Locator is used to find the location of the resource on the web. It is a reference for a resource and a way to access that resource. A URL always shows a unique resource, and it can be an HTML page, a CSS document, an image, etc. Layers of API Testing Three separate layers Presentation (or user interface) layer, the business layer, and the database layer for modeling and manipulating data. API Test Actions Verify correct HTTP status code requests should return 403 FORBIDDEN, etc. Verify response payload check valid JSON body and correct field names, types, and values — including in error responses. Verify response headers HTTP server headers have implications on both security and performance. Verify correct application This is optional and applies mainly to manual testing, or when a UI or another interface can be easily inspected. Verify basic performance in case an operation was completed successfully but took an unreasonable amount of time, the test fails. API Test Scenario Categories 1 Basic positive testing with optional parameters 2 Extended positive testing with optional parameters 3 Destructive testing 4 Security, authorization, and permission tests (which are out of the scope of this post) from the security and performance. API Example with Test Matrix API Call Action GET Jusers Get all configurations for user Jusers/(id)/configurations POST Create a new configuration for user	API	Application Programming Interface (API) is software that acts as an intermediary for
the World Wide Web, like graphic images, text, video, sound, and other multimedia HTTPS The S in HTTPS stands for "secure." HTTPS uses TLS (or SSL) to encrypt HTTP requests and responses URI Uniform Resource Identifier is a string identifier that refers to a resource on the internet. It is a string of characters that is used to identify any resource on the internet using location, name, or both. URL Uniform Resource Locator is used to find the location of the resource on the web. It is a reference for a resource and a way to access that resource. A URL always shows a unique resource, and it can be an HTML page, a CSS document, an image, etc. Layers of API Testing Three separate layers Presentation (or user interface) layer, the business layer, and the database layer for modeling and manipulating data. API Test Actions Verify correct HTTP status code Verify response payload Check valid JSON body and correct field names, types, and values — including in error responses. Verify correct application This is optional and applies mainly to manual testing, or when a UI or another interface can be easily inspected. Verify basic performance In case an operation was completed successfully but took an unreasonable amount of time, the test fails. API Test Scenario Categories 1 Basic positive tests (happy paths) 2 Extended positive testing with optional parameters 3 Destructive testing 4 Security, authorization, and permission tests (which are out of the scope of this post) Negative testing with valid input API Example with Test Matrix API Call Get user by ID Get Users / (id) Get user by ID Get users/(id) Get user by ID Get an ewe configurations for user		
The S in HTTPS stands for "secure." HTTPS uses TLS (or SSL) to encrypt HTTP requests and responses: URI Uniform Resource Identifier is a string identifier that refers to a resource on the internet. It is a string of characters that is used to identify any resource on the internet using location, name, or both. URI Uniform Resource Locator is used to find the location of the resource on the web. It is a reference for a resource and a way to access that resource. A URL always shows a unique resource, and it can be an HTML page, a CSS document, an image, etc. Layers of API Testing Three separate layers Presentation (or user interface) layer, the business layer, and the database layer for modeling and manipulating data. API Test Actions Verify correct HTTP status code Yerify response payload Check valid JSON body and correct field names, types, and values — including in error responses. Verify response headers HTTP server headers have implications on both security and performance. Verify correct application This is optional and applies mainly to manual testing, or when a UI or another interface can be easily inspected. In case an operation was completed successfully but took an unreasonable amount of time, the test fails. API Test Scenario Categories 1 Basic positive testing with optional parameters 2 Extended positive testing with optional parameters 3 Destructive testing 4 Security, authorization, and permission tests (which are out of the scope of this post) 5 Negative testing with valid input 6 Negative testing with valid input API Example with Test Matrix API Call GET Get user by ID Get all configurations for user POST Create a new configuration for user	HTTP	**
