

УДК 004.6

***ИСПОЛЬЗОВАНИЕ ЛОКАЛЬНОГО ХРАНИЛИЩА БРАУЗЕРА ДЛЯ  
ХРАНЕНИЯ ДАННЫХ ПОЛЬЗОВАТЕЛЯ В ПРИЛОЖЕНИИ REACT***

***Акимов А.Е.,***

*магистрант,*

*Калужский государственный университет им. К.Э. Циолковского,*

*Калуга, Россия*

***Виноградская М.Ю.,***

*к.пед.н., доцент,*

*Калужский государственный университет им. К.Э. Циолковского,*

*Калуга, Россия*

**Аннотация.**

Статья посвящена описанию примера использования локального хранилища браузера для хранения данных, полученных от пользователя, для дальнейшего использования в информационной системе. В статье автор описывает основные преимущества использования локального хранилища для выполнения такого рода задачи, проводит сравнение локального и сессионного хранилища браузера, приводит пример использования такого метода, демонстрирует результат работы, подводит итоги.

**Ключевые слова:** веб-приложение, локальное хранилище, управление задачами и временем, Localstorage, Sessionstorage, React, JavaScript.

***USING THE BROWSER'S LOCAL STORAGE TO STORE USER DATA  
IN THE REACT APP***

***Akimov A.E.,***

*Undergraduate,*

Дневник науки | [www.dnevniknauki.ru](http://www.dnevniknauki.ru) | СМИ Эл № ФС 77-68405 ISSN 2541-8327

*Kaluga State University named after K.E. Tsiolkovsky,  
Kaluga, Russia*

***Vinogradskaya M.Y.,***  
*Candidate of Pedagogical Sciences, Associate Professor,  
Kaluga State University named after K.E. Tsiolkovsky,  
Kaluga, Russia*

### **Annotation.**

The article describes an example of using local browser storage to store data received from the user for further use in an information system. In the article, the author describes the main advantages of using local storage to perform this kind of task, compares the local and session storage of the browser, gives an example of using this method, demonstrates the result of the work, and summarizes the results.

**Keywords:** web application, local storage, task and time management, Localstorage, Sessionstorage, React, JavaScript.

При разработке веб-приложений или сайтов, в чей функционал входит сохранение и последующее использование данных пользователя, возникает вопрос – как хранить все полученные данные? Вариантов много, это может быть традиционная база данных на стороне сервера, но что, если необходимо, например по максимуму снизить нагрузку на серверную часть? Или же идеология будущего приложения включает в себя надобность хранения этих данных у пользователя, а не на сервере? Решением для таких вопросов может быть использование локального хранилища браузера пользователя.

Локальное хранилище — это идейное продолжение технологии cookies, которое сохраняет данные пользователя, в файл с расширением json. Данная

технология на данный момент поддерживается всеми современным браузерами и даже большинством старых и мало популярных.

Существует два вида локального хранилища: `sessionStorage` и `localStorage`. Первое это сессионное хранилище, которое очищается каждый раз, когда пользователь закрывает браузер завершая сессию работы с ним. Второе, это хранилище, которое сохраняется на постоянной основе, вне зависимости от действий пользователя, оно не отчищается не при закрытии браузера, ни при завершении работы ПК.

Локальное хранилище представляет из себя файл типа JSON, это текстовый формат хранения данных, легко читаемый как машиной, так и человеком. В локальном хранилище такой файл является способом хранения информации по типу ключ-значение.

Плюсами использования локального хранилища для хранения данных пользователя в рамках веб-приложения являются:

- полная гарантия конфиденциальных данных пользователя, вследствие хранения данных не на удаленном сервере;
- удобство и быстрое начало работы с сервисом без надобности регистрации и авторизации;
- отсутствие дополнительной нагрузки на хост;
- возможность работать с сервисом без подключения к сети интернет, при условии предварительного скачивания пользователем сборки проекта с сервиса GitHub;
- подход открытого исходного кода, который внушает доверие пользователям и возможность персонального изменения самим пользователем под себя, при имении у него должных навыков.

