

Storage Manager Overview (Speicherverwaltung-Übersicht)

Kurzübersicht

Der WalkMe Player speichert Daten zur Wartung des WalkMe Player-Status, zur Überwachung der Kompilierung der Walk-Thru-Wiedergabe und zur Erfassung von Analysen. Zu diesem Zweck verwendet WalkMe Erstanbieter-Cookies, Drittanbieter-Cookies (lokale Speicherung) und lokale Erstanbieter-Speicher. Es werden keine personenbezogenen Daten gespeichert und keine Informationen über die Nutzung der Website durch die Benutzer überwacht.

Von WalkMe gesetzte Cookies enthalten ein vorangestelltes „wm-“ in ihrem Namen. WalkMe verwendet je nach Ihrer WalkMe-Konfiguration unterschiedliche Cookies. So können wir z. B. Benutzerinformationen über die Vollendung von Walk-Thrus mithilfe von Cookies oder lokaler Speicherung verfolgen.

Sie möchten Ihre eigenen Cookies mit WalkMe erstellen? [Sehen Sie sich diesen Artikel zu WalkMe Data an.](#)

WalkMe speichert Daten auf dem Browser des Endbenutzers aus verschiedenen Gründen, unter anderem:

1. Smart Walk-Thru-Status (ob der Endbenutzer sich inmitten eines Smart Walk-Thrus befindet)
2. Autoplay-Status (ob ein Deployable bereits abgespielt wurde)
3. Zielerreichung
4. WalkMe-Daten

Anmerkung

Die Benutzererweiterung erfordert keine Cookies von Drittanbietern.

Definitionen

1. **Speichermethode:** So wird der Speicher auf dem Browser gespeichert, ob Erstanbieter (gleiche Domain) oder Drittanbieter (domainübergreifend).
2. **Speichertyp:** Hier wird der Speicher im Browser gespeichert (lokaler Speicher vs. Cookies)

3. **Browser-Speicher vs. Server-Speicher:** Ob die Daten nur im aktuellen Browser des Endbenutzers bzw. der Endbenutzerin gespeichert werden oder auch an die Server von WalkMe gesendet werden (mehr dazu unten).

Was ist ein Serverspeicher?

Der Serverspeicher synchronisiert den Endbenutzerstatus zwischen verschiedenen Browsern über den WalkMe Player-Server.

Der folgende Endbenutzerstatus wird im Serverspeicher gespeichert und mit den verschiedenen Browsern synchronisiert:

1. Autoplay nach Auswertungen
2. Zielerreichung
3. TeachMe-Kursabschluss
4. WalkMe-Daten

Hinweis

Beachten Sie Folgendes in Bezug auf Serverspeicher:

- Es erfordert die Definition einer Endbenutzeridentifizierung und funktioniert nicht mit der von WalkMe generierten Endbenutzer-ID.
- Es erfordert, dass die Endbenutzer-ID im Client bei der Laufzeit gefunden werden kann.
- Es erfordert, dass die Endbenutzer-ID weniger als 60 Zeichen aufweist.
- Die Daten werden periodisch alle 24 Stunden synchronisiert (kann von F & E konfiguriert werden). Das bedeutet, wenn Sie bereits einen Status im Client haben und der Status in einem anderen Browser geändert wurde, dauert es ein paar Stunden, bis er aktualisiert wird.

Extension Storage

Der Erweiterungsspeicher soll den Bereich der WalkMe-Speicheroptionen erweitern und schließlich zur Vorgabe-Speichereinstellung für Implementierungen mit einer Erweiterung werden. WalkMe-bezogene Daten werden sicher in den Einstellungen der Erweiterung gespeichert und erfordern keine Anpassung der Browser-Sicherheitseinstellungen.

Der Erweiterungsspeicher ist kompatibel mit Chrome, Firefox, Edge Chromium.

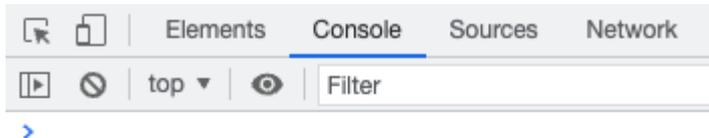
Storage API

Die Storage API ist eine neue browserbasierte API, die eine einfache und einheitliche Oberfläche für die Verwaltung verschiedener Speichertypen bietet. Damit können Sie Daten effizienter und selektiver verwalten, ohne möglicherweise wichtige Informationen zu verlieren oder sich erneut anmelden zu müssen.

Es ersetzt die Notwendigkeit, den Browser-Cache und die Cookies für das Testen oder die Fehlerbehebung von WalkMe-Inhalten zu löschen, die auf Browser-Daten beruhen. Das Löschen des Cache und der Cookies war eine gängige Methode, um die WalkMe-bezogenen Daten auf lokaler Ebene zurückzusetzen, aber es löscht auch alle gespeicherten Daten und erfordert, dass sich der Benutzer wieder bei der Website anmeldet.

Darüber hinaus ist die Storage API ein nützlicher Bestandteil der schrittweisen Einführung des Erweiterungsspeichers von WalkMe. Mit dem Extension Storage ist das Löschen des Browser-Cache und der Cookies keine Option. Verwenden Sie stattdessen die Storage API, wenn Sie WalkMe-Daten löschen müssen.

Die API ist in der Konsole der Browser Developer Tools verfügbar. Um auf die Konsole zuzugreifen, klicken Sie mit der rechten Maustaste auf die Webseite und wählen Sie **„Inspect“**. Klicken Sie dann auf die Registerkarte „Console“.



