

ORITEDDY'S TEST PLAN

Test plan objective :

Our test plan will allow us to set up the unit tests necessary to verify the proper functioning of our « Oriteddy » application.

Test environment :

Unit tests will be made with the JavaScript tests framework : Jest.
All unit tests will be stored in a separate folder named « tests » at the root of the project.

Test N°1 : Check Fetch API

Description : Check the fetchAPI function get a positive response from the API.

Parameters : url + endpoint '/teddies', method, body

Pass if returns : A 201 response contains list of products.

Test N° 2 : Check Fetch API error

Description : Check the fetchAPI function get an error in the response from the API.

Parameters : url, method, body

Pass if returns : 400, 404 or 500 http error code.

Test N°3 : Check TeddyBear's ID is saved

Description : Check new TeddyBear instance saves a new ID.

Parameters : new TeddyBear(id, image, name, description, colors, price, quantity)

Pass if returns : teddyBear with the new ID.

Test N°4 : Check TeddyBear's image is saved

Description : Check new TeddyBear instance saves a new image.

Parameters : new TeddyBear(id, image, name, description, colors, price, quantity)

Pass if returns : teddyBear with the new image.

Test N°5 : Check TeddyBear's name is saved

Description : Check new TeddyBear instance saves a new name.

Parameters : new TeddyBear(id, image, name, description, colors, price, quantity)

Pass if returns : teddyBear with the new name.

Test N°6 : Check TeddyBear's description is saved

Description : Check new TeddyBear instance saves a new description.

Parameters : new TeddyBear(id, image, name, description, colors, price, quantity)

Pass if returns : teddyBear with the new description.

Test N°7 : Check TeddyBear's array of colors is saved

Description : Check new TeddyBear instance saves a new array of colors.

Parameters : new TeddyBear(id, image, name, description, colors, price, quantity)

Pass if returns : teddyBear with the new array of colors.

Test N°8 : Check TeddyBear's default color does not exist

Description : Check new TeddyBear instance don't have a default color.

Parameters : new TeddyBear(id, image, name, description, colors = [], price, quantity)

Pass if returns : teddyBear don't have color saved.

Test N°9 : Check TeddyBear's price is saved

Description : Check new TeddyBear instance saves a new price.

Parameters : new TeddyBear(id, image, name, description, colors, price, quantity)

Pass if returns : teddyBear with the new price.

Test N°10 : Check TeddyBear has default quantity

Description : Check new TeddyBear instance has a default quantity.

Parameters : new TeddyBear(id, image, name, description, colors, price, quantity)

Pass if returns : teddyBear has a default quantity = 0.

Test N°11 : Check TeddyBear can set a new ID

Description : Check new TeddyBear instance can set a new ID.

Parameters : teddyBear.Id = « new Id ».

Pass if returns : teddyBear has the new ID.

Test N°12 : Check TeddyBear can set a new image

Description : Check new TeddyBear instance can set a new image.

Parameters : teddyBear.Image = « new Image ».

Pass if returns : teddyBear has the new image.

Test N°13 : Check TeddyBear can set a new name

Description : Check new TeddyBear instance can set a new name.

Parameters : teddyBear.Name = « new Name ».

Pass if returns : teddyBear has the new name.

Test N°14 : Check TeddyBear can set a new description

Description : Check new TeddyBear instance can set a new description.

Parameters : teddyBear.Description = « new Description ».

Pass if returns : teddyBear has the new description.

Test N°15 : Check TeddyBear can set a new color

Description : Check new TeddyBear instance can set a new color.

Parameters : teddyBear.ChosenColor = « new Color ».

Pass if returns : teddyBear has the new color.

Test N°16 : Check TeddyBear can set a new price

Description : Check new TeddyBear instance can set a new price.

Parameters : teddyBear.Price = « new Price ».

Pass if returns : teddyBear has the new price.

Test N°17 : Check TeddyBear can set a new quantity

Description : Check new TeddyBear instance can set a new quantity.

Parameters : teddyBear.Quantity = « new Quantity ».

Pass if returns : teddyBear has the new quantity.

Test N°18 : Check Append Items list in Dropdown Menu

Description : Check the appendItemInViewDropdownMenu returns items list in dropdown menu

Parameters : item, url

Pass if returns : items list in dropdown menu.

