

# **Работа с данными на клиенте**

## **Хранилища данных**

# Table of contents

1. Cookies
2. Web Storage API
  - LocalStorage
  - SessionStorage
3. IndexedDB
4. WebSQL



CREATE



READ



UPDATE



DELETE

---

C

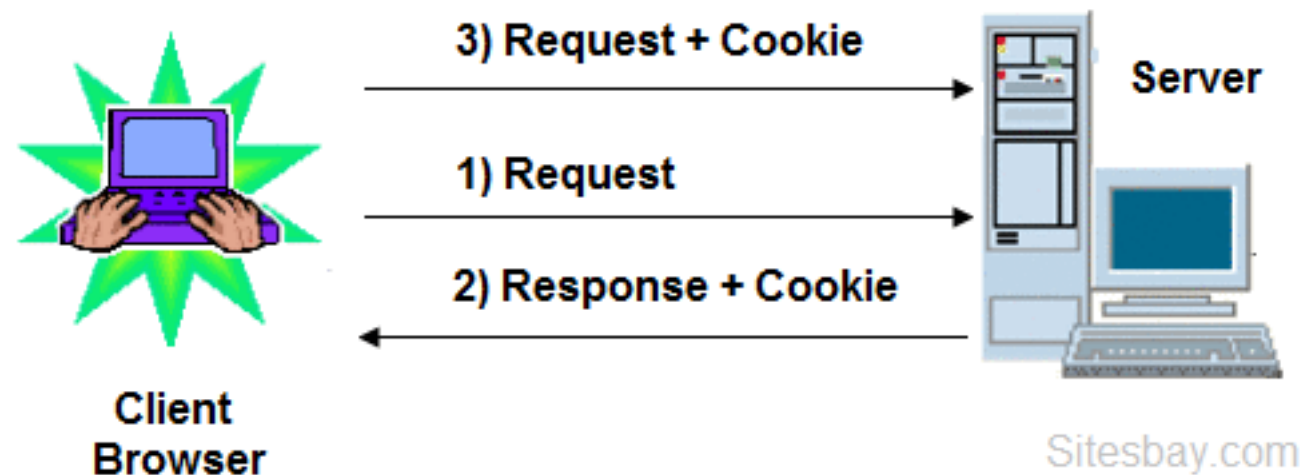
R

U

D

# Cookies

Классический способ хранения  
простых строчных данных  
внутри документа



Обычно cookies отсылаются с сервера на клиент,  
который может сохранять их, а затем отправлять  
обратно на сервер в ответ на последующие запросы

# Cookies: примеры использования

- Управление сессиями аккаунта
- Отслеживание пользовательской информации
- Хранение данных на клиентской стороне
- Хранение общих данных – например, пользовательских настроек

# Cookies: преимущества и недостатки

## Преимущества

- Можно использовать для коммуникации с сервером
- Можно определить срок автоматического окончания вместо того, чтобы удалять вручную

## Недостатки

- Добавляются к загрузке страницы документа
- Могут хранить только небольшое количество данных
- Могут содержать только строки
- Потенциальные проблемы с безопасностью
- Не рекомендуется для хранения данных на клиенте с момента появления Web Storage API

# CRUD-операции с Cookies

```
// Create
```

```
document.cookie = "user_name=Ire Aderinokun";
```

```
document.cookie = "user_age=25;max-age=31536000;secure";
```

```
// Read (All)
```

```
console.log( document.cookie );
```

