const { test, expect } = require('@playwright/test');

//const { PassThrough } = require('stream');

var userid;

test("PL\_AS02-get user", async ({ request}) => {

  const response=await request.get('https://reqres.in/api/users?page=2')

  console.log(await response.json())

  expect(response.status()).toBe(200)

})

test("PL\_AS03-schema", async ({ request}) => {

  const response=await request.get('https://reqres.in/api/users/1')

  var body=await response.json()

  console.log(await response.json())

  expect(response.status()).toBe(200)

  var data1=body.data

  console.log("data",data1)

  console.log( typeof(data1))

 // var jboday=JSON.parse(data1)

  console.log(Array.isArray(data1)); // Should print true

  const isValidResponse = (

     typeof data1.id === 'number' &&

    typeof data1.email === 'string' &&

     typeof data1.first\_name === 'string' &&

      typeof data1.last\_name === 'string' &&

       typeof data1.avatar === 'string');

       console.log('Is Valid Response:', isValidResponse);

test("PL\_AS04-pass parameter", async ({ request}) => {

  const response=await request.get('https://reqres.in/api/users/1')

  console.log(await response.json())

  expect(response.status()).toBe(200)

})

test("PL\_AS05-query parameter", async ({ request}) => {

    const response=await request.get('https://reqres.in/api/users?id=2&first\_name=Michael')

    console.log(await response.json())

    expect(response.status()).toBe(200)

  })

})

 test("post user", async ({ request}) => {

   const response=await request.post('https://reqres.in/api/users',

   { data:{"name": "poornima","job": "leader"},

       headers:{ "Accept":"application/json" }

   })

   console.log(await response.json())

  expect(response.status()).toBe(201)

  var res=await response.json()

  //userid=res.id

 })

test("PL\_AS06-POST response validation", async ({ request}) => {

   const response=await request.post('https://reqres.in/api/users',

   { data:{"name": "poornima","job": "leader"},

       headers:{ "Accept":"application/json" }

   })

   console.log(await response.json())

 console.log("Response code ",response.status())

  expect(response.status()).toBe(201)

  var res=await response.json()

  userid=res.id

  let name =res.name

 await expect(res.name).toBe("poornima")

 console.log("full moon  transaltion equal to  ",name)

})

test("PL\_AS07-PUT body validation", async ({ request }) => {

  const response = await request.post('https://reqres.in/api/users', {

    data: {

      "name": "poornima",

      "job": "trainer"

    },

    headers: {

      "Accept": "application/json"

    }

  });

  console.log(await response.json());

  expect(response.status()).toBe(201);

  var res = await response.json();

  let job = res.job;

  await expect(job).toBe("trainer");

  let name = res.name;

  await expect(name).toBe("poornima");

  console.log("Body validation done");

});

test("PL\_AS08-post response datatype validation", async ({ request }) => {

  const response = await request.get('https://reqres.in/api/users/' + userid, {

    headers: { "Accept": "application/json" }

  });

  console.log(await response.json());

  console.log("Response code ", response.status());

  expect(response.status()).toBe(200);

  var body = await response.json();

  console.log("data", body);

  var data1 = body.data;

  console.log("data", data1);

  console.log(typeof(data1));

  console.log(Array.isArray(data1));

  // Check if 'data1' is an object before validating the 'job' property.

  const isValidResponse = typeof data1 === 'object' && typeof data1.job === 'string';

  console.log('Is Valid Response:', isValidResponse);

});

test("PL\_AS09-Validation of Data created", async ({ request }) => {

  const response = await request.post('https://reqres.in/api/users', {

    data: {

      "name": "poornima",

      "job": "leader"

    },

    headers: {

      "Accept": "application/json"

    }

  });

  const responseBody = await response.json();

  console.log(responseBody);

  // Validate the response status code.

  expect(response.status()).toBe(201);

  // Validate the created data against your expectations.

  expect(responseBody.name).toBe("poornima");

  expect(responseBody.job).toBe("leader");

  // Add more validation checks as needed.