URI Uniform Resource Identifier is a string identifier that refers to a resource on the internet. It is a string of characters that is used to identify any resource on the internet using location, name, or both. URL Uniform Resource Locator is used to find the location of the resource on the web. It is a reference for a resource and a way to access that resource. A URL always shows a unique resource, and it can be an HTML page, a CSS document, an image, etc. Layers of API Testing Three separate layers Presentation (or user interface) layer, the business layer, and the database layer for modeling and manipulating data. API Test Actions Verify correct HTTP status for example, creating a resource should return 201 CREATED and unpermitted requests should return 403 FORBIDDEN, etc. Verify response payload Check valid JSON body and correct field names, types, and values — including in error responses. Verify response headers HTTP server headers have implications on both security and performance. Verify basic performance In its is optional and applies mainly to manual testing, or when a UI or another interface can be easily inspected. Verify basic performance In case an operation was completed successfully but took an unreasonable amount of time, the test fails. API Test Scenario Categories 1 Basic positive testing with optional parameters 2 Extended positive testing with optional parameters 3 Destructive testing 4 Security, authorization, and permission tests (which are out of the scope of this post) Negative testing with valid input API Example with Test Matrix API Call Get Jusers Users/(id) Get user by ID Get Get user by UB Get Jusers/(id) Get user by UB Get Jusers/(id) Create a new configuration for user		
URL URL Uniform Resource Identifier is a string identifier that refers to a resource on the internet. It is a string of characters that is used to identify any resource on the internet using location, name, or both. URL Uniform Resource Locator is used to find the location of the resource on the web. It is a reference for a resource and a way to access that resource. A URL always shows a unique resource, and it can be an HTML page, a CSS document, an image, etc. Layers of API Testing Three separate layers Presentation (or user interface) layer, the business layer, and the database layer for modeling and manipulating data. API Test Actions Verify correct HTTP status Code Perify correct HTTP status Code Verify response payload Check valid JSON body and correct field names, types, and values — including in error responses. Werify correct application State HTTP server headers have implications on both security and performance. Werify basic performance In case an operation was completed successfully but took an unreasonable amount of time, the test fails. API Test Scenario Categories 1 Basic positive testing with optional parameters 2 Extended positive testing with optional parameters 3 Destructive testing 4 Security, authorization, and permission tests (which are out of the scope of this post) 5 Negative testing with invalid input API Call API Call ACtion GET /users List all users GET GET /users/fidl Get user by ID GET /users/fidl Get user by ID GET /users/fidl Get user by ID GET /users/fidl/configurations POST Create a new configuration for user	HTTPS	
internet. It is a string of characters that is used to identify any resource on the internet using location, name, or both. URL Uniform Resource Locator is used to find the location of the resource on the web. It is a reference for a resource and a way to access that resource. A URL always shows a unique resource, and it can be an HTML page, a CSS document, an image, etc. Layers of API Testing Three separate layers Presentation (or user interface) layer, the business layer, and the database layer for modeling and manipulating data. API Test Actions Verify correct HTTP status for example, creating a resource should return 201 CREATED and unpermitted requests should return 403 FORBIDDEN, etc. Verify response payload check valid JSON body and correct field names, types, and values — including in error responses. Verify response headers Verify response headers HTTP server headers have implications on both security and performance. Verify basic performance interface can be easily inspected. Verify basic performance incase an operation was completed successfully but took an unreasonable amount of time, the test fails. API Test Scenario Categories 1 Basic positive tests (happy paths) 2 Extended positive testing with optional parameters 3 Destructive testing 4 Security, authorization, and permission tests (which are out of the scope of this post) Negative testing with invalid input API Example with Test Matrix API Call Action GET / users List all users GET Get user by ID GET Gusers/(id) Get user by ID GET Gusers/(id) Get user by ID GET Gusers/(id)/configurations POST Create a new configuration for user		
internet using location, name, or both. Uniform Resource Locator is used to find the location of the resource on the web. It is a reference for a resource and a way to access that resource. A URL always shows a unique resource, and it can be an HTML page, a CSS document, an image, etc. Layers of API Testing Three separate layers Presentation (or user interface) layer, the business layer, and the database layer for modeling and manipulating data. API Test Actions Verify correct HTTP status requests should return 403 FORBIDDEN, etc. Verify response payload requests should return 403 FORBIDDEN, etc. Verify response payload responses. Verify response headers HTTP server headers have implications on both security and performance. Verify correct application state interface can be easily inspected. In case an operation was completed successfully but took an unreasonable amount of time, the test fails. API Test Scenario Categories 1 Basic positive tests (happy paths) 2 Extended positive testsing with optional parameters 3 Destructive testing 4 Security, authorization, and permission tests (which are out of the scope of this post) Negative testing with valid input API Example with Test Matrix API Call Action Get Users / (id) Get user by ID Get Jusers/(id) Get user by ID Get Jusers/(id) Get user by ID Get Jusers/(id)/configurations POST Create a new configuration for user	URI	-