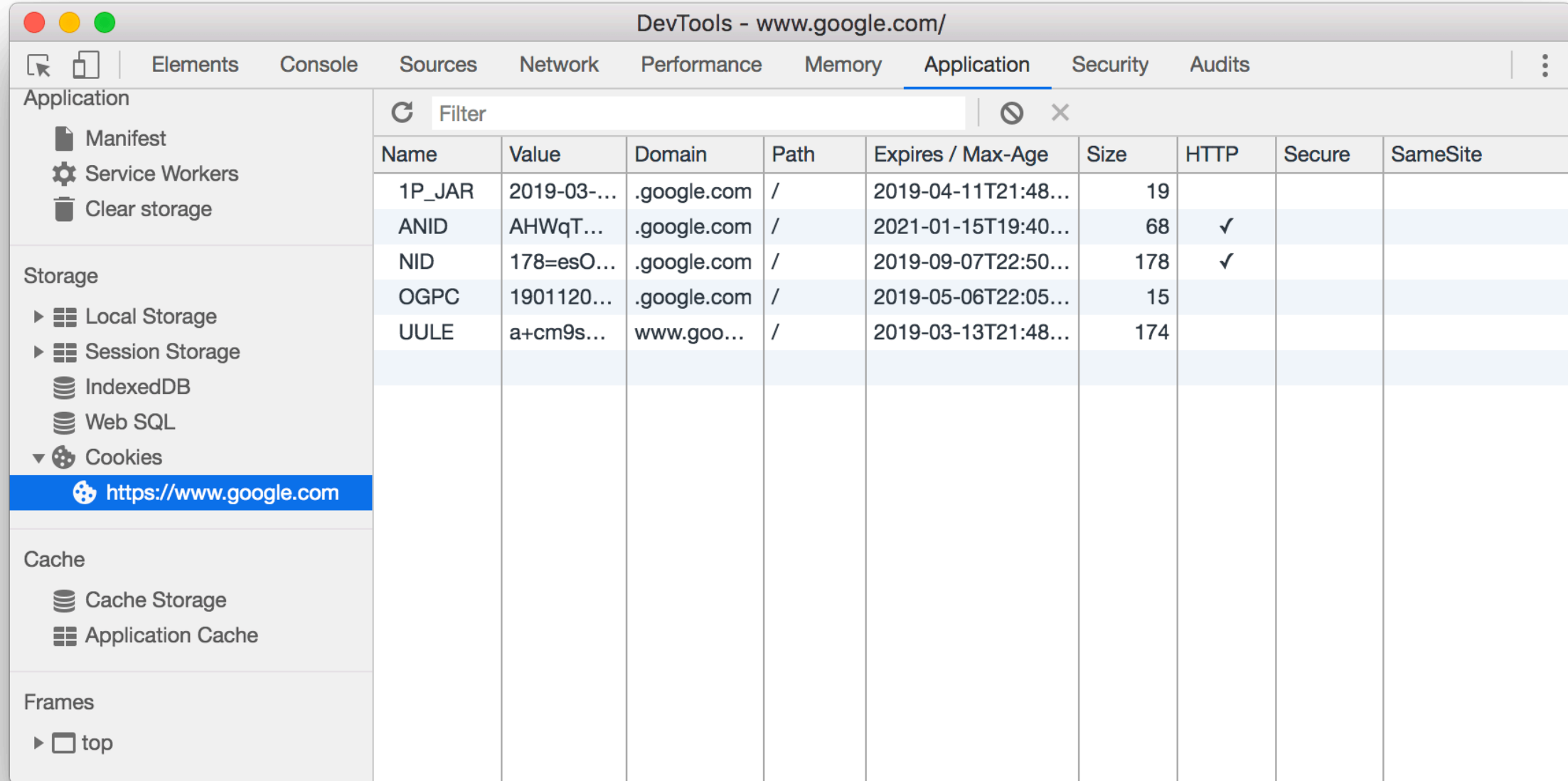
```
// Update
```

```
document.cookie = "user_age=24;max-age=31536000;secure";
```

```
// Delete
```

```
document.cookie = "user_name=Ire Aderinokun;expires=Thu, 01 Jan 1970  
00:00:01 GMT";
```

# Работа с Cookies в Devtools



The screenshot shows the Chrome DevTools interface with the 'Application' tab selected. The left sidebar shows the 'Application' panel with 'Cookies' expanded. The main area displays a table of cookies for the domain 'www.google.com'.

