GetBookingIds Tests

Test Case ID	Test Scenario Description	Test Case Description	Prerequisites	Steps Description	Method	Request	Expected Response
GBTC01	GetBookingIds endpoint works when no parameters are sent	GetBookingIds returns a list of BookingIds when invoked with no parameters	- At least a couple of Bookings should exist	1. Perform a request to GetBookingIds using no parameters 2. Validate the response	GET	curl --location --request GET 'https://restful-booker.herokuapp.com/booking'	- 200 OK status - JSON Collection contais Bookings - JSON Collection is not empty
GBTC02	GetBookingIds endpoint works when one parameter is sent	GetBookingIds returns a list of BookingIds when invoked with one valid parameter	- At least a couple of Bookings should exist	1. Perform a request to GetBookingIds using a valid parameter 2. Validate the response	GET	curl --location --request GET 'https://restful-booker.herokuapp.com/booking?firstname=Sally'	- 200 OK status - JSON Collection with all BookingIds filtered by the parameter
GBTC03	GetBookingIds endpoint works when one parameter is sent	GetBookingIds returns a list of BookingIds when invoked with one invalid parameter	- At least a couple of Bookings should exist	1. Perform a request to GetBookingIds using an invalid parameter 2. Validate the response	GET	curl --location --request GET 'https://restful-booker.herokuapp.com/booking?RandomFirstname=Sally'	- 200 OK status - JSON Collection with all existing BookingIds
GBTC04	GetBookingIds endpoint works when one parameter is sent	GetBookingIds returns an empty list when invoked with one parameter for non existing value	- At least a couple of Bookings should exist	1. Perform a request to GetBookingIds using non existing parameter value 2. Validate the response	GET	curl --location --request GET 'https://restful-booker.herokuapp.com/booking?firstname=Sally1986'	- 200 OK status - Empty JSON Collection
GBTC05	GetBookingIds endpoint works when two parameter are sent	GetBookingIds returns a list of BookingIds when invoked with two valid parameters	- At least a couple of Bookings should exist	1. Perform a request to GetBookingIds using two valid parameters 2. Validate the response	GET	curl --location --request GET 'https://restful-booker.herokuapp.com/booking?firstname=Sally&lastname=Brown'	- 200 OK status - JSON Collection with all BookingIds filtered by both parameters
GBTC06	GetBookingIds endpoint works when two parameter are sent	GetBookingIds returns a list of BookingIds when invoked with one valid parameter and another invalid parameter	- At least a couple of Bookings should exist	1. Perform a request to GetBookingIds using one valid parameter and one invalid parameter 2. Validate the response	GET	curl --location --request GET 'https://restful-booker.herokuapp.com/booking?firstname=Javier&RandomLastname=Brown'	- 200 OK status - JSON Collection with all BookingIds filtered by valid parameter and ignore invalid parameter
GBTC07	GetBookingIds endpoint works when two parameter are sent	GetBookingIds returns a list of BookingIds when invoked with two invalid parameters	- At least a couple of Bookings should exist	1. Perform a request to GetBookingIds using two invalid parameters 2. Validate the response	GET	curl --location --request GET 'https://restful-booker.herokuapp.com/booking?RandomFirstname=Javier&RandomLastname=Brown'	- 200 OK status - JSON Collection with all existing BookingIds
GBTC08	GetBookingIds endpoint works when two parameter are sent	GetBookingIds returns an empty list of BookingIds when invoked with two non existing parameter values	- At least a couple of Bookings should exist	1. Perform a request to GetBookingIds using two non existing parameter values 2. Validate the response	GET	curl --location --request GET 'https://restful-booker.herokuapp.com/booking?firstname=Sally1984&lastname=Brown1986'	- 200 OK status - Empty JSON Collection

