Proyect Name	datausa API/NBA Page
Document Type	TC-test_Pages
Autor	Christopher Rodriguez
reviewed by	<<Chris >>

Project Manager	Chris
Version	1
creation date	01/01/01
review date	01/01/01

TC-ID	TC-type	Description	Pre-conditions	Expected Result	Results obtained	Status	Comments
Tc_01		This test case evaluates both the performance (via response time) and basic functionality (via HTTP status code) of the API, ensuring that the API is accessible and responds in a reasonable time.	The specific API or endpoint must be operational and accessible at the time of testing. There must be a properly configured HTTP request pointing to the endpoint you want to test.	The HTTP status code returned by the response should be 200, indicating that the request was successful. The response time should be less than 5000 milliseconds (or 5 seconds), which indicates a timely response from the server.		Pass	If the test fails due to high response time, it could indicate performance issues on the server or network. If the status code is not 200, it indicates that there was an error processing the request, which requires additional investigation to identify the cause (for example, authentication issues, server errors).
Tc_02		This test confirms that the response body is a valid JSON object, which is essential for APIs that communicate data in this format.	An HTTP request must be made to the API	The response body must be a valid JSON object	The response is a valid JSON object	Pass	Essential for APIs that are expected to return data in JSON format
Tc_03		The API does not return a response in JSON format	An HTTP request must be made to the API	the API returns a response in XML format	The API response is a JSON object and not an XML	Fail	The API was built under a JSON format therefore it returns that format.
Tc_04		Make sure the API handles queries for non-existent data gracefully	An HTTP request must be made to the API using the wrong data	Status code 200 response contains an empty data array	the API response was an empty array	Pass	The API should be prepared to have error handling
Tc_05		Ensure the API handles case-insensitive queries	An HTTP request must be made to the API using upper/lower case	Status code 200 Response contains data related	The response contains the desired information regardless of whether it is uppercase or lowercase.	Pass	Request parameters should be case-insensitive to requests, if not specified within the documentation
Tc_06		Use an invalid HTTP (POST) method to make the request	The API must be configured and expect specific HTTP methods (GET) for	Status code 405 (method not allowed)	Status code 404 (not found)	Fail	The API response should be set to code 405, for unsupported methods.

			the defined endpoints.				
Tc_07		Use a limit parameter value greater than the maximum allowed	Make an HTTP request to the API, including an amount in the response	Status code 400 (bad request)	Status code 400 (bad request)	Pass	If the limit amount to be displayed is established, it should not display more than the existing data.
Tc_08		Use a non-valid value for set parameters	Make an HTTP request to the API, including an invalid parameter in the response	Status code 400 (bad request)	Status code 400 (bad request)	Pass	The API should be prepared to handle invalid data errors.
Tc_09		Omit the query parameters entirely	Make an HTTP request to the API without including data	Status code 200 (OK)	Status code 200 "error": "Query must contain at least one measure."	Pass	Exception handling was added to the API in case of not having measures
Tc_10		Insert a SQL injection through search parameters, such as "DROP table **"	Online API a request is made to the API	The API handles the injection attempt safely, without running the script.	Status 200 The API reacts to the script and returns a response	Fail	The API reacted to the DROP TABLE injection and made a modification
Tc_11		Attempt to inject a malicious script, such as "<script>alert('Hack!')</script>".	Online API a request is made to the API	The API handles the injection attempt safely, without running the script.	Status 200 The API reacts to the script and returns a response	Fail	The API reacted to the SCRIPT insert and made a modification
Tc_12		Try to login to NBA account.	have a valid account and password	enter your NBA profile	Successfully enter your NBA profile	Pass	The page usually takes time to load data such as carousels, images, etc.
Tc_13		Try to fail login to NBA account	have a dummy account or password	It should show us a message informing us that the credentials failed and a ReChapta	successfully displays the message in red and a ReCaptcha	Pass	After failing to log in, the page does not delete the failed credentials, you have to delete them manually

Tc_14		Hover over any of the "NavBar" fields, a field should be displayed.	Enter to NBA.COM	It should show the fields of your section	shows the fields of each section	PASS	
Tc_15		Hover over any of the “Game Chards” fields, a field should be displayed.	Enter to NBA.COM	It should show the game chards of every game	show every section	PASS	game chard sections are often cut in "mobile" versions