Katalon Postman SoapUI Rest-Assured CITRUS Karate Jimeter 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 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. 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 FORBIODEN, etc. Verify response headers For example, creating a resource should return 201 CREATED and unpermitted requests should return 403 FORBIODEN, etc. Verify response headers HTTP server headers have implications on both security and performance. Verify torect application in 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 the protocology of the string with optional parameters Destructive testing Basic positive tests (happy paths) Extended positive testing with optional parameters Destructive testing with part of the scope of this post; Negative testing with valid input Pressor Late 1 Security, authorization, and permission tests (which are out of the scope of this	Top API Testing Tools	
SoapUI Rest-Assured CITRUS Karate Jimeter apigee API Terminologies API Terminologies 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 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. 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 This is optional and applies mainly to manual testing, or when a UI or another interface can be easily inspected. Verify sponse headers Verify correct application This is optional and applies mainly to manual testing, or when a UI or another interface can be easily inspected. Verify sponse headers 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 with valid input API Example with Test Matrix API Test Scenario Categories Extended positive testing with optional parameters Destructive testing API Example with Test Matrix API Call Action GET / User		Doctman
CITRUS Karate Imeter 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 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. It 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 correct application This is optional and applies mainly to manual testing, or when a U1 or another interface can be easily inspected. Verify basic performance In case an operation was completed successfully but took an unreasonable amount of the second of the seco		
API Crest Actions API Test Secanario Categories AP	·	
API Terminologies APP 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 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. 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 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 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 This is optional and applies mainly to manual testing, or when a UI or another interface can be easily inspected. Verify basic performance Basic positive tests (happy paths) Extended positive testing with optional parameters Presented positive testing with optional parameters Destructive testing Presented	CITRUS	
API Application Programming Interface (API) is software that acts as an intermediary for two apps to communicate with each other. 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 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 case an operation was completed successfully but took an unreasonable amount of the state interface can be easily inspected. Verify basic performance Jacaba and papilies mainly to manual testing, or when a UI or another interface can be easily inspected. Verify the sound Categories 1 Basic positive tests (happy paths) Extended positive testing with optional parameters Destructive testing API Test Scenario Categories Negative testing with valid input API Example with Test Matrix API Call Action GET /users List all users Get user by ID GET Get all configurations for user		apigee
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 and responses URI Uniform Resource Identifier is a string identifier that refers to a 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 URI 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 tassic 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; Security, authorization, and permission tests (which are out of the scope of this post; Security and performance) 6ET / Users List all users GET Get user by UB GET Get all configurations 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 S in HTPS 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 even the internet using location, name, or both. URL Uniform Resource Locator is used to find the location of the resource on the internet using location, name, or both. URL 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 headers Presentation (and presentation and presentation on both security and performance. Verify response headers HTP 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 testis (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) for the secure of this post; 5 Negative testing with invalid input API Example with Test Matrix API Example with Test Matrix GET Get user by ID GET Get all configurations 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 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 for example, creating a resource should return 201 CREATED and unpermitted requests should return 403 FORBIDDEN, etc. Verify response payload Check valid ISON body and correct field names, types, and values — including in error responses. Verify correct application State interface and performance 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; he test fails with valid input 6 Negative testing with valid input 6 Negative testing with valid input API Example with Test Matrix API Call Get user by ID GET /users Get user by UB GET Get all configurations for user		
HTTPS 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 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 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) for the scape 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 List all users GET Get user by ID GET Get all configurations for user	HTTP	1
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. 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 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) for the scape of this post; when a UI or another interface can be easily invalid input 6 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 ID GET of Get user by ID GET for Get all configurations for user		
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 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) 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 ID GET Get all configurations 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 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 tests (happy paths) 2 Extended positive testing with optional parameters 3 Destructive testing with optional parameters 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 Get user by ID GET Get all configurations for user		
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 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) Security, authorization, and permission tests (which are out of the scope of this post) Security authorization, and permission tests (which are out of the scope of this post) Security authorization, and permission tests (which are out of the scope of this post) Security authorization, and permission tests (which are out of the scope of this post) Security authorization, and permission tests (which are out of the scope of this post) Security authorization, and permission tests (which are out of the scope of this post) Security authorization, and permission tests (which are out of the scope of this post) Security authorization, and permission tests (which are out of the scope of this post) Security authorization, and permission tests (which are out of the scope of this post) Security authorization authorization Security authorization Security authorizati	URI	_
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 correct application 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) Segative testing with valid input 6 Negative testing with valid input API Example with Test Matrix API Call ACtion GET / users GET / users Get user by ID GET Get all configurations for user		
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 erro 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 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) Segative testing with valid input 6 Negative testing with valid input API Example with Test Matrix API Call Action GET / users GET Get all configurations for user		
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 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 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 / users / (id) Get user by ID GET / users / (id)	UKL	
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 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		· · · · · · · · · · · · · · · · · · ·
Three separate layers	Lovers of ADI Tasking	Junique resource, and it can be an HTML page, a CSS document, an image, etc.