DeleteBooking Tests

Test Case ID	Test Scenario Description	Test Case Description	Prerequisites	Steps Description	Method	Request	Expected Response
DBTC01	DeleteBooking endpoint works successfully	DeleteBooking endpoint removes a booking when using a valid ID	- A Booking should exist - Valid Token	1. Add Valid Token to request 2. Perform a request to DeleteBooking using valid Booking ID 3. Validate the response	DELETE	curl --location --request DELETE 'https://restful-booker.herokuapp.com/booking/21966' \ --header 'Cookie: token=08db0a558d0ad68'	- 201 Created status - Created message is displayed
DBTC02	DeleteBooking endpoint doesn't crash when no parameters are sent	DeleteBooking endpoint returns NotFound when no ID parameter is sent	- Valid Token	1. Add Valid Token to request 2. Perform a request to DeleteBooking without ID parameter 3. Validate the response	DELETE	curl --location --request DELETE 'https://restful-booker.herokuapp.com/booking/21966' \ --header 'Cookie: token=08db0a558d0ad68'	- 404 Not Found Status - Not Found message is displayed
DBTC03	DeleteBooking endpoint doesn't crash when invalid parameters is sent	DeleteBooking endpoint returns NotAllowed when ID parameter is invalid	- Valid Token	1. Add Valid Token to request 2. Perform a request to DeleteBooking with invalid ID value 3. Validate the response	DELETE	curl --location --request DELETE 'https://restful-booker.herokuapp.com/booking/abcd' \ --header 'Cookie: token=08db0a558d0ad68'	- 405 Method Not Allowed status - Method Not Allowed message is displayed
DBTC04	DeleteBooking endpoint doesn't remove bookings when it doesn't find bookings	DeleteBooking endpoint returns NotAllowed when ID parameter doesn't exist	- Valid Token	1. Add Valid Token to request 2. Perform a request to DeleteBooking with non existing ID value 3. Validate the response	DELETE	curl --location --request DELETE 'https://restful-booker.herokuapp.com/booking/99999' \ --header 'Cookie: token=08db0a558d0ad68'	- 405 Method Not Allowed status - Method Not Allowed message is displayed
DBTC05	DeleteBooking endpoint doesn't remove bookings when unauthorized	DeleteBooking endpoint returns Forbidden when a valid Booking ID and no Token is sent	- A Booking should exist	1. Perform a request to DeleteBooking using valid Booking ID 2. Validate the response	DELETE	curl --location --request DELETE 'https://restful-booker.herokuapp.com/booking/2160' \ --header 'Cookie;'	- 403 Forbidden status - Forbidden message is displayed
DBTC06	DeleteBooking endpoint doesn't remove bookings when unauthorized	DeleteBooking endpoint returns Forbidden when a valid Booking ID and invalid Token is sent	- A Booking should exist	1. Add Invalid Token to request 2. Perform a request to DeleteBooking using valid Booking ID 3. Validate the response	DELETE	curl --location --request DELETE 'https://restful-booker.herokuapp.com/booking/722' \ --header 'Cookie: token=zzz972zz993z284'	- 403 Forbidden status - Forbidden message is displayed

PartialUpdateBooking Tests

Test Case ID	Test Scenario Description	Test Case Description	Prerequisites	Steps Description	Method	Request	Expected Response
PUTC01	PartialUpdate endpoint works successfully	PartialUpdate endpoint updates a booking when using valid parameters	- A Booking should exist - Valid Token - Valid data to update	1. Add Valid Token to request 2. Perform a request to PartialUpdate endpoint using valid data 3. Validate the response	PATCH	curl --location --request PATCH 'https://restful-booker.herokuapp.com/booking/1585' \ --header 'Content-Type: application/json' \ --header 'Accept: application/json' \ --header 'Cookie: token=dbe972ee993b284' \ --data-raw '{ "firstname" : "James", "lastname" : "Brown" }'	- 200 Ok status - Booking data updated
PUTC02	PartialUpdate endpoint doesn't update Bookings when Unauthorized	PartialUpdate endpoint displays Forbidden message when valid parameters are sent and token is empty	- A Booking should exist - Valid data to update	1. Perform a request to PartialUpdate endpoint using valid data 2. Validate the response	PATCH	curl --location --request PATCH 'https://restful-booker.herokuapp.com/booking/1585' \ --header 'Content-Type: application/json' \ --header 'Accept: application/json' \ --data-raw '{ "firstname" : "James1", "lastname" : "Brown1" }'	- 403 Forbidden status - Forbidden message is displayed - Booking data is not updated
PUTC03	PartialUpdate endpoint doesn't update Bookings when Unauthorized	PartialUpdate endpoint displays Forbidden message when valid parameters are sent and token is invalid	- A Booking should exist - Valid data to update	1. Add Invalid Token to request 2. Perform a request to PartialUpdate endpoint using valid data 3. Validate the response	PATCH	curl --location --request PATCH 'https://restful-booker.herokuapp.com/booking/1585' \ --header 'Content-Type: application/json' \ --header 'Accept: application/json' \ --header 'Cookie: token=zzz972zz993z284' \ --data-raw '{ "firstname" : "James1", "lastname" : "Brown1" }'	- 403 Forbidden status - Forbidden message is displayed - Booking data is not updated
PUTC04	PartialUpdate endpoint doesn't update Bookings when Booking invalid ID	PartialUpdate endpoint displays Method not allowed message when valid parameters are sent, token is valid but BookingID doesn't exist	- A Booking should exist - Valid data to update	1. Add Invalid Token to request 2. Perform a request to PartialUpdate endpoint using valid data 3. Validate the response	PATCH	curl --location --request PATCH 'https://restful-booker.herokuapp.com/booking/1585' \ --header 'Content-Type: application/json' \ --header 'Accept: application/json' \ --header 'Cookie: token=zzz972zz993z284' \ --data-raw '{ "firstname" : "James1", "lastname" : "Brown1" }'	- 405 Method Not Allowed status - Method Not Allowed message is displayed - Booking data is not updated