Hier ist die Liste der verfügbaren Befehle:

WalkMeAPI.storage.getAll()

- Gibt alle lokal im aktuellen Browser gespeicherten WalkMe-Daten zurück.
- Mit diesem Befehl können Sie alle bestehenden Daten sowie jene Daten anzeigen, die Sie mit einem der folgenden Schlüssel festgelegt haben.

```
WalkMeAPI.storage.getAll()
▼ {wm-ds-b: '[]', wm-ds-lb: '{}', wm-cseu-id: '6d7e1d59-ff6e-4867-b072-efba9460d67b', wm-ds-s: '[]', wm-ueug: '6478b0d6-2ee9-4837-8bb0-6a328f292274', ...} ⓘ
  wm-cseu-id: "6d7e1d59-ff6e-4867-b072-efba9460d67b"
  wm-ds-b: "[]"
  wm-ds-hb: "[]"
  wm-ds-lb: "{}"
  wm-ds-lbb: "{}"
  wm-ds-lbp: "[]"
  wm-ds-lfb: "{}"
  wm-ds-s: "[]"
  ▶ wm-hb: {sendBaseTime: 1681028422879}
  wm-lnchr-ply-ssn: "a17d688b-26eb-4907-8b6f-92705ed9d699"
  ▶ wm-session-per-user: {bf7ed7d7-4ec5-45f9-aa72-f8024a2c145e: {...}, 6478b0d6-2ee9-4837-8bb0-6a328f292274: {...}}
  ▶ wm-smtp-init: {type: 6}
  wm-ueug: "6478b0d6-2ee9-4837-8bb0-6a328f292274"
  wm-wmv: "6478b0d6-2ee9-4837-8bb0-6a328f292274"
  ▶ [[Prototype]]: Object
```

WalkMeAPI.storage.removeAll()

- Entfernt alle lokal im aktuellen Browser gespeicherten WalkMe-Daten.
- Die standardmäßigen WalkMe-Datenschlüssel werden nach einer Aktualisierung der Seite zurückgesetzt.

WalkMeAPI.storage.setItem()

- Verwenden Sie dies zum Erstellen eines benutzerdefinierten Datenschlüssels für Testzwecke.
- Akzeptiert zwei Pflichtfelder und ein optionales:
- **Key**: String-Name des Schlüssels, den Sie dem Speicher hinzufügen möchten. Wir empfehlen, „wm-“ standardmäßig als Präfix hinzuzufügen, um sicherzustellen, dass der Schlüssel einzigartig ist.
- **Value** : Der Wert des Schlüssels, den Sie hinzufügen möchten; kann eine Zeichenfolge oder eine Nummer sein.
- **(optional) Expiry time in seconds**: Anzahl der Sekunden, für die der Schlüssel in Ihrem lokalen Speicher existieren soll. Wenn nicht anders angegeben, wird ein Standardwert von zwei Jahren zugewiesen.
- Beispiele:
 - WalkMeAPI.storage.setItem('test-1',123) legt einen Schlüssel mit dem Namen 'wm-test-1' und einem Wert von 123 für zwei Jahre fest. Wenn Sie diesen Schlüssel festlegen und WalkMeAPI.storage.getAll() ausführen, erhalten Sie das folgende Ergebnis:
- WalkMeAPI.storage.setItem('test-2','Test value', 360) legt einen Schlüssel mit dem Namen 'wm-test-2' und dem Wert 'Test value' für 360 Sekunden fest.

```
> WalkMeAPI.storage.getAll()
< {wm-ds-b: '[]', wm-ds-lb: '{}', wm-cseu-id: '6d7e1d59-ff6e-4867-b072-efba9460d67b', wm-test-2: 'Test value', wm-ds-s: '[]', ...}
  wm-cseu-id: "6d7e1d59-ff6e-4867-b072-efba9460d67b"
  wm-ds-b: "[]"
  wm-ds-hb: "[]"
  wm-ds-lb: "{}"
  wm-ds-lbb: "{}"
  wm-ds-lbp: "[]"
  wm-ds-lfb: "{}"
  wm-ds-s: "[]"
  ▶ wm-hb: {sendBaseTime: 1681028422879}
  wm-lnchr-ply-ssn: "a17d688b-26eb-4907-8b6f-92705ed9d699"
  ▶ wm-session-per-user: {bf7ed7d7-4ec5-45f9-aa72-f8024a2c145e: {...}, 6478b0d6-2ee9-4837-8bb0-6a328f292274: {...}}
  ▶ wm-smtp-init: {type: 6}
  wm-test-1: 123
  wm-test-2: "Test value"
  wm-ueug: "6478b0d6-2ee9-4837-8bb0-6a328f292274"
  wm-wmv: "6478b0d6-2ee9-4837-8bb0-6a328f292274"
  ▶ [[Prototype]]: Object
```

WalkMeAPI.storage.getExpiryDate()

- Gibt das Datum und die Uhrzeit zurück, zu der der Datenschlüssel abläuft.
- Akzeptiert den Schlüsselnamen
- Beispiel:
 - `WalkMeAPI.storage.getExpiryDate('wm-test-2')`

```
> WalkMeAPI.storage.getExpiryDate('wm-test-2')
< 'Sun Apr 09 2023 11:29:27'
```

Wenden Sie sich bitte an Ihren WalkMe Kontakt oder [an den Support](#), wenn Sie Fragen haben oder Hilfe bei Ihren Speichereinstellungen benötigen.