



# Getting Started

---

The SpeedOf.Me API allows you to embed an internet speed test directly into your website or web application. The API uses a JavaScript interface that handles all the complexity of running speed tests.

## Quick Start

Follow these three steps to integrate the SpeedOf.Me speed test:

### 1. Include the API Script

Add the SpeedOf.Me API script to your HTML page:

HTML

Copy

```
<script src="//speedof.me/api/api.js"></script>
```

**⚠ Important:** Do not download and host the API script yourself. Always load it from [speedof.me](https://speedof.me) to ensure you have the latest version.

### 2. Configure the API

Set your API key and registered domain:

JavaScript

Copy

```
SomApi.account = "YOUR_API_KEY"; // Your API key from the portal  
SomApi.domainName = "yourdomain.com"; // Your registered domain
```

### 3. Handle Results

Set up callbacks to receive test results:

JavaScript

Copy

```
SomApi.onTestCompleted = function(result) {
  console.log("Download: " + result.download + " Mbps");
  console.log("Upload: " + result.upload + " Mbps");
  console.log("Latency: " + result.latency + " ms");
};

SomApi.onError = function(error) {
  console.error("Error: " + error.message);
};

// Start the test
SomApi.startTest();
```

💡 **Tip:** Get your API key from the [API Portal](#). Free trials are available for testing.

## Configuration

Configure the API behavior using the `SomApi` global object.

### SomApi.account

**SomApi.account** `string` Required

Your unique API key provided when you register. Format: `SOMxxxxxxxxxx`

### SomApi.domainName

**SomApi.domainName** `string` Required

The domain where your speed test is hosted. Supports wildcards: `*.example.com`

⚠️ **Important:** The domain in your [account Settings](#), the `SomApi.domainName` in your JavaScript code, and the actual website domain must all match.

## SomApi.config

Fine-tune the speed test behavior with these configuration options:

Property	Type	Default	Description
<code>sustainTime</code>	number	4	Controls how long to sustain each download sample (1-8 seconds). Lower values (1-2) = faster but less accurate tests. Higher values (6-8) = slower but more accurate results. The actual download test takes 1-10s at <code>sustainTime=1</code> , or 8-87s at <code>sustainTime=8</code> . Upload time is also affected since its sample size depends on the last download.
<code>uploadTestEnabled</code>	boolean	true	Enable or disable the upload speed test.
<code>latencyTestEnabled</code>	boolean	true	Enable or disable latency (ping) and jitter measurement.
<code>testServerEnabled</code>	boolean	true	Enable or disable detection of the CDN test server location.
<code>userInfoEnabled</code>	boolean	true	Enable or disable fetching user IP address and hostname.
<code>progress.enabled</code>	boolean	true	Enable or disable progress callbacks during the test.
<code>progress.verbose</code>	boolean	false	Enable verbose mode for more frequent progress updates with current speed.
<code>maxTestPass</code>	number	0 (auto)	Maximum test passes (8-11). Controls data usage. 0 = auto-detect based on speed.

### Data Usage

Data consumption depends on connection speed. Use `maxTestPass` to limit bandwidth consumption per test:

maxTestPass	Largest Sample	Max Bandwidth per Test
0 (default)	No restrictions	256 MB
8	16 MB	32 MB
9	32 MB	64 MB
10	64 MB	128 MB
11	128 MB	256 MB

💡 **Tip:** To limit data usage, set `SomApi.config.maxTestPass = 8` (max 32 MB) or disable upload with `SomApi.config.uploadTestEnabled = false`.

## Localhost Development

Register `localhost` in your API account, then use it in your JavaScript code:

JavaScript

Copy

```
SomApi.domainName = "localhost";
```

If you are using a port other than 80, include the port in your JavaScript code:

JavaScript

Copy

```
SomApi.domainName = "localhost:3000";
```

## Example Configuration

JavaScript

Copy

```
// Configure the API
SomApi.account = "SOM_YOUR_API_KEY";
SomApi.domainName = "/*.example.com";

// Customize test behavior
SomApi.config.sustainTime = 6; // Longer test for accuracy
```

```
SomApi.config.uploadTestEnabled = true;
SomApi.config.latencyTestEnabled = true;
SomApi.config.progress.enabled = true;
SomApi.config.progress.verbose = true; // Get real-time speed updates
```

## Callbacks

---

The API uses callback functions to communicate test progress and results.

### onTestCompleted

Called when the speed test completes successfully. Receives a result object with all test metrics.

JavaScript

 Copy

```
SomApi.onTestCompleted = function(result) {
  // Speed results
  console.log("Download Speed: " + result.download + " Mbps");
  console.log("Upload Speed: " + result.upload + " Mbps");
  console.log("Max Download: " + result.maxDownload + " Mbps");
  console.log("Max Upload: " + result.maxUpload + " Mbps");

  // Latency results
  console.log("Latency: " + result.latency + " ms");
  console.log("Jitter: " + result.jitter + " ms");

  // Additional info
  console.log("Test Server: " + result.testServer);
  console.log("IP Address: " + result.ip_address);
  console.log("Hostname: " + result.hostname);
  console.log("Test Date: " + result.testDate);
```

### onProgress

Called during the test to report progress. Enable with `SomApi.config.progress.enabled = true`.

