**b545fbbd3bf34b167c67b05f64e4a92a --- API Key for metals**

https://api.metalpriceapi.com/v1/**latest**

?api\_key=b545fbbd3bf34b167c67b05f64e4a92a

&base=**USD**

&currencies=**EUR,XAU,XAG**

**Using Fetch to fetch live silver price in Javascript**

**var** requestOptions = {

method: 'GET',

redirect: 'follow'

};

**fetch**("https://api.metalpriceapi.com/v1/latest?api\_key=REPLACE\_ME&base=USD&currencies=XAU", requestOptions)

.**then**(response => response.**text**())

.**then**(result => console.**log**(result))

.**catch**(error => console.**log**('error', error));

**Instructions**

1. Get a free API key to use by signing up at [Metalpriceapi.com](https://metalpriceapi.com/) and replace REPLACE\_ME with your API key.
2. Update base value to a currency or remove this query param.
3. Update currencies value to a list of values or remove this query param. (XAG)

For 2 and 3, you can find this documented at [Offical Documentation](https://metalpriceapi.com/documentation)

Here's how you can fetch the live silver price (XAG) from the API and display it in an HTML page:

**1. HTML Structure**

Create an HTML file (index.html) with a placeholder for the silver price:

html

CopyEdit

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Live Silver Price</title>

</head>

<body>

<h1>Live Silver Price (USD per XAG)</h1>

<p id="silver-price">Loading...</p>

<script>

var requestOptions = {

method: 'GET',

redirect: 'follow'

};

fetch("https://api.metalpriceapi.com/v1/latest?api\_key=REPLACE\_ME&base=USD&currencies=XAG", requestOptions)

.then(response => response.json()) // Convert response to JSON

.then(data => {

if (data.rates && data.rates.XAG) {

document.getElementById("silver-price").innerText = `$${data.rates.XAG.toFixed(2)} per XAG`;

} else {

document.getElementById("silver-price").innerText = "Error fetching price";

}

})

.catch(error => {

console.error('Error:', error);

document.getElementById("silver-price").innerText = "Failed to load price";

});

</script>

</body>

</html>

**2. Replace API Key**

* Replace REPLACE\_ME with your actual API key.

**3. Explanation**

* Fetches data from the API.
* Converts the response to JSON.
* Extracts the silver (XAG) price.
* Updates the HTML <p> tag with the silver price.
* Handles errors in case the API fails.

To fetch and display both **Gold (XAU) and Silver (XAG)** prices in your HTML page, update your code as follows:

**1. Updated HTML File (index.html)**

html

CopyEdit

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Live Gold & Silver Prices</title>

</head>

<body>

<h1>Live Metal Prices (USD)</h1>

<p><strong>Gold (XAU):</strong> <span id="gold-price">Loading...</span></p>

<p><strong>Silver (XAG):</strong> <span id="silver-price">Loading...</span></p>

<script>

var requestOptions = {

method: 'GET',

redirect: 'follow'

};

fetch("https://api.metalpriceapi.com/v1/latest?api\_key=REPLACE\_ME&base=USD&currencies=XAU,XAG", requestOptions)

.then(response => response.json()) // Convert response to JSON

.then(data => {

if (data.rates) {

document.getElementById("gold-price").innerText = `$${data.rates.XAU.toFixed(2)} per XAU`;

document.getElementById("silver-price").innerText = `$${data.rates.XAG.toFixed(2)} per XAG`;

} else {

document.getElementById("gold-price").innerText = "Error fetching price";

document.getElementById("silver-price").innerText = "Error fetching price";

}

})

.catch(error => {

console.error('Error:', error);

document.getElementById("gold-price").innerText = "Failed to load price";

document.getElementById("silver-price").innerText = "Failed to load price";

});

</script>

</body>

</html>

**2. Replace API Key**

* Replace REPLACE\_ME with your **actual API key**.

**3. Explanation**

* Fetches **Gold (XAU) and Silver (XAG)** prices in a single request.
* Converts the response to JSON.
* Updates the respective <span> elements with the price values.
* Handles errors properly.