Минусом, вытекающим из факта отсутствия возможности авторизации, является отсутствие облачной синхронизации, но данный минус можно легко нивелировать с помощью интеграции с облачными сервисами хранения

информации, например google drive или Яндекс Диск . Это позволит облачно и автоматически переносить информацию между устройствами пользователя и при этом сохранит конфиденциальность данных, так как они все будут храниться на личном облачном диске пользователя.

Большинство веб-приложений разрабатываются на трех языках программирования: JavaScript, HTML, CSS. Для выполнения действий с локальным хранилищем и реализацией другого функционала приложений используют язык JavaScript.

Для использования локального хранилища в проекте веб-приложения существуют следующие методы и функции на языке JavaScript (таблица 1).

Таблица 1 - Методы и функции JavaScript

localStorage	
Метод	Пояснение
setItem(key, value)	сохранить пару ключ/значение
getItem(key)	получить данные по ключу key
removeItem(key)	удалить данные с ключом key
clear()	удалить всё
key(index)	получить ключ на заданной позиции
length	количество элементов в хранилище

Чтобы использовать локальное хранилище по максимально похожему принципу на традиционные базы данных с сущностями, атрибутами и кортежами, надо чтобы каждая строка файла JSON локального хранилища с определенным ключом содержала многомерный массив данных. Такую операцию можно провести без особых проблем. Таким образом получится JSON строка, ключом которой будет название сущности, а содержание самой строки (значение) будет многомерным массивом, который и будет этой сущностью. Атрибутами будут столбцы, а кортежами строки массива.

В качестве примера такого массива, можно рассмотреть массив, созданный для хранения «задач» в приложении для управления временем и задачами в учебном процессе (таблица 2).

Таблица 2 - Массив «Хранение задач»

Ключ в локальном хранилище: «all_task»		
Атрибут	Тип данных	Примечание
ID	Целочисленный	Идентификационный уникальный номер, для облегчения написания логики работы приложения.
Name	Строковый	Название задачи
Tsk_	Строковый	Краткое описание задачи
Start	Дата	Когда необходимо начать выполнение задачи
End	Дата	Когда необходимо завершить задачу

Внутри значения JSON строки может быть что угодно.

Для удобного использования локального хранилища в проекте React необходимо разработать функцию работы с ним с использованием «Хуков» React.

Для начала нужно импортировать `useEffect` и `useState` из библиотеки React.

Далее мы можем использовать «React Hook» для выполнения эффекта сохранения данных в хранилище браузера.

Использование метода `JSON.stringify` требуется для сохранения различных типов данных помимо строчных, например, как в нашем случае – массив.

Для извлечения данных из хранилища используем метод `getItem`.

Чтобы постоянно не выполнять однотипные действия при разработке, где требуется сохранение множества различных данных от пользователя, можно создать собственный пользовательский «React Hook» для использования локального хранилища. Такой «хук» будет иметь следующий код (рисунок 1):

```
1 import { useState, useEffect } from "react";
2
3 function getStorageValue(key, defaultValue) {
4   if (typeof window !== "undefined") {
5     const saved = localStorage.getItem(key);
6     const initial = saved !== null ? JSON.parse(saved) : defaultValue;
7     return initial;
8   }
9 }
10
11 export const useLocalStorage = (key, defaultValue) => {
12   const [value, setValue] = useState(() => {
13     return getStorageValue(key, defaultValue);
14   });
15
16   useEffect(() => {
17     localStorage.setItem(key, JSON.stringify(value));
18   }, [key, value]);
19
20   return [value, setValue];
21 };
```

Рис. 1 - Пример пользовательского "Хука" (составлено авторами)

Данный пользовательский хук ожидает два аргумента: key и value. При вызове такого «хука» в необходимом нам компоненте мы будем передавать в написанный нами «хук» эти данные. Key – ключ необходимой строки, value – значение строки которое необходимо записать или же получить.