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 or time, the test fails. API Test Scenario Categories 1 Basic positive tests (happy paths) 2 Extended 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 Get user by username GET / users/[id] Get user by ID GET / users/[id] 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 6 Negative testing with invalid input API Example with Test Matrix API Call Action GET / users / Get user by username GET / users/{id} Get user by ID GET / users/{id} Get all configurations for user	Three separate layers	
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 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 GET / users/fid} Get user by ID GET Get all configurations for user	ADI Took Askinson	Imodeling and manipulating data.
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 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 6 Negative testing with invalid input API Example with Test Matrix API Call Action GET / users GET Get user by username GET / users/fidl Get user by ID GET Get all configurations for user		Is
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 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 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/fid} Get user by ID GET Get all configurations for user	•	· · · · · · · · · · · · · · · · · · ·
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 inc 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 {users?name={username}} GET / users/{id} Get all configurations for user		·
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 Get all configurations for user	verity response payload	<u> </u>
Verify correct application state interface can be easily inspected. Verify basic performance sanity image. 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 6 Negative testing with invalid 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 Get all configurations for user	Varify response headers	
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 GET Get user by username {users?name={username}} GET / users/{id} Get user by ID GET Get all configurations for user		
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/fid} Get user by ID GET / Get all configurations 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 (users Get user by username) (users?name={username} GET /users/{id} Get user by ID GET Get all configurations 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 API Example with Test Matrix API Call Action GET /users GET Jesers List all users GET Get user by username /users?name={username} GET /users/{id} Get user by ID GET Get all configurations 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		tille, the test rails.
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 GET / users/{id} Get user by ID GET Get all configurations for user		Pacie positiva taste (hanny paths)
Destructive testing 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 Get all configurations 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 Get all configurations for user		
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 Get all configurations for user		-
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 Get all configurations 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	5	<u> </u>
API Call GET /users List all users GET /users?name={username} GET /users/{id} Get user by ID GET Get all configurations for user		, ,
GET /users GET /users Get user by username /users?name={username} GET /users/{id} Get user by ID GET Get all configurations for user	•	X .
GET / Get user by username / Users?name={username} GET / users/{id} Get user by ID GET Get all configurations for user	API Call	Action
/users?name={username} GET /users/{id} Get user by ID GET Get all configurations for user	GET /users	List all users
GET /users/{id} Get user by ID GET Get all configurations for user	GET	Get user by username
GET Get all configurations for user	/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		Create a new configuration for user
/users/{id}/configurations	/users/{id}/configurations	

Delete configuration for user // users/(id)/configurations/(id) PATCH // users/(id)/configuration/(id) Web Services SOAP (Simple Object Access Protocol) is a standard protocol defined by the W3C standard for sending and receiving web service requests and responses. REST (REpresentational State Transfer) is the web standards-based architecture that use HTTP. Unlike SOAP-based Web services, there is no official standard for RESTful V CRUD Create, Read, Update & Delete HTTP Request Methods GET It fetches the information from the server. Moreover, it is the most commonly use method which does not have a request body. Every time you open a website, the request fires to retrieve the website contents. Additionally, it is equivalent to the request fires to retrieve the information that needs to 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 the we use it when we have to replace an existing entity completely PATCH It is again similar to Post and Put methods, but user use it when they have to update one data partially. Moreover, unlike the Post and Put methods, user may send of 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 six 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 other compared to other ones.
Description of the service of the standard for service requests and responses.
PATCH /users/{id}/configuration/{id}} Web Services SOAP (Simple Object Access Protocol) is a standard protocol defined by the W3C standard for sending and receiving web service requests and responses. REST (REpresentational State Transfer) is the web standards-based architecture that us HTTP. Unlike SOAP-based Web services, there is no official standard for RESTful V CRUD Create, Read, Update & Delete HTTP Request Methods GET It fetches the information from the server. Moreover, it is the most commonly us method which does not have a request body. Every time you open a website, the 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 the 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 of 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 six 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
/users/{id}/configuration/{id} Web Services SOAP (Simple Object Access Protocol) is a standard protocol defined by the W3C standard for sending and receiving web service requests and responses. REST (REpresentational State Transfer) is the web standards-based architecture that us HTTP. Unlike SOAP-based Web services, there is no official standard for RESTful V CRUD (Create, Read, Update & Delete) HTTP Request Methods GET It fetches the information from the server. Moreover, it is the most commonly us method which does not have a request body. Every time you open a website, the 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 the 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 on the entity that needs updation in the request body with the Patch method. 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 similar to 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. It is not a widely used method when compared to other ones. It returns data
Web Services SOAP (Simple Object Access Protocol) is a standard protocol defined by the W3C standard for sending and receiving web service requests and responses. REST (REpresentational State Transfer) is the web standards-based architecture that us HTTP. Unlike SOAP-based Web services, there is no official standard for RESTful V Creute, Read, Update & Delete HTTP Request Methods GET It fetches the information from the server. Moreover, it is the most commonly us method which does not have a request body. Every time you open a website, the 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 the 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 of the entity that needs updation in the request body with the Patch method. 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 sit 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
(Simple Object Access Protocol) is a standard protocol defined by the W3C standard for sending and receiving web service requests and responses. REST (REpresentational State Transfer) is the web standards-based architecture that us HTTP. Unlike SOAP-based Web services, there is no official standard for RESTful V CRUD Create, Read, Update & Delete HTTP Request Methods GET It fetches the information from the server. Moreover, it is the most commonly us method which does not have a request body. Every time you open a website, the 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 the 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 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 six 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.