Name	Value	Domain	Path	Expires / Max-Age	Size	HTTP	Secure	SameSite
1P_JAR	2019-03-...	.google.com	/	2019-04-11T21:48...	19			
ANID	AHWqT...	.google.com	/	2021-01-15T19:40...	68	✓		
NID	178=esO...	.google.com	/	2019-09-07T22:50...	178	✓		
OGPC	1901120...	.google.com	/	2019-05-06T22:05...	15			
UULE	a+cm9s...	www.goo...	/	2019-03-13T21:48...	174			

[View, Edit, And Delete Cookies With Chrome DevTools](#)



# Поддержка Cookies браузерами

Current aligned Usage relative Date relative Apply filters Show all ?															
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baid Brow:
6-7		2-3		3.1-3.2	10.1										
8-10	12-17	3.5-68	4-76	4-12.1	11.5-60	3.2-12.3		2.1-4.4.4	12-12.1				4-9.2		
11	18	69	77	13	62	13.1	all	76	46	76	68	12.12	10.1	1.2	7.1
	76	70-71	78-80	13.1-TP											

[CanIUse: name/value pairs](#)

# LocalStorage

WebStorage API – специальный API для хранения данных в браузере в формате ключ-значение



По сравнению с cookies и является более интуитивным и безопасным способом хранения простых данных внутри браузера

# CRUD-операции с LocalStorage

// Create

```
const user = { name: 'Ire Aderinokun', age: 25 }  
localStorage.setItem('user', JSON.stringify(user));
```

// Read

```
console.log( JSON.parse(localStorage.getItem('user')) )
```

// Update

```
const updatedUser = { name: 'Ire Aderinokun', age: 24 }  
localStorage.setItem('user', JSON.stringify(updatedUser));
```

// Delete

```
localStorage.removeItem('user');
```

# **LocalStorage: преимущества и недостатки**

## **Преимущества (по сравнению с cookies)**

- Более простой и интуитивный интерфейс хранения данных
- Более безопасно для хранения данных на клиенте
- Позволяет хранить больше данных

