عنوان: **WebStorage**
نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۴/۰۴/۰۸ ۱۱:۵۰
آدرس: www.dotnettips.info
گروه‌ها: Cookie, IndexedDB, Webstorage, HTML 5

webstorage تقریبا فناوری جدیدی است که برای نگهداری ثابت داده‌ها بر روی سیستم کاربر استفاده می‌شود. webstorage مزایای زیادی برای برنامه‌های تحت وب دارد. برای مثال با استفاده از آن میتوان فعالیت‌های کاربر را رصد کرد، بدون اینکه کد و دیتابیس سمت سرور را دخالت دهیم. حتی اگر سیستم کاربر آفلاین هم بشود برنامه می‌تواند همچنان به فعالیتش ادامه دهد. در این مقاله به مزایای این روش می‌پردازیم.

WebStorage در برابر کوکی ها

یکی از روش‌های سنتی ذخیره اطلاعات در سیستم کاربر، کوکی‌ها در بستر Http هستند. تفاوت‌های زیادی بین این دو وجود دارد که تعدادی از آن‌ها را در زیر بررسی می‌کنیم:

مکانیزم ذخیره سازی:

کوکی‌ها داده‌های ساخت یافته‌ای هستند که از وب سرور به سمت مرورگر کاربر به عنوان پاسخی در ازای درخواستی ارسال می‌شوند. درخواست و پاسخ کوکی‌ها شامل بخش هدر بوده که اطلاعات آن باعث شناسایی کوکی برای مدیریت و شناسایی آن‌ها می‌گردد تا هر موقع درخواستی صورت بگیرد، به سمت سرور برگشت خواهد خورد.

به طور خلاصه اینکه کوکی‌ها توسط درخواست‌ها و پاسخ‌ها ایجاد یا به روز می‌شوند. در نتیجه داده‌ها چه تغییر کرده باشند چه تغییر نکرده باشند، همیشه بخشی از هدر Http هستند. در سوی دیگر webstorage به طور کامل به صورت کلاینتی پیاده سازی گشته است و وب سرور را درگیر کار خودشان نمی‌کنند و کارایی بهتری را ارائه می‌دهند. از آنجا که همه چیز در خود سیستم کاربر اتفاق می‌افتد، در صورت از دست دادن کانکشن شبکه، کاربر می‌تواند همچنان به فعالیت‌های به روزرسانی و تغییر ادامه دهد.

چند نسخه از مرورگر

کاربری که با وب سایت مدنظر کار میکند میتواند توسط چند مرورگر مختلف یا چند تب و پنجره مختلف به طور همزمان کار کند و اطلاعات آن در همه‌ی مرورگرها و دیگر پنجره‌های آن مرورگر قابل دسترس می‌باشد.

محدودیت حجمی

محدودیت کوکی‌ها و webstorage از نظر حجم ذخیره سازی بین مروگرهای مختلف، متفاوت است. ولی در حالت کلی در بیشتر مرورگرها محدودیت حجم 4 کیلوبایت برای کوکی‌ها موجود است. (این ابزار می تواند نهایت حجمی را که که مرورگر شما از کوکی پشتیبانی می‌کند، نشان دهد.)

در مورد webstorage استاندارد W3C محدودیتی اعلام نکرده است و تصمیم گیری بر سر این موضوع را به سازندگان مرورگرها واگذار کرده است. ولی در حالت کلی حجم بیشتری از کوکی را ذخیره میکند و عموما تا 5 مگابایت توانایی ذخیره سازی وجود دارد. بدین ترتیب حجم آن 124,527% بیشتر از کوکی‌ها است. (این ابزار می‌تواند نهایت حجمی را که مرورگر شما از webstorage پشتیبانی می‌کند، ببینید).

انواع Webstorage

دو نوع متد ذخیره سازی در webstorage داریم:

session storage

local storage

Web Storage type	Lifetime of stored data	Data structure	Data type
sessionStorage	Session only	Key/value pairs	String
localStorage	Persistent	Key/value pairs	String

SessionStorage

داده‌هایی که بدین صورت ذخیره می‌شوند تنها تا زمانی دوام آورده و پایدار هستند که session مرورگر فعال است؛ یعنی تا زمانیکه کاربر در سایت فعلی حضور دارد.
استفاده از این روش برای ذخیره سازی‌های موقت عالی است. برای نمونه مقادیر ورودی فرمی که کاربر در حال کار با آن است، میتواند به طور موقت ذخیره شوند تا زمانی که کاربر بتواند تمامی مراحل را طی کرده، بدون اینکه ارجاعی به دیتابیس سمت سرور داشته باشد. همچنین ذخیره موقت داده‌ها می‌تواند به کاربر کمک کند تا در صورت rereshهای ناگهانی یا بسته شدن‌های ناگهانی مرورگرها، نیازی به ورود مجدد داده‌ها نداشته باشد.