});

test("PL\_AS10-PUT validation", async ({ request }) => {

  // Ensure 'userid' is set to a valid user ID.

  userid = 1;

  const response = await request.put('https://reqres.in/api/users/' + userid, {

      data: {"name": "poornima", "job": "trainer"},

      headers: {"Accept": "application/json"}

  });

  const responseBody = await response.json();

  console.log(responseBody);

  expect(response.status()).toBe(200);

  // Validate the response data

  expect(responseBody.job).toBe("trainer");

  console.log("Validation done");

});

test("PL\_AS11 -PUT body validation", async ({ request }) => {

const updateData = {

  job: "leader", // Replace with the actual ID

  updatedJob: 'trainer',

};

// Make a PUT request to the API endpoint using Playwright's request function

const response = await request.put('https://reqres.in/api/users/2', {

  data: JSON.stringify(updateData),

  headers: {

    'Content-Type': 'application/json',

  },

});

const responseBody = await response.json();

const updatedJob = responseBody.updatedJob;

expect(typeof updatedJob).toBe('string'); // You can replace 'string' with the expected data type

});

test("PL\_AS12 -PUT datatype validation", async ({ request }) => {

  // Ensure 'userid' is set to a valid user ID.

  userid = 1; // Replace with the appropriate user ID.

  const response = await request.get('https://reqres.in/api/users/' + userid, {

      data: {"name": "poornima", "job": "trainer"},

      headers: {"Accept": "application/json"}

  });

  const responseText = await response.text();

  if (typeof responseText === 'string') {

      console.log('The response content is a string.');

  } else {

      console.error('The response content is not a string.');

  }

});

test("PL\_AS13-PATCH validation", async ({ request }) => {

  // Ensure 'userid' is set to a valid user ID.

  userid = 1; // Replace with the appropriate user ID.

  const response = await request.patch('https://reqres.in/api/users/' + userid, {

      data: {"name": "poornima hugar", "job": "trainer"},

      headers: {"Accept": "application/json"}

  });

  const responseBody = await response.json();

  console.log(responseBody);

  expect(response.status()).toBe(200);

  // Validate the response data

  expect(responseBody.name).toBe("poornima hugar");

  console.log("PATCH validation done");

});

test("PL\_AS14 -PATCH body validation", async ({ request }) => {

  const response = await request.patch('https://reqres.in/api/users/' + userid, {

    data: {

      "name": "poornima hugar",

      "job": "trainer"

    },

    headers: {

      "Accept": "application/json"

    }

  });

  console.log(await response.json());

  expect(response.status()).toBe(200);

  var res = await response.json();

  let { name, job } = res;

  // Validate the updated data against your expectations.

  await expect(name).toBe("poornima hugar");

  await expect(job).toBe("trainer");

  console.log("PATCH body validation done");

});

test("PL\_AS15-PATCH datatype validation", async ({ request}) => {

  userid = 1; // Replace with the appropriate user ID.

  const response = await request.put('https://reqres.in/api/users/' + userid, {

      data: {"name": "poornima hugar", "job": "trainer"},

      headers: {"Accept": "application/json"}

  });

  const responseText = await response.text();

  if (typeof responseText === 'string') {

      console.log('The response content is a string.');

  } else {

      console.error('The response content is not a string.');

  }

});

test("PL\_AS16-Validation of Data created", async ({ request }) => {

// Ensure 'userid' is set to a valid user ID.

userid = 1; // Replace with the appropriate user ID.

const response = await request.patch('https://reqres.in/api/users/' + userid, {

  data: {

    "name": "poornima hugar",

    "job": "trainer"

  },

  headers: {

    "Accept": "application/json"

  }

});

const responseBody = await response.json();

console.log(responseBody);

// Validate the response status code.

expect(response.status()).toBe(200);

// Validate the updated data against your expectations.

expect(responseBody.name).toBe("poornima hugar");

expect(responseBody.job).toBe("trainer");

// Add more validation checks as needed.

});