Пример использования только что написанного «хука» в форме веб-приложения (рисунок 2):

```
6 const [nameTsk, setNameTsk] = useState('');
7 const [Tsk_, setTsk_] = useState('');
8 const [StartDate, setStartDate] = useState('');
9 const [EndDate, setEndDate] = useState('');
10 const [name, setName] = useLocalStorage("all_tsk", "");
11
12 var list = Object.assign([], name);
13 list.reverse();
14
15 function pushArr() {
16
17   if (name != ""){
18     let arr = Object.assign([], name);
19     console.log({arr});
20     let last = arr.slice(-1);
21     let l;
22     let kk = last.filter(function(item) {l = item.id});
23     console.log("its l id = ",l+1)
24     let add =
25     {id: l + 1, name: nameTsk, start: StartDate, end: EndDate, tsk__: Tsk_};
26     arr.push(add);
27     setName(arr);
28   }
29   else {
30     const arr = [
31       {id: 1, name: nameTsk, start: StartDate, end: EndDate, tsk__: Tsk_},
32     ];
33     setName(arr);
34   }
```

Рис. 2 - Использование созданного пользовательского "хука" (составлено авторами)

Для начала импортирован написанный «хук»:

```
import { useLocalStorage } from "../GComponents/useLocalStorage";
```

Далее он вызывается в строках 6–10 продемонстрированных на рисунке 2. Где «nameTsk» является ключом, а setNameTsk методом записи значения по ключу.

Далее в коде элемента продемонстрирована функция для записи массива из полученных с формы данных в строку, которая сначала определяет есть ли уже такой массив, в который необходимо добавить запись, или массив необходимо для начала создать. Надобность в создании массива возникает при первом использовании данной формы пользователем. Этот пример показывает вариант использования «хука» для записи массива или добавления записи в массив данных, полученных с формы.

Следующий пример (рисунок 3) показывает вариант использования всё того же «хука» для чтения и вывода данных из локального хранилища:

```
10  const [name, setName] = useLocalStorage("all_tsk", "");  
11  
12  var list = Object.assign([], name);  
13  list.reverse();  
--
```

Рис. 3 - Использование пользовательского "хука" для записи данных в переменную (составлено авторами)

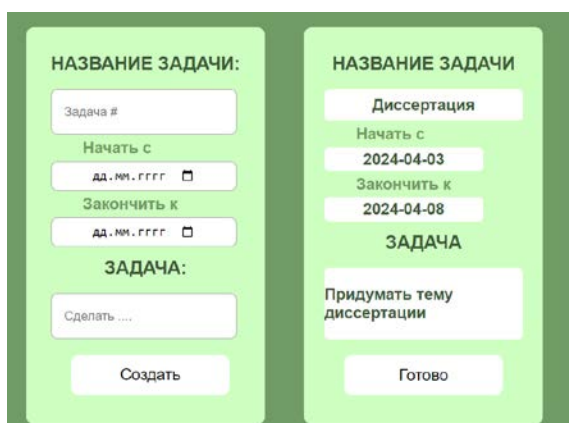
В данном примере мы видим прямое обращение к значению из строки с ключом «all\_tsk, которое имеет вид «name» (строка 12).

Тут происходит запись значения из хранилища в переменную «list» как в массив с сохранением его параметров.

Далее в коде написано стандартное решение для вывода массива блоков, содержащих по одному кортежу массива значений в каждом.

Результатом работы обоих примеров является страница, которая содержит в себе форму для записи данных слева, и два выведенных блока справа,

состоящие из данных, записанных в локальное хранилище с помощью первой формы (рисунок 4).