## **Недостатки**

- Позволяет хранить только строки

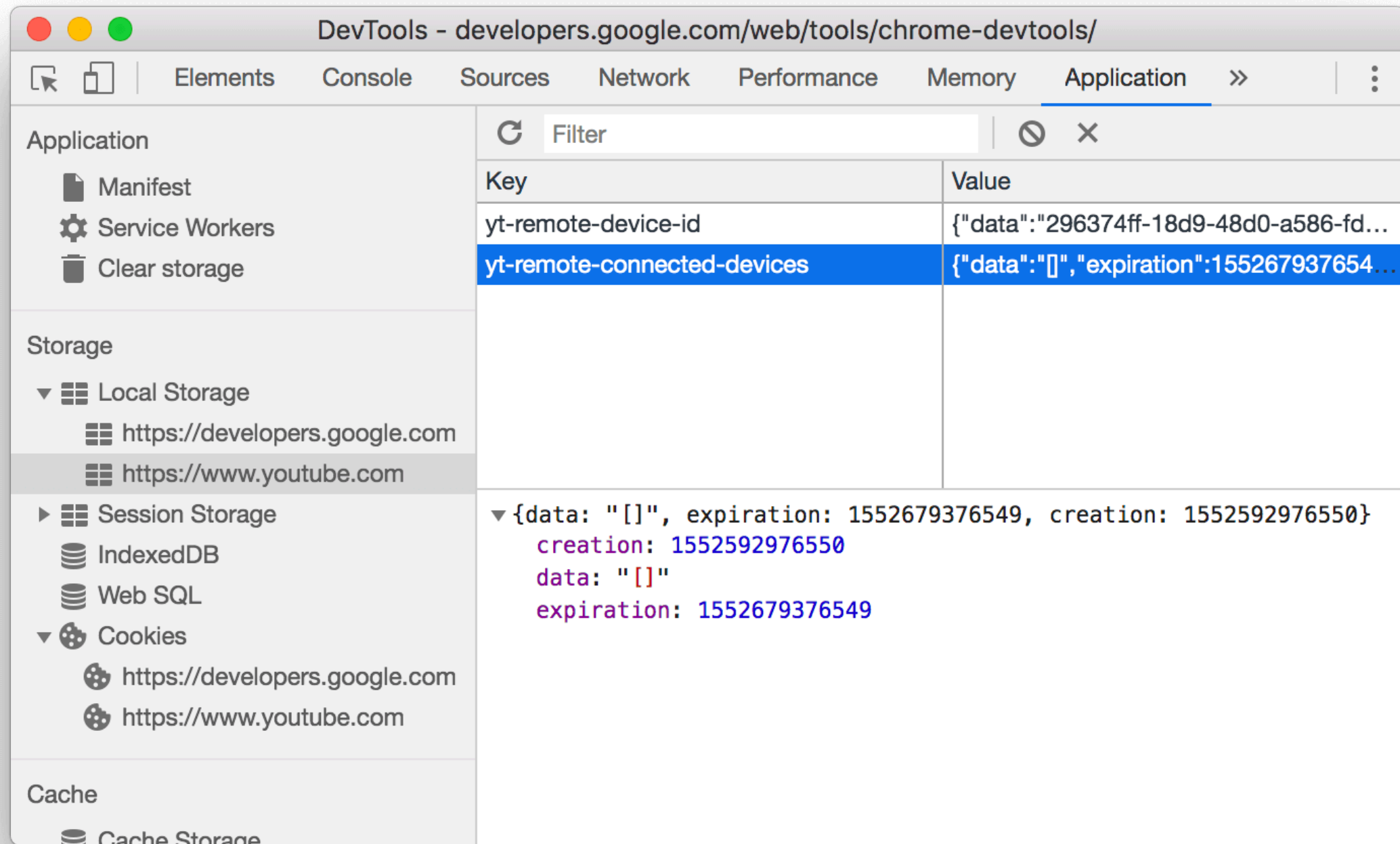
# LocalStorage: храним JSON

Хотя технически мы можем хранить в LocalStorage только строки, это обходится за счет преобразования в JSON.

Таким образом мы можем сохранять в локальном хранилище более сложные данные по сравнению с Cookies

```
// Create  
  
const user = { name: 'Ire Aderinokun', age: 25 }  
  
localStorage.setItem('user', JSON.stringify(user)) ;
```

# Работа с LocalStorage в Devtools



[View And Edit Local Storage With Chrome DevTools](#)

# Поддержка LocalStorage браузерами

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android
6-7		2-3		3.1-3.2	10.1						
8-10	12-17	3.5-68	4-76	4-12.1	11.5-60	3.2-12.3		2.1-4.4.4	12-12.1		
11	18	69	77	13	62	13.1	all	76	46	76	68
	76	70-71	78-80	13.1-TP							

UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser	KaiOS Browser
	4-9.2			
12.12	10.1	1.2	7.12	2.5

[CanIUse: LocalStorage](#)

# SessionStorage

Как и LocalStorage, является разновидностью WebStorage API

Аналогично по своей работе LocalStorage, за исключением того, что данные хранятся только для сессии вкладки браузера

Как только пользователь уходит со страницы и закрывает браузер, данные очищаются



# CRUD-операции с SessionStorage

// Create

```
const user = { name: 'Ire Aderinokun', age: 25 }  
sessionStorage.setItem('user', JSON.stringify(user));
```

// Read

```
console.log( JSON.parse(sessionStorage.getItem('user')) )
```

// Update

```
const updatedUser = { name: 'Ire Aderinokun', age: 24 }  
sessionStorage.setItem('user', JSON.stringify(updatedUser));
```