JavaScript

 Copy

```
SomApi.onProgress = function(progress) {
  console.log("Phase: " + progress.type);           // "download", "upload", or "late
  console.log("Pass: " + progress.pass);           // Current pass number
  console.log("Progress: " + progress.percentDone + "%");

  // Available in verbose mode
  if (progress.currentSpeed) {
    console.log("Current Speed: " + progress.currentSpeed + " Mbps");
  }

  // Status message
  console.log("Status: " + progress.message);
};
```

## onError

Called if an error occurs during the test.

JavaScript

 Copy

```
SomApi.onError = function(error) {
  console.error("Error Code: " + error.code);
  console.error("Error Message: " + error.message);

  // Handle specific errors
  switch(error.code) {
    case 1001:
      console.log("Invalid API key");
      break;
    case 1002:
      console.log("Domain not registered");
      break;
    default:
      console.log("Test failed");
  }
};
```

## Result Objects

---

## Test Result Object

The `onTestCompleted` callback receives an object with these properties:

Property	Type	Description
<code>download</code>	number	Average download speed in Mbps
<code>upload</code>	number	Average upload speed in Mbps
<code>maxDownload</code>	number	Peak download speed in Mbps
<code>maxUpload</code>	number	Peak upload speed in Mbps
<code>latency</code>	number	Minimum latency (ping) in milliseconds
<code>jitter</code>	number	Latency variance in milliseconds
<code>testServer</code>	string	Test server location (e.g., "Los Angeles 1", "New York 2") or "Unknown"
<code>ip_address</code>	string	Client's public IP address
<code>hostname</code>	string	Reverse DNS hostname (if available)
<code>userAgent</code>	string	Browser user agent string
<code>testDate</code>	string	ISO 8601 timestamp of test completion

## Progress Object

The `onProgress` callback receives an object with these properties:

Property	Type	Description
<code>type</code>	string	Test phase: "download", "upload", or "latency"
<code>pass</code>	number	Current pass number (1-based). Empty for latency phase.
<code>percentDone</code>	number	Percentage complete (0-100)
<code>currentSpeed</code>	number	Current speed in Mbps (verbose mode only)

Property	Type	Description
<code>maxSpeed</code>	number	Peak speed so far in Mbps (verbose mode only)

## Error Codes

---

### API Validation Errors

These errors occur during API key and domain validation:

Code	Message	Cause
<code>1000</code>	Valid	API key and domain match - test proceeds normally
<code>1001</code>	Invalid account	API key not found or account is inactive
<code>1002</code>	Domain mismatch	The page URL doesn't match any registered domains

### Speed Test Errors

These errors occur during the actual speed test:

Code	Message	Cause
<code>2001</code>	Test error	Download or upload test failed
<code>2002</code>	Invalid server response	JSON parsing error from CDN server
<code>2003</code>	Request timeout	Network timeout during test

**ⓘ Troubleshooting:** If you receive error 1002, verify that your `SomApi.domainName` matches one of the domains registered in your [API Portal](#) account.

## Advanced Topics

---

## Mobile App Integration

The SpeedOf.Me API is JavaScript-based and requires a web browser environment. To integrate with native mobile apps:

1. Host your speed test HTML page on a domain registered with your API account (e.g., `yourdomain.com/speedtest.html` )
2. Load that page in a WebView (iOS or Android) within your app
3. Use JavaScript bridges to communicate results back to your native code

### iOS (WKWebView)

Use WKWebView with a JavaScript message handler to receive results:

Swift

 Copy

```
// In your WKWebView configuration
let config = WKWebViewConfiguration()
config.userContentController.add(self, name: "speedTestHandler")

// JavaScript in the WebView calls:
// window.webkit.messageHandlers.speedTestHandler.postMessage(result);
```

 **Important:** UIWebView is deprecated since iOS 12. Always use WKWebView.

### Android (WebView)

Use `addJavascriptInterface` to bridge JavaScript and native code:

Java

 Copy

```
// Enable JavaScript
webView.getSettings().setJavaScriptEnabled(true);

// Add interface for JavaScript to call native code
webView.addJavascriptInterface(new SpeedTestInterface(), "Android");

// In JavaScript:
// Android.onTestCompleted(JSON.stringify(result));
```

ⓘ **Security:** Only expose necessary methods via `@JavascriptInterface` annotation (required for API 17+).

## Examples

### Basic Sample

basic.html

Copy

```
<!DOCTYPE html>
<html>
<head>
  <script src="//speedof.me/api/api.js"></script>
</head>
<body>
  <button onclick="startTest()">Start Speed Test</button>
  <div id="results"></div>

  <script>
    // Configure the API
    SomApi.account = "SOM_YOUR_API_KEY"; // Replace with your API key
    SomApi.domainName = "yourdomain.com"; // Replace with your domain
    SomApi.config.sustainTime = 4; // Test duration: 1-8 seconds

    // Handle results
```

### More Examples

Ready-to-use integrations for React, Vue, Angular, iOS, Android, Flutter, and more.

[Browse All Examples](#) →

### Quick Links

- [Basic Sample](#) - Minimal implementation
- [Advanced Sample](#) - With progress updates and configuration options

Need help? Visit the [API Portal](#) or contact support.

© 2026 SpeedOf.Me