LocalStorage

در این روش داده‌ها با از دست رفتن session مرورگر جاری از بین نمی‌رود و برای بازدیدهای آتی کاربر از سایت، داده‌ها همچنان در دسترس هستند. **پشتیبانی مرورگرها**
وب سایت caniuse گزارش می‌دهد که اکثر مرورگرها پشتیبانی خوبی از webstorage دارند.

Browser	Version
Internet Explorer	**IE 8** and above
Mozilla Firefox	**Firefox 3.5** and above
Google Chrome	**Chrome 4** and above
Apple Safari	**Safari 4** and above
Opera	**Opera 11.5** and above

با اینکه توصیه نامه W3C از پایان کار پیاده سازی این قابلیت خبر میدهد ولی در حال حاضر که این مقاله تدوین شده است هنوز نهایی اعلام نشده است. برای پشتیبانی مرورگرهای قدیمی از webstorage می‌توان از فایل جاوااسکریپتی Store.js کمک گرفت.

مفاهیم امنیتی و محافظت از داده ها

محدودیت‌های حمایتی و حفاظتی webstorage دقیقا همانند کوکی هاست. به این معنی که وب سایت‌های دیگر توانایی اتصال به webstorage سایت دیگری را ندارند. البته این مورد ممکن است برای وب سایت هایی که بر ساب دومین تکیه کرده‌اند ایجاد مشکل کند. برای حل این مسائل می‌توانید از کتابخانه‌های سورس بازی چون Cross Storage که توسط Zendesk ارائه شده است، استفاده کرد.

همانند هر مکانیزم ذخیره سازی سمت کلاینت، مواردی توصیه میگردد که رعایت آن‌ها از لحاظ امنیتی پر اهمیت است. به عنوان نمونه ذخیره‌ی اطلاعات شخصی و موارد حساس توصیه نمی‌گردد؛ چرا که احتمال دسترسی آسان نفوذگران به داده‌های محلی و خواندن آن‌ها وجود دارد.

Data Integrity یا یکپارچگی داده‌ها نیز در نظر گرفته شده است. باید حفاظتی در برابر عدم موفقیت ذخیره سازی داده‌ها نیز وجود داشته باشد. این عدم موفقیت‌ها میتواند به دلایل زیر رخ دهد:
ا گر کاربر قابلیت webstorage را غیرفعال کرده باشد.

اگر فضایی برای کاربر باقی نمانده باشد.

با محدودیت حجمی webstorage مواجه شده است.

با مواجه شدن با خطاها یک استثنا صادر می‌شود که می‌توانید آن را دریافت و کنترلی را روی برنامه تحت وب داشته باشید. یک نمونه استثنا QuotaExceededError

IndexedDB

یکی از فرایندهای ذخیره سازی داده‌ها که همان مزایای webstorage را ارائه میدهد indexed Database API است. این قابلیت از HTML 5 اضافه شده است و قسمتی از مشخصات webstorage شناخته نمی‌شود. برای همین مستنداتی در حوزه‌ی webstorage برای آن پیدا نخواهید کرد ولی قابلیت‌هایی فراتر از webstorage دارد.

این قابلیت پیچیدگی بیشتری را نسبت به خود webstorage ایجاد می‌کند، ولی فرصت‌های بسیاری را برای ذخیره سازی داده‌هایی با معماری‌های پیچیده‌تر و رابطه‌ها را می‌دهد. با استفاده از IndexedDB داده‌ها به شکل دیتابیس‌های سمت سرور RDMS ذخیره می‌شوند و این قابلیت را دارید که به سمت آن کوئری هایی مشابه بانک‌های اطلاعاتی سمت سرور را ارسال کنید.

در قسمت آتی نحوه کدنویسی آن را فرا خواهیم گرفت.

نظرات خوانندگان

نویسنده: احمد نواصری
تاریخ: ۱۳۹۴/۰۴/۱۴ ۲:۵۹

آیا روش‌های ذکر شده (Session & Local Storage) برای طراحی یک سبد خرید (در یک پروژه فروشگاه اینترنتی) مناسب هستند؟ اگر مناسب هستند، بهتر از Session معمولی کار میکنند؟

نویسنده: علی یگانه مقدم
تاریخ: ۱۳۹۴/۰۴/۱۴ ۳:۷

اینجا را بخوانید