// Delete

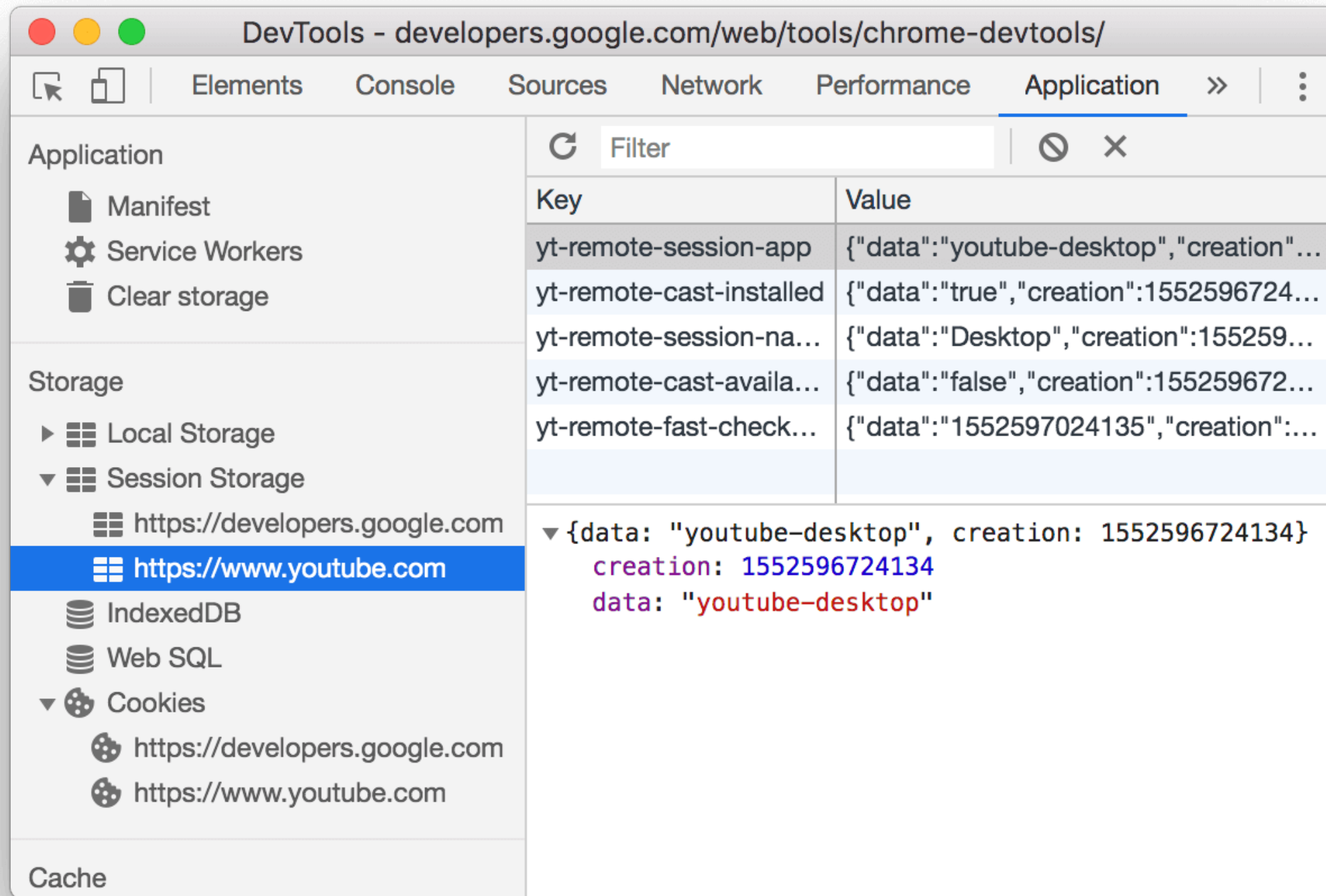
```
sessionStorage.removeItem('user');
```

# **SessionStorage: достоинства и недостатки, поддержка**

Достоинства, недостатки и поддержка браузерами точно такие же как и у LocalStorage