Uniform Resource Locator is used to find the location of the resource on the web. It is a reference for a resource and a way to access that resource. A URL always shows a unique resource, and it can be an HTML page, a CSS document, an image, etc. Layers of API Testing Three separate layers Presentation (or user interface) layer, the business layer, and the database layer for modeling and manipulating data. API Test Actions Verify correct HTTP status for example, creating a resource should return 201 CREATED and unpermitted requests should return 403 FORBIDDEN, etc. Verify response payload requests should return 403 FORBIDDEN, etc. Verify response headers Verify response headers Verify correct application state interface can be easily inspected. Verify basic performance sanity the test fails. API Test Scenario Categories 1 Basic positive test fails. API Test Scenario Categories 1 Basic positive tests (happy paths) 2 Extended positive testing with optional parameters 3 Destructive testing 4 Security, authorization, and permission tests (which are out of the scope of this post) Negative testing with valid input 6 Negative testing with valid input API Example with Test Matrix API Call Action GET /users GET Get user by ID Get user by ID Get user by ID Get users/fidly Get user by ID Get users/fidly/configurations POST Create a new configuration for user		<u> </u>
a reference for a resource and a way to access that resource. A URL always shows a unique resource, and it can be an HTML page, a CSS document, an image, etc. Layers of API Testing Three separate layers Presentation (or user interface) layer, the business layer, and the database layer for modeling and manipulating data. API Test Actions Verify correct HTTP status requests should return 403 FORBIDDEN, etc. Verify response payload Check valid JSON body and correct field names, types, and values — including in error responses. Verify response headers HTTP server headers have implications on both security and performance. Verify correct application interface can be easily inspected. Verify basic performance la case an operation was completed successfully but took an unreasonable amount of time, the test fails. API Test Scenario Categories 1 Basic positive tests (happy paths) 2 Extended positive testing with optional parameters 3 Destructive testing 4 Security, authorization, and permission tests (which are out of the scope of this post) 5 Negative testing with valid input API Example with Test Matrix API Call Action GET /users List all users GET /users List all users GET /users List all users GET /users/(id) Get user by ID GET Get user by ID GET Get all configurations for user Create a new configuration for user	LIDI	<u> </u>
Layers of API Testing Three separate layers	UKL	
Three separate layers modeling and manipulating data. API Test Actions Verify correct HTTP status requests should return 403 FORBIDDEN, etc. Verify response payload requests should return 403 FORBIDDEN, etc. Verify response headers requests should return 403 FORBIDDEN, etc. Verify response headers requests should return 403 FORBIDDEN, etc. Verify response headers requests should return 403 FORBIDDEN, etc. Verify response headers requests should return 403 FORBIDDEN, etc. Verify response headers requests should return 403 FORBIDDEN, etc. Verify response headers requests should return 403 FORBIDDEN, etc. Verify response headers requests should return 403 FORBIDDEN, etc. Verify response headers requests should return 403 FORBIDDEN, etc. Verify response headers requests should return 201 CREATED and unpermitted requests should return 403 FORBIDDEN, etc. Verify response payload requests should return 403 FORBIDDEN, etc. Verify response payload requests should return 403 FORBIDDEN, etc. Verify response payload requests should return 201 CREATED and unpermitted requests should return 201 CREATED and unpermitted requester should return 2		· ·
Presentation (or user interface) layer, the business layer, and the database layer for modeling and manipulating data. API Test Actions Verify correct HTTP status code requests should return 403 FORBIDDEN, etc. Verify response payload Check valid JSON body and correct field names, types, and values — including in error responses. Verify response headers HTTP server headers have implications on both security and performance. Verify correct application Incase an operation was completed successfully but took an unreasonable amount of time, the test fails. API Test Scenario Categories 1 Basic positive tests (happy paths) 2 Extended positive testing with optional parameters 3 Destructive testing 4 Security, authorization, and permission tests (which are out of the scope of this post) 5 Negative testing with valid input 6 Negative testing with valid input API Example with Test Matrix API Call Action GET / users Get user by ID GET / users / Gid John Care a new configurations for user	Lavors of ADI Tostina	Junique resource, and it can be an HTML page, a CSS document, an image, etc.