The image shows two side-by-side forms with a light green background. The left form is titled 'НАЗВАНИЕ ЗАДАЧИ:' and contains a text input field with 'Задача #', a 'Начать с' section with a date input 'дд.мм.гггг' and a calendar icon, a 'Закончить к' section with a date input 'дд.мм.гггг' and a calendar icon, a 'ЗАДАЧА:' section with a text input field 'Сделать ....', and a 'Создать' button. The right form is also titled 'НАЗВАНИЕ ЗАДАЧИ:' and contains a text input field with 'Диссертация', a 'Начать с' section with a date input '2024-04-03' and a calendar icon, a 'Закончить к' section with a date input '2024-04-08' and a calendar icon, a 'ЗАДАЧА' section with a text input field 'Придумать тему диссертации', and a 'Готово' button.

Рис. 4 - Пример результата работы функции (составлено авторами)

В режиме разработчика в браузере мы можем увидеть наполнение локального хранилища (рисунок 5), в нашем случае оно будет иметь следующий вид:

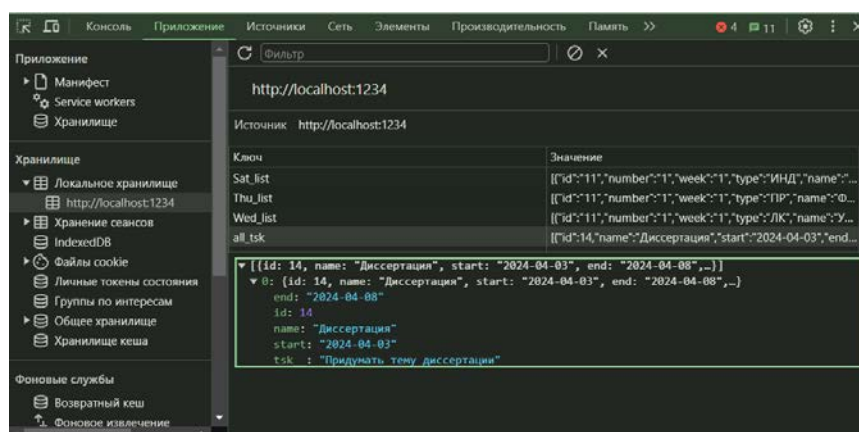


Рис. 5 - Пример вида локального хранилища из панели разработчика в браузере (составлено авторами)

Таким образом, в статье продемонстрирован пример работы приложения использующего метод сохранения данных пользователя в локальное хранилище, разобрана и написана функция пользовательского «React hook» Для удобного использования локального хранилища в проектах React. Описаны плюсы и минусы подобного метода.



**Библиографический список:**

1. Алекс Бэнкс, Ева Порселло. React и Redux. Функциональная веб-разработка. / А. Бенкс, Е. Порселло. – СПб. : ПИТЕР, 2018. -336 с.
2. Что такое локальное хранилище и как его используют. [Электронный ресурс]. – URL: [https://codernet.ru/articles/web/chto\\_takoe\\_lokalnoe\\_xranilishhe\\_i\\_kak\\_ego\\_ispolzuyut\\_v\\_programmirovanii](https://codernet.ru/articles/web/chto_takoe_lokalnoe_xranilishhe_i_kak_ego_ispolzuyut_v_programmirovanii) / (Дата обращения: 29.12.2023)
3. Лабберс П., Олберс Б., Салим Ф. HTML5 для профессионалов: мощные инструменты для разработки современных веб-приложений / Пер. с англ. М.: ООО “И.Д. Вильямс”, 2011 272 с.
4. Хранение данных в браузере. [Электронный ресурс]. – URL: <https://learn.javascript.ru/localstorage> (Дата обращения: 29.12.2023)
5. Хранилища данных и их использование: хрестоматия / сост. А.Б. Кригер ; Владивостокский государственный университет экономики и сервиса. – Владивосток : Изд-во ВГУЭС, 2017 – 120 с.

*Оригинальность 88%*