[CanIUse: SessionStorage](#)

# Работа с SessionStorage в Devtools



[View And Edit Session Storage With Chrome DevTools](#)

# IndexedDB

IndexedDB – более сложное и проработанное решение для хранения данных в браузере



Низкоуровневый API для хранения на клиенте значительного количества структурированных данных

[IndexedDB Promised library](#)

# CRUD-операции с IndexedDB

## Открытие БД и создание хранилища

```
function OpenIDB() {  
    return idb.open('SampleDB', 1, function(upgradeDb) {  
        const users = upgradeDb.createObjectStore('users', {  
            keyPath: 'name'  
        });  
    });  
}
```

# CRUD-операции с IndexedDB

## Создание и обновление данных

```
// 1. Открытие БД
OpenIDB().then((db) => {
  const dbStore = 'users';

  // 2. Создание транзакции на чтение/запись для текущего store БД
  const transaction = db.transaction(dbStore, 'readwrite');
  const store = transaction.objectStore(dbStore);

  // 3. Добавление данных в store
  store.put({
    name: 'Ire Aderinokun',
    age: 25
  });

  // 4. Исполнение транзакции
  return transaction.complete;
});
```

# CRUD-операции с IndexedDB

## Получение данных

// 1. Открытие БД

```
OpenIDB().then((db) => {  
    const dbStore = 'users';
```

// 2. Открытие транзакции к store БД только на чтение

```
const transaction = db.transaction(dbStore);  
const store = transaction.objectStore(dbStore);
```

// 3. Возвращаем результат

```
return store.get('Ire Aderinokun');
```

```
}).then((item) => {  
    console.log(item);  
})
```

# CRUD-операции с IndexedDB

## Удаление данных

```
// 1. Открытие БД
OpenIDB().then((db) => {
  const dbStore = 'users';

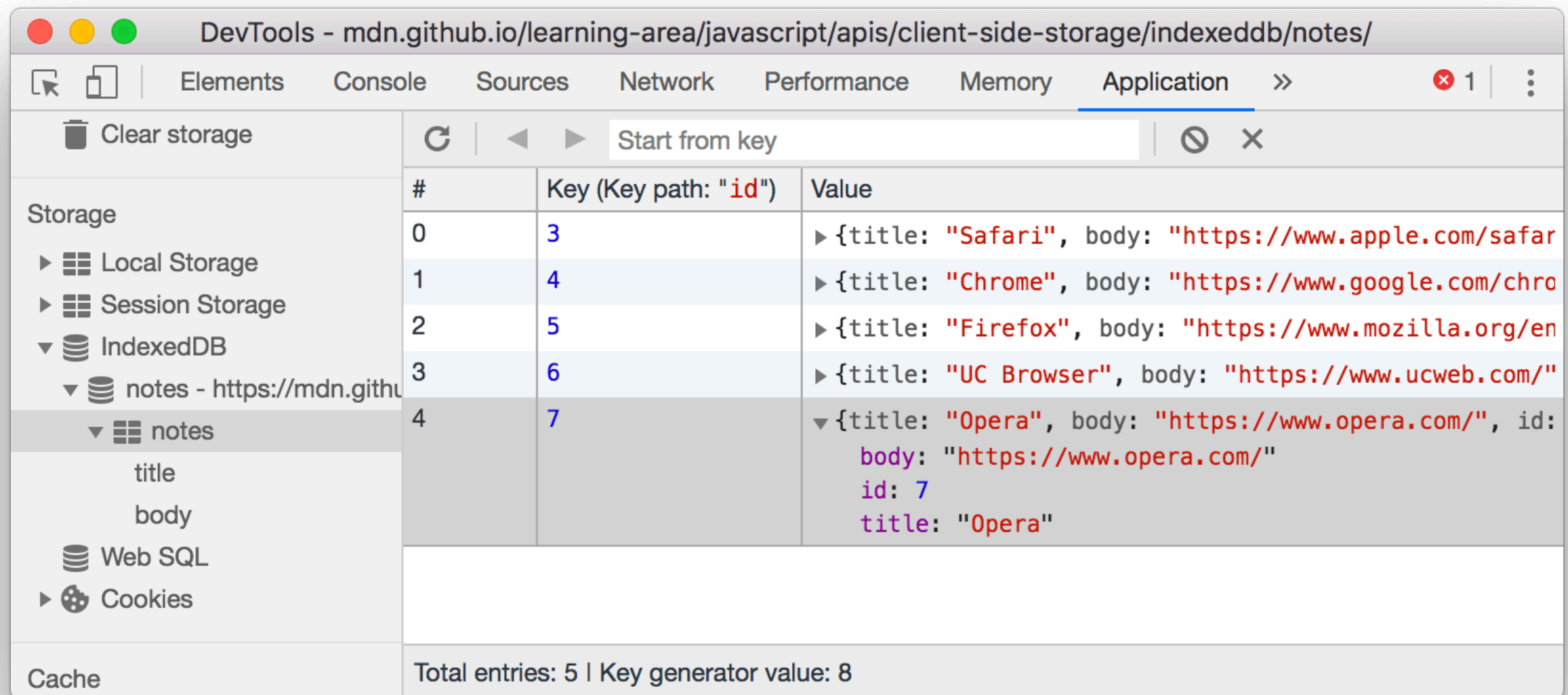
  // 2. Открытие транзакции к store БД на запись и чтение
  const transaction = db.transaction(dbStore, 'readwrite');
  const store = transaction.objectStore(dbStore);

  // 3. Удаление данных по передаваемому ключу
  store.delete('Ire Aderinokun');

  // 4. Исполнение транзакции
  return transaction.complete;
})
```