modeling and manipulating data. API Test Actions Verify correct HTTP status code requests should return 403 FORBIDDEN, etc. Verify response payload requests should return 403 FORBIDDEN, etc. Verify response payload responses. Verify response headers HTTP server headers have implications on both security and performance. Verify correct application state interface can be easily inspected. Verify basic performance In case an operation was completed successfully but took an unreasonable amount of time, the test fails. API Test Scenario Categories 1 Basic positive tests (happy paths) 2 Extended positive testing with optional parameters 3 Destructive testing 4 Security, authorization, and permission tests (which are out of the scope of this post) Negative testing with valid input API Example with Test Matrix API Call Action GET /users List all users GET Get user by username Get user by username Get user by username Get users/(id) Get user by ID GET Get all configurations for user		
Verify correct HTTP status code requests should return 403 FORBIDDEN, etc. Verify response payload Check valid JSON body and correct field names, types, and values — including in error responses. Verify response headers HTTP server headers have implications on both security and performance. Verify correct application state interface can be easily inspected. Verify basic performance sanity In case an operation was completed successfully but took an unreasonable amount of time, the test fails. API Test Scenario Categories 1 Basic positive tests (happy paths) 2 Extended positive testing with optional parameters 3 Destructive testing 4 Security, authorization, and permission tests (which are out of the scope of this post) 5 Negative testing with valid input API Example with Test Matrix API Call Action GET / users GET Get user by username Get user by ID GET Get all configurations for user Oreate a new configuration for user	Inree separate layers	
Verify correct HTTP status code Verify response payload requests should return 403 FORBIDDEN, etc. Verify response payload Check valid JSON body and correct field names, types, and values — including in error responses. Verify response headers HTTP server headers have implications on both security and performance. Verify correct application state interface can be easily inspected. Verify basic performance In case an operation was completed successfully but took an unreasonable amount of time, the test fails. API Test Scenario Categories 1 Basic positive tests (happy paths) 2 Extended positive testing with optional parameters 3 Destructive testing 4 Security, authorization, and permission tests (which are out of the scope of this post) 5 Negative testing with valid input API Example with Test Matrix API Call Action GET / users List all users GET Get user by username GET / users/{id} Get user by ID GET / users/{id}/configurations POST Create a new configuration for user	ADI Tost Astions	modeling and manipulating data.
