

## Test cases for the **GET /simple/price** endpoint

### Test cases scenarios, steps and expected results:

Test Case ID	Test Scenario	Steps	Expected Result	Status (Pass/Fail)
TC001	Verify successful response with valid parameters	1. Send a GET request to `/simple/price` with `ids=bitcoin&vs_currencies=usd`. 2. Check the response status code. 3. Validate the response contains correct price data.	Status code = <b>200 OK</b>  Response contains `{ "bitcoin": { "usd": 50000 } }` (example).	<b>Passed</b>
TC002	Verify response with multiple cryptocurrencies and currencies	1. Send a GET request with `ids=bitcoin,ethereum&vs_currencies=usd,eur`. 2. Validate response contains data for both cryptocurrencies in both currencies.	Status code = <b>200 OK</b>  Response includes `{ "bitcoin": { "usd": 50000, "eur": 46000 }, "ethereum": { "usd": 3500, "eur": 3200 } }` (example).	<b>Passed</b>
TC003	Verify response when requesting an invalid cryptocurrency	1. Send a GET request with `ids=invalidcoin&vs_currencies=usd`. 2. Observe the API response.	Status code = <b>200 OK</b>  API returns `{}` (empty json object).	<b>Passed</b>
TC004	Verify response when requesting an invalid currency	1. Send a GET request with `ids=bitcoin&vs_currencies=invalidcurrency`. 2. Check the response format. 3. (variations with invalid ids could be done)	Status code = <b>200 OK</b>  API returns `{}` <u>if the id is unsupported</u> . If supported, the result is `{ "bitcoin": {} }`, for example.	<b>Passed</b>
TC005	Verify response when missing required parameters	1. Send a GET request without `ids` and `vs_currencies`. 2. Observe the response.	Status code = <b>422 Unprocessable Entity</b>  API returns an error message	<b>Passed</b>
TC006	Verify additional boolean parameters (`include_market_cap`, `include_24hr_vol`, `include_24hr_change`, `include_last_updated_at`)	1. Send a GET request with `include_market_cap=true&include_24hr_vol=true&include_24hr_change=true&include_last_updated_at=true`. 2. Validate the response includes additional fields.	Status code = <b>200 OK</b>  Response includes `market_cap`, `24h_vol`, `24h_change`, and `last_updated_at`.	<b>Passed</b>
TC007	Verify response precision parameter	1. Send a GET request with `precision=2`. 2. Validate the response precision points.	Status code = <b>200 OK</b>  Response numbers must have precision point '2'.	<b>Passed</b>

## Summary of the results:

## Public API Testing - Run results

TC001

Parameters:

GET

⌵

{{base\_url}} /simple/price?ids=bitcoin&vs\_currencies=usd

Params

Authorization

Headers (8)

Body

Scripts

Settings

Query Params

<input checked="" type="checkbox"/>	Key	Value
<input checked="" type="checkbox"/>	ids	bitcoin
<input checked="" type="checkbox"/>	vs_currencies	usd
	Key	Value

Script Tests:

Params

Authorization

Headers (8)

Body

Scripts

Settings

Pre-request

Post-response

```
1 pm.test("Status code is 200", function () {
2   pm.response.to.have.status(200);
3 });
4
5 pm.test("Response contains bitcoin price", function () {
6   var jsonData = pm.response.json();
7   pm.expect(jsonData).to.have.property('bitcoin');
8   pm.expect(jsonData.bitcoin).to.have.property('usd');
9   pm.expect(jsonData.bitcoin.usd).to.be.a('number');
10  });
```

Results:

Body

Cookies

Headers (28)

Test Results (2/2)

⌵

200 OK

1.48 s

1.08 KB

{ } JSON

⌵

▶ Preview

🔗 Visualize

⌵

```
1 {
2   "bitcoin": {
3     "usd": 84620
4   }
5 }
```

## TC002

Parameters:

GET ⌵ `{{base_url}} /simple/price?ids=bitcoin,ethereum&vs_currencies=usd,eur`

Params • Authorization Headers (8) Body Scripts • Settings

Query Params

<input checked="" type="checkbox"/>	Key	Value
<input checked="" type="checkbox"/>	ids	bitcoin,ethereum
<input checked="" type="checkbox"/>	vs_currencies	usd,eur
	Key	Value