# Работа с IndexedDB в Devtools



[View And Change IndexedDB Data With Chrome DevTools](#)

# **IndexedDB: преимущества и недостатки**

## **Преимущества**

- Может обрабатывать более сложные структурированные данные
- Может работать с разными “базами данными” и “таблицами” внутри каждой “базы данных”
- Больше объем хранения

## **Недостатки**

- Более сложное взаимодействие по сравнению с Web Storage API

# Поддержка IndexedDB браузерами

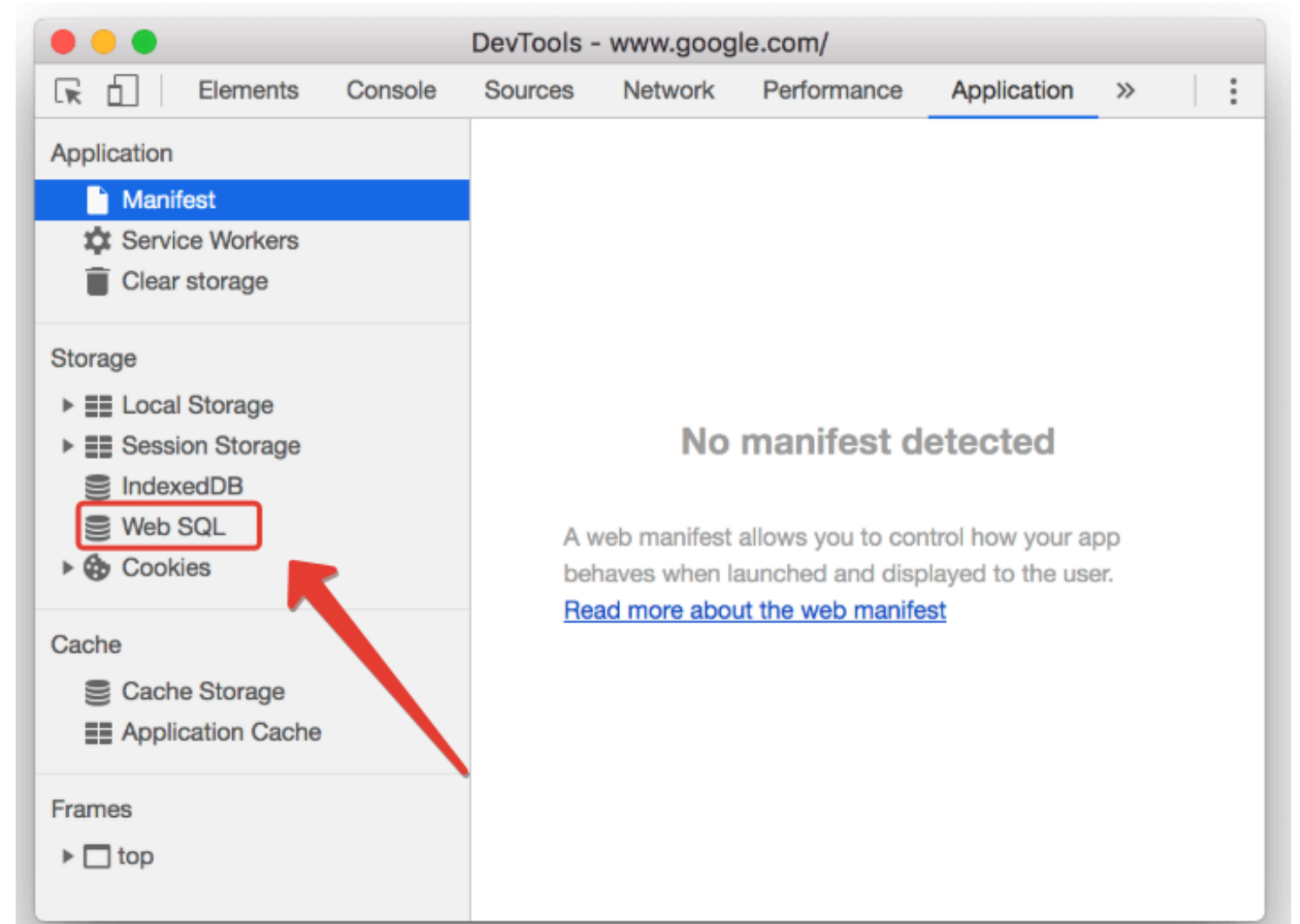
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android
		2-3.6	4-10								
		4-9 <sup>1</sup>	11-22 <sup>1</sup>	3.1-7		3.2-7.1					
6-9		10-15 <sup>1</sup>	23 <sup>1</sup>	<sup>2</sup> 7.1-9.1	10-12.1	<sup>2</sup> 8-9.3		2.1-4.3			
<sup>1</sup> 10	<sup>1</sup> 12-17	16-68	24-76	10-12.1	15-60	10-12.3		4.4-4.4.4	12-12.1		
<sup>1</sup> 11	<sup>1</sup> 18	69	77	13	62	13.1	all	76	46	76	68
	76	70-71	78-80	13.1-TP							

UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser	KaiOS Browser
	4-9.2			
12.12	10.1	1.2	7.12	2.5

[CanIUse: IndexedDB](https://caniuse.com/indexeddb)

# WebSQL

API для реляционной  
базы на клиенте,  
сходное с SQLite



**DEPRECATED**

С 2010 рабочая группа W3C прекратила работу над этой спецификацией и этот API больше не является частью спецификации HTML и не должен использоваться

# Сравнение

	Cookies	Local Storage	Session Storage	IndexedDB
Лимит хранимых данных	~ 4KB	~ 5MB	~ 5MB	До половины жесткого диска
Постоянное хранение	Да	Да	Нет	Да
Тип данных	Строка	Строка	Строка	Любые структурированные данные
Индексируемость	Нет	Нет	Нет	Да

# Еще материалы

["Instant Loading" with IndexedDB \(Building a PWA, Part 2\)](#)

[Краткое, но исчерпывающее руководство по IndexedDB](#)

[Почему не стоит использовать LocalStorage](#)

[Ещё раз о безопасности или где хранить токен](#)