requests should return 403 FORBIDDEN, etc. Verify response payload Check valid JSON body and correct field names, types, and values — including in error responses. Verify response headers HTTP server headers have implications on both security and performance. This is optional and applies mainly to manual testing, or when a UI or another interface can be easily inspected. Verify basic performance In case an operation was completed successfully but took an unreasonable amount of time, the test fails. API Test Scenario Categories Basic positive tests (happy paths) Extended positive testing with optional parameters Destructive testing Security, authorization, and permission tests (which are out of the scope of this post) Negative testing with valid input Negative testing with invalid input API Example with Test Matrix API Call Action GET / users GET Users List all users GET Users/{id} Get user by username GET / users/{id} Get user by ID GET Users/{id}/configurations POST Create a new configuration for user		For average provides a resource should return 204 CDFATED and are resulted
Check valid JSON body and correct field names, types, and values — including in error responses. Verify response headers HTTP server headers have implications on both security and performance. Verify correct application state interface can be easily inspected. Verify basic performance In case an operation was completed successfully but took an unreasonable amount of time, the test fails. API Test Scenario Categories 1 Basic positive tests (happy paths) 2 Extended positive testing with optional parameters 3 Destructive testing 4 Security, authorization, and permission tests (which are out of the scope of this post) 5 Negative testing with valid input 6 Negative testing with invalid input API Example with Test Matrix API Call Action GET / users GET Get user by username GET / users/{id} Get user by ID GET / users/{id}/configurations POST Create a new configuration for user		1
responses. Verify response headers HTTP server headers have implications on both security and performance. Verify correct application state This is optional and applies mainly to manual testing, or when a UI or another interface can be easily inspected. Verify basic performance sanity In case an operation was completed successfully but took an unreasonable amount of time, the test fails. API Test Scenario Categories Basic positive tests (happy paths) Extended positive testing with optional parameters Destructive testing Security, authorization, and permission tests (which are out of the scope of this post) Negative testing with valid input Negative testing with invalid input API Example with Test Matrix API Call Action GET /users GET Get user by username GET /users/{id} Get user by ID GET Jusers/{id} Create a new configuration for user		·
Verify response headers Verify correct application state Verify correct application state Verify basic performance interface can be easily inspected. Verify basic performance sanity API Test Scenario Categories 1 Basic positive tests (happy paths) 2 Extended positive testing with optional parameters 3 Destructive testing 4 Security, authorization, and permission tests (which are out of the scope of this post) 5 Negative testing with valid input Negative testing with invalid input API Example with Test Matrix API Call Action GET /users GET Get user by username GET /users/(id) GET get all configurations POST Create a new configuration for user	verify response payload	
Verify correct application state interface can be easily inspected. Verify basic performance sanity In case an operation was completed successfully but took an unreasonable amount of time, the test fails. API Test Scenario Categories 1 Basic positive tests (happy paths) 2 Extended positive testing with optional parameters 3 Destructive testing 4 Security, authorization, and permission tests (which are out of the scope of this post) 5 Negative testing with valid input 6 Negative testing with invalid input API Example with Test Matrix API Call Action GET /users List all users GET Get user by username {username} GET /users/{id} Get user by ID GET /users/{id}/configurations POST Create a new configuration for user	Verify response headers	'
state interface can be easily inspected. Verify basic performance In case an operation was completed successfully but took an unreasonable amount of time, the test fails. API Test Scenario Categories 1 Basic positive tests (happy paths) 2 Extended positive testing with optional parameters 3 Destructive testing 4 Security, authorization, and permission tests (which are out of the scope of this post) 5 Negative testing with valid input 6 Negative testing with invalid input API Example with Test Matrix API Call Action GET / users GET Get user by username {		
Verify basic performance sanity In case an operation was completed successfully but took an unreasonable amount of time, the test fails. API Test Scenario Categories 1 Basic positive tests (happy paths) 2 Extended positive testing with optional parameters 3 Destructive testing 4 Security, authorization, and permission tests (which are out of the scope of this post) 5 Negative testing with valid input 6 Negative testing with invalid input API Example with Test Matrix API Call Action GET / Users GET Get user by username { GET / Users/{id} Get user by ID GET / Users/{id}/configurations POST Create a new configuration for user		
sanity time, the test fails. API Test Scenario Categories 1 Basic positive tests (happy paths) 2 Extended positive testing with optional parameters 3 Destructive testing 4 Security, authorization, and permission tests (which are out of the scope of this post) 5 Negative testing with valid input 6 Negative testing with invalid input API Example with Test Matrix API Call Action GET /users List all users GET Get user by username /users?name={username} GET /users/{id} Get user by ID GET /users/{id}/configurations POST Create a new configuration for user		