Test N° 19 : Check Append items in index page

Description : Check the appendItemInViewIndex function displays items in index page.

Parameters : item

Pass if returns : items number in index page > 0.

Test N°20 : Check Get Url Parameter

Description : Check the getUrlParam function returns url parameter contains product's ID

Parameters : url parameter name

Pass if returns : product's ID from Url Parameter

Test N°21 : Check Fetch API to get product's ID

Description : Check the fetchAPI function get a positive response from the API contains product's information.

Parameters : url + endpoint '/ID', method, body

Pass if returns : A JSON response contains product's information in relation to its ID.

Tests N°22 : Check Append item's information to product page

Description : Check the appendItemInfosInViewProduct function displays item's information in the product page.

Parameters : product

Pass if returns : list of item's information in product page.

Test N°23 : Check Cart sent to localStorage

Description : Check the sendCartToLocalStorage function sends array containing products to localStorage when user click on « Add to cart » button.

Parameters : object, localStorage.key, localStorage.value

Pass if returns : localStorage.key and localStorage.value != empty.

Test N°24 : Check success message

Description : Check the successMessage function adding success message in div container

Parameters : messageContainer, message

Pass if returns : true.

Test N°25 : Check appear success message on cart page

Description : Check the appearSuccessMessageAddToCart function adding to div « item-confirm » a confirmation message.

Parameters : messageContainer

Pass if returns : true.

Test N°26 : Check number of items in cart

Description : Check the numberOfItemsInCart function displaying number of products in cart on item menu « cart ».

Parameters : localStorageKey.length

Pass if returns : number in item menu « cart » == number of products in localStorage « cart ».

Test N°27 : Check get Cart from LocalStorage

Description : Check the getCartFromLocalStorage function get the content of localStorage and applies a new TeddyBear class forEach product.

Parameters : localStorage.key, TeddyBear's Class.

Pass if returns : array of TeddyBears.

Test N°28 : Check Append cart content in cart page

Description : Check the appendProductsInViewCart function displays list of products contained in the cart.

Parameters : array of TeddyBears.

Pass if returns : products list contained in cart to the cart page.

Test N°29 : Check addition item's total cost

Description : Check the appendCartTotalCostInViewCart function adding cost of each product and display the total cost in the cart page.

Parameters : itemsTotalPrice, itemTotalPriceNumber

Pass if returns : itemsTotalPrice > 0.

Test N°30 : Check delete product on cart

Description : Check the deleteItemOnCart function deleting a product in the cart page.

Parameters : array of TeddyBears.

Pass if returns : true.

Test N°31 : Check form is valid

Description : Check the postOrderToAPI function continues when the form is valid.

Parameters : object Contact

Pass if returns : the posrtOrderToAPI returns a 201 response.

Test N°32 : Check firstName in form is invalid

Description : Check the postOrderToAPI function displaying an error message when the firstName's input is invalid.

Parameters : firstName's input, nameRegex.

Pass if returns : an error message when the firstName's input is invalid.

Test N°33 : Check lastName in form is invalid

Description : Check the postOrderToAPI function displaying an error message when the lastName's input is invalid.

Parameters : lastName's input, nameRegex.

Pass if returns : an error message when the lastName's input is invalid.

Test N°34 : Check address in form is invalid

Description : Check the postOrderToAPI function displaying an error message when the address input is invalid.

Parameters : address input, addressRegex.

Pass if returns : an error message when the address input is invalid.

Test N°35 : Check city in form is invalid

Description : Check the postOrderToAPI function displaying an error message when the city input is invalid.

Parameters : city input, cityRegex.

Pass if returns : an error message when the city input is invalid.

Test N°36 : Check email in form is invalid

Description : Check the postOrderToAPI function displaying an error message when the email input is invalid.

Parameters : email input, emailRegex.

Pass if returns : an error message when the email input is invalid.

Test N°37 : Check post order to API

Description : Check the postOrderToAPI function sending the order correctly to the server.

Parameters : localStorage, order object

Pass if returns : a 201 response of the API containing products information and orderID.

Test N° 38 : Check orderId and totalCost appear

Description : Check the confirmedOrder function displaying the order id and total cost in the order page.

Parameters : orderId, totalCost

Pass if returns : true.

END OF UNIT TESTS.