for sending and receiving web service requests and responses. REST (REpresentational State Transfer) is the web standards-based architecture that us HTTP. Unlike SOAP-based Web services, there is no official standard for RESTful V CRUD Create, Read, Update & Delete HTTP Request Methods GET It fetches the information from the server. Moreover, it is the most commonly us method which does not have a request body. Every time you open a website, the 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 the we use it when we have to replace an existing entity completely PATCH It is 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 of 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 six 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
for sending and receiving web service requests and responses. REST (REpresentational State Transfer) is the web standards-based architecture that us HTTP. Unlike SOAP-based Web services, there is no official standard for RESTful V CRUD Create, Read, Update & Delete HTTP Request Methods GET It fetches the information from the server. Moreover, it is the most commonly us method which does not have a request body. Every time you open a website, the 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 the we use it when we have to replace an existing entity completely PATCH It is 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 of 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 six 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
REST (REpresentational State Transfer) is the web standards-based architecture that us HTTP. Unlike SOAP-based Web services, there is no official standard for RESTful V CRUD Create, Read, Update & Delete HTTP Request Methods GET It fetches the information from the server. Moreover, it is the most commonly us method which does not have a request body. Every time you open a website, the 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 the we use it when we have to replace an existing entity completely PATCH It is 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 of 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 six 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
HTTP. Unlike SOAP-based Web services, there is no official standard for RESTful V CRUD Create, Read, Update & Delete HTTP Request Methods GET It fetches the information from the server. Moreover, it is the most commonly us method which does not have a request body. Every time you open a website, the 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 the 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 of 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 similar to 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
CRUD Create, Read, Update & Delete HTTP Request Methods GET It fetches the information from the server. Moreover, it is the most commonly us method which does not have a request body. Every time you open a website, the 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 the 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 of 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 six 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
GET It fetches the information from the server. Moreover, it is the most commonly us method which does not have a request body. Every time you open a website, the request fires to retrieve the website contents. Additionally, it is equivalent to the 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 the we use it when we have to replace an existing entity completely PATCH It is 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 of 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 six 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. It is not a widely used method when compared to other ones. It returns data
GET It fetches the information from the server. Moreover, it is the most commonly us method which does not have a request body. Every time you open a website, the request fires to retrieve the website contents. Additionally, it is equivalent to the 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 the we use it when we have to replace an existing entity completely PATCH It is 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 of 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 six 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. It is not a widely used method when compared to other ones. It returns data
method which does not have a request body. Every time you open a website, the 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 the we use it when we have to replace an existing entity completely PATCH It is 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 of 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 six 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
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 the we use it when we have to replace an existing entity completely PATCH It is 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 of 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 six 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
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 the 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 of 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 six 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
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 the we use it when we have to replace an existing entity completely PATCH It is 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 of 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 six 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
PUT It is similar to the Post method since it updates the data. The only difference is the we use it when we have to replace an existing entity completely It is 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 of the entity that needs updation in the request body with the Patch method. 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 six 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
PATCH It is 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 of 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 six 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
PATCH It is 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 of 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 six 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
some data partially. Moreover, unlike the Post and Put methods, user may send of 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 six 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
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 six 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
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 six 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
entire response body. User use it when they need to check the document's file six 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
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
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
OPTIONS It is not a widely used method when compared to other ones. It returns data
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 t
papernying the different methods and the operations supported by the server at t
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 reject
101 Switching Protocols: The server is switching protocols.
102 Processing, It indicates that the server has received and is processing the request
no response is available yet.
103 Early Hints, it primarily intended to be used with the Link header, letting the user
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
typically the response sent after POST requests, or some PUT requests.
202 Accepted: Request accepted for processing, but in progress
Non-Authoritative Information: The information in the entity header is not from a
original source but a third-party
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	Adulti Status Commission and include a substitution of the status of the
207	Multi-Status, Conveys information about multiple resources, for situations where
3xx	multiple status codes might be appropriate. 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
	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
511	Network Authentication Required, it indicates that the client needs to authenticate
	to gain network access