API Test Scenario Categories 1 Basic positive tests (happy paths) 2 Extended positive testing with optional parameters 3 Destructive testing 4 Security, authorization, and permission tests (which are out of the scope of this post) 5 Negative testing with valid input 6 Negative testing with invalid input 6 Negative testing with invalid input API Example with Test Matrix API Call Action GET /users List all users GET Get user by username /users?name={username} GET /users/{id} Get user by ID GET /users/{id}/configurations POST Create a new configuration for user		
Basic positive tests (happy paths) Extended positive testing with optional parameters Destructive testing Security, authorization, and permission tests (which are out of the scope of this post) Negative testing with valid input Negative testing with invalid input Negative testing with invalid input API Example with Test Matrix API Call Action GET / users GET Get user by username /users?name={username} GET / users/{id} Get user by ID GET Get all configurations for user /users/{id}/configurations POST Create a new configuration for user		
Extended positive testing with optional parameters Destructive testing Security, authorization, and permission tests (which are out of the scope of this post) Negative testing with valid input Negative testing with invalid input API Example with Test Matrix API Call Action GET / users GET Get user by username GET / users/{id} GET / users by username GET / users/{id} GET Get all configurations for user Create a new configuration for user		
Destructive testing Security, authorization, and permission tests (which are out of the scope of this post) Negative testing with valid input Negative testing with invalid input API Example with Test Matrix API Call Action GET / users GET Get user by username /users?name={username} GET / users/{id} Get user by ID GET / users/{id}/configurations POST Create a new configuration for user	-	
Security, authorization, and permission tests (which are out of the scope of this post) Negative testing with valid input Negative testing with invalid input API Example with Test Matrix API Call Action GET / users GET Get user by username /users?name={username} GET / users/{id} Get user by ID GET / users/{id}/configurations POST Create a new configuration for user		
Negative testing with valid input Negative testing with invalid input API Example with Test Matrix API Call Action GET / users GET / users GET Get user by username / users?name={username} GET / users/{id} Get user by ID GET Get all configurations for user / users/{id}/configurations POST Create a new configuration for user		
API Example with Test Matrix API Call Action GET / users GET Get user by username /users?name={username} GET / users/{id} Get user by ID GET Get all configurations for user /users/{id}/configurations POST Regative testing with invalid input Action Get user Action Get users Get users Get user by username Get user by username Create a new configuration for user		
API Example with Test Matrix API Call Action GET / users List all users GET Get user by username / users?name={username} GET / users/{id} Get user by ID GET Get all configurations for user / users/{id}/configurations POST Create a new configuration for user	5	<u> </u>
API Call GET / users List all users GET / users?name={username} GET / users/{id} Get user by ID GET / users/{id}/configurations POST Action Action Get users Get users Get user by username Get user by ID Get all configurations for user Create a new configuration for user	6	1 · · · ·
GET /users GET Get user by username /users?name={username} GET /users/{id} Get user by ID GET Get all configurations for user /users/{id}/configurations POST Create a new configuration for user	•	ix
GET /users?name={username} GET /users/{id} Get user by ID GET Get all configurations for user /users/{id}/configurations Create a new configuration for user	API Call	Action
/users?name={username} GET /users/{id} Get user by ID GET Get all configurations for user /users/{id}/configurations POST Create a new configuration for user	GET /users	List all users
GET /users/{id} Get user by ID GET Get all configurations for user /users/{id}/configurations POST Create a new configuration for user	GET	Get user by username
GET Get all configurations for user /users/{id}/configurations POST Create a new configuration for user	/users?name={username}	
/users/{id}/configurations POST Create a new configuration for user	GET /users/{id}	Get user by ID
/users/{id}/configurations POST Create a new configuration for user	GET	Get all configurations for user
POST Create a new configuration for user	/users/{id}/configurations	
/users/{id}/configurations	POST	Create a new configuration for user
	/users/{id}/configurations	

DELETE	Delete and Security of the second
DELETE	Delete configuration for user
/users/{id}/configurations/{i	
d}	
PATCH	Update configuration for use
/users/{id}/configuration/{id}	
Web Services	
SOAP	(Simple Object Access Protocol) is a standard protocol defined by the W3C standards
	for sending and receiving web service requests and responses.
REST	(REpresentational State Transfer) is the web standards-based architecture that uses
	HTTP. Unlike SOAP-based Web services, there is no official standard for RESTful Web
CRUD	Create, Read, Update & Delete
HTTP Request Methods	
GET	It fetches the information from the server. Moreover, it is the most commonly used
	method which does not have a request body. Every time you open a website, the Get
	request fires to retrieve the website contents. Additionally, it is equivalent to the
POST	It works to send data to the server. User may add or update data using the Post
	request. They send the information that needs to update in the request body.