Script Tests:

Params • Authorization Headers (8) Body Scripts • Settings

Pre-request

Post-response •

```
1 pm.test("Status code is 200", function () {
2   pm.response.to.have.status(200);
3 });
4 pm.test("Response contains bitcoin price", function () {
5   var jsonData = pm.response.json();
6   pm.expect(jsonData).to.have.property('bitcoin');
7   pm.expect(jsonData).to.have.property('ethereum');
8   pm.expect(jsonData.bitcoin).to.have.property('usd');
9   pm.expect(jsonData.bitcoin).to.have.property('eur');
10  // validate numbers
11  pm.expect(jsonData.bitcoin.usd).to.be.a('number');
12  pm.expect(jsonData.bitcoin.eur).to.be.a('number');
13  pm.expect(jsonData.ethereum.usd).to.be.a('number');
14  pm.expect(jsonData.ethereum.eur).to.be.a('number');
15 });
```

Results:

Body Cookies Headers (27) Test Results (2/2) 🔄 200 OK • 719 ms • 1.13 KB •

{} JSON ⏮ Preview 🔗 Visualize ⌵

```
1 {
2   "bitcoin": {
3     "usd": 84629,
4     "eur": 80745
5   },
6   "ethereum": {
7     "usd": 2342.33,
8     "eur": 2234.83
9   }
10 }
```

## TC003

Parameters:

GET ⌵ {{base\_url}} /simple/price?ids=invalidcoin&vs\_currencies=usd

Params ● Authorization Headers (8) Body Scripts ● Settings

Query Params

<input checked="" type="checkbox"/>	Key	Value
<input checked="" type="checkbox"/>	ids	invalidcoin
<input checked="" type="checkbox"/>	vs_currencies	usd
	Key	Value

Script Tests:

Params ● Authorization Headers (8) Body Scripts ● Settings

Pre-request

Post-response ●

```
1 pm.test("Status code is 200", function () {
2   pm.response.to.have.status(200);
3 });
4
5 pm.test("Response should be empty", function () {
6   var jsonData = pm.response.json();
7   pm.expect(jsonData).to.be.an('object');
8 });
```

Results:

Body Cookies Headers (28) Test Results (2/2) ↺ 200 OK • 217 ms • 1.06 KB 🌐

{} JSON ⌵ ▶ Preview 🔗 Visualize ⌵

1 {}

## TC004

Parameters:

GET

{{base\_url}} /simple/price?ids=bitcoin&vs\_currencies=invalidcurrency

Params

Authorization

Headers (8)

Body

Scripts

Settings

Query Params

<input checked="" type="checkbox"/>	Key	Value
<input checked="" type="checkbox"/>	ids	bitcoin
<input checked="" type="checkbox"/>	vs_currencies	invalidcurrency
	Key	Value

Script Tests:

Params

Authorization

Headers (8)

Body

Scripts

Settings

Pre-request

Post-response

```
1 pm.test("Status code is 200", function () {
2   pm.response.to.have.status(200);
3 });
4
5 pm.test("Response should be empty", function () {
6   var jsonData = pm.response.json();
7   pm.expect(jsonData).to.be.an('object');
8   // for both invalid this assert must be gone
9   pm.expect(jsonData).to.have.property('bitcoin');
10  });
```

Results:

Body

Cookies

Headers (27)

Test Results (2/2)

200 OK

{ } JSON

Preview

Visualize

```
1 {
2   "bitcoin": {}
3 }
```

TC005

Parameters:

GET

▼

{{base\_url}} /simple/price

ParamsAuthorizationHeaders (8)BodyScripts ●Settings

Query Params

	Key	Value
	Key	Value

Script Tests:

ParamsAuthorizationHeaders (8)BodyScripts ●Settings

Pre-request

Post-response •

```
1 pm.test("Should return an error or empty response", function () {
2   pm.response.to.have.status(422);
3 });
4
5 pm.test("Validate error message from ids", function () {
6   var jsonData = pm.response.json();
7   pm.expect(jsonData).to.have.property('error');
8   pm.expect(jsonData.error).to.eql('Missing parameter ids');
9 });
```

Results:

BodyCookiesHeaders (20)Test Results (2/2)↺422 Unprocessable Entity (WebDAV) (RFC 4918) • 439 ms • 751 B • 🌐

{ } JSON ▼▶ Preview🔗 Visualize ▼

```
1 {
2   "error": "Missing parameter ids"
3 }
```

Parameters:

GET	/simple/price?ids=bitcoin&vs_currencies=usd&include_market_cap=true&include_24hr_vol=true&include_24hr_change=true&include_last_updated_at=true	
Params	Authorization	Headers (8)
Query Params		
<input checked="" type="checkbox"/>	Key	Value
<input checked="" type="checkbox"/>	ids	bitcoin
<input checked="" type="checkbox"/>	vs_currencies	usd
<input checked="" type="checkbox"/>	include_market_cap	true
<input checked="" type="checkbox"/>	include_24hr_vol	true
<input checked="" type="checkbox"/>	include_24hr_change	true
<input checked="" type="checkbox"/>	include_last_updated_at	true

Script Tests:

```
Params • Authorization Headers (8) Body Scripts • Settings
Pre-request
Post-response •
1 pm.test("Status code is 200", function () {
2   pm.response.to.have.status(200);
3 });
4
5 pm.test("Response contains all the fields", function () {
6   var jsonData = pm.response.json();
7   pm.expect(jsonData).to.have.property('bitcoin');
8   // usd
9   pm.expect(jsonData.bitcoin).to.have.property('usd');
10  pm.expect(jsonData.bitcoin.usd).to.be.a('number');
11  // usd_market_cap
12  pm.expect(jsonData.bitcoin).to.have.property('usd_market_cap');
13  pm.expect(jsonData.bitcoin.usd).to.be.a('number');
14  // usd_24h_vol
15  pm.expect(jsonData.bitcoin).to.have.property('usd_24h_vol');
16  pm.expect(jsonData.bitcoin.usd).to.be.a('number');
17  // usd_24h_change
18  pm.expect(jsonData.bitcoin).to.have.property('usd_24h_change');
19  pm.expect(jsonData.bitcoin.usd).to.be.a('number');
20  // last_updated_at
21  pm.expect(jsonData.bitcoin).to.have.property('last_updated_at');
22  pm.expect(jsonData.bitcoin.usd).to.be.a('number');
23 });
```

Results:

Body Cookies Headers (27) Test Results (2/2) 200 OK • 415 ms • 1.2 KB

{ } JSON Preview Visualize

```
1 {
2   "bitcoin": {
3     "usd": 84785,
4     "usd_market_cap": 1676583424195.7297,
5     "usd_24h_vol": 76019485942.08984,
6     "usd_24h_change": -3.8206890993195386,
7     "last_updated_at": 1740619479
8   }
9 }
```



TC007

Parameters:

GET

▼

{{base\_url}} /simple/price?ids=bitcoin&vs\_currencies=usd&precision=2

Params ●

Authorization

Headers (8)

Body

Scripts ●

Settings

Query Params

<input checked="" type="checkbox"/>	Key	Value
<input checked="" type="checkbox"/>	ids	bitcoin
<input checked="" type="checkbox"/>	vs_currencies	usd
<input checked="" type="checkbox"/>	precision	2
	Key	Value

Script Tests:

Params ●

Authorization

Headers (8)

Body

Scripts ●

Settings

Pre-request

Post-response ●

```
1 pm.test("Status code is 200", function () {
2   pm.response.to.have.status(200);
3 });
4
5 pm.test("Response contains bitcoin price", function () {
6   var jsonData = pm.response.json();
7   pm.expect(jsonData).to.have.property('bitcoin');
8   // usd
9   pm.expect(jsonData.bitcoin).to.have.property('usd');
10  pm.expect(jsonData.bitcoin.usd).to.be.a('number');
11  //validate precision point
12  pm.expect(String(jsonData.bitcoin.usd).split(".")[1]?.length == 2).to.be.true;
13 });
```

Results:

Body

Cookies

Headers (27)

Test Results (2/2)

🔄

200 OK

• 737 ms

• 1.07 KB

🌐

{ } JSON ▼

▶ Preview

🔗 Visualize ▼

```
1 {
2   "bitcoin": {
3     "usd": 84962.22
4   }
5 }
```