PUT	It is similar to the Post method since it updates the data. The only difference is that
	we use it when we have to replace an existing entity completely
PATCH	It s again similar to Post and Put methods, but user use it when they have to update
	some data partially. Moreover, unlike the Post and Put methods, user may send only
	the entity that needs updation in the request body with the Patch method.
HEAD	It is similar to the Get method, but it retrieves only the header data and not the
	entire response body. User use it when they need to check the document's file size
	without downloading the document.
DELETE	It deletes the server's representations of resources through the specific URL.
	Additionally, just like the Get method, it does not have a request body.
OPTIONS	It is not a widely used method when compared to other ones. It returns data
	specifying the different methods and the operations supported by the server at the
HTTP Response Status Codes	
Code	Description
1xx	informational response, request was received, continuing process
100	Continue: The client can continue with the request as long as it doesn't get rejected.
101	Switching Protocols: The server is switching protocols.
102	Processing, It indicates that the server has received and is processing the request, but
102	no response is available yet.
102	Early Hints, it primarily intended to be used with the Link header, letting the user
103	agent start preloading resources while the server prepares a response.
2xx	Success, request was successfully received, understood, and accepted
	· · · · · · · · · · · · · · · · · · ·
200	OK: The request succeeded
201	Created: The request succeeded, and a new resource was created as a result. This is
202	typically the response sent after POST requests, or some PUT requests.
202	Accepted: Request accepted for processing, but in progress
203	Non-Authoritative Information: The information in the entity header is not from an
	original source but a third-party
204	No Content: Response with status code and header but no response body
205	Reset Content: The form for the transaction should clear for additional input
206	Partial Content: Response with partial data as specified in Range header
	-

207	Multi-Status, Conveys information about multiple resources, for situations where
207	multiple status codes might be appropriate.
3xx	Redirection, further action needed in order to complete the request
300	Multiple Choices: Response with a list for the user to select and go to a location
301	Moved Permanently: Requested page moved to a new url
302	Found: Requested page moved to a temporary new URL
303	See Other: One can find the Requested page under a different URL
305	Use Proxy: Requested URL need to access through the proxy mentioned in the
303	Location header
307	Temporary Redirect: Requested page moved to a temporary new URL
308	Permanent Redirect: This means that the resource is now permanently located at
	another URI, specified by the Location: HTTP Response header.
4xx	Client Error, request contains bad syntax or cannot be fulfilled
400	Bad Request: Server unable to understand the request
401	Unauthorized: Requested content needs authentication credentials
403	Forbidden: Access is forbidden
404	Not Found: Server is unable to find the requested page
405	Method Not Allowed: Method in the request is not allowed
407	Proxy Authentication Required: Need to authenticate with a proxy server
408	Request Timeout: The request took a long time as expected by the server
409	Conflict: Error in completing request due to a conflict
411	Length Required: We require the "Content-Length" for the request to process
415	Unsupported Media Type: Unsupported media-type
417	Expectation Failed, it means the expectation indicated by the Expect request header
	field cannot be met by the server.
421	Misdirected Request, request was directed at a server that is not able to produce a
423	Locked, the resource that is being accessed is locked
429	Too Many Requests, user has sent too many requests in a given amount of time
5xx	Server Error, the server failed to fulfil an apparently valid request
500	Internal Server Error: Request not completed due to server error
501	Not Implemented: Server doesn't support the functionality
502	Bad Gateway: Invalid response from an upstream server to the server. Hence, the
	request not complete
503	Service Unavailable: The server is temporarily down
504	Gateway Timeout: The gateway has timed out
505	HTTP Version Not Supported: Unsupported HTTP protocol version
507	Insufficient Storage, method could not be performed on the resource because the
	server is unable to store the representation needed to successfully complete the
<u>511</u>	Network Authentication Required, it indicates that the client needs to authenticate
	to gain network access