

Раздел

Бинарные данные и файлы

Навигация по уроку



TextEncoder

Комментарии

Поделиться



Редактировать на GitHub

 → Бинарные данные и файлы 20-го августа 2019

TextDecoder и TextEncoder

Что если бинарные данные фактически являются строкой? Например, мы получили файл с текстовыми данными.

Встроенный объект `TextDecoder` позволяет декодировать данные из бинарного буфера в обычную строку.

Для этого прежде всего нам нужно создать сам декодер:

```
1 let decoder = new TextDecoder([label], [options]);
```

- **label** – тип кодировки, utf-8 используется по умолчанию, но также поддерживаются big5, windows-1251 и многие другие.
- **options** – объект с дополнительными настройками:
 - **fatal** – boolean, если значение true, тогда генерируется ошибка для невалидных (не декодируемых) символов, в ином случае (по умолчанию) они заменяются символом \uFFFD.
 - **ignoreBOM** – boolean, если значение true, тогда игнорируется BOM (дополнительный признак, определяющий порядок следования байтов), что необходимо крайне редко.

...и после использовать его метод decode:

```
1 let str = decoder.decode([input], [options]);
```



- **input** – бинарный буфер (BufferSource) для декодирования.
- **options** – объект с дополнительными настройками:
 - **stream** – true для декодирования потока данных, при этом decoder вызывается вновь и вновь для каждого следующего фрагмента данных. В этом случае многобайтовый символ может иногда быть разделён и попасть в разные фрагменты данных. Это опция указывает TextDecoder запомнить символ, на котором остановился процесс, и декодировать его со следующим фрагментом.



Например:

```
1 let uint8Array = new Uint8Array([72, 101, 108, 108, 104]);  
2  
3 alert( new TextDecoder().decode(uint8Array) ); // Hello
```

```
1 let uint8Array = new Uint8Array([228, 189, 160, 229, 166]);  
2  
3 alert( new TextDecoder().decode(uint8Array) ); // 你好
```

Мы можем декодировать часть бинарного массива, создав подмассив:

```
1 let uint8Array = new Uint8Array([0, 72, 101, 108, 104]);  
2  
3 // Возьмём строку из середины массива  
4 // Также обратите внимание, что это создаёт только новый  
5 // Изменения в содержимом созданного подмассива повлияют  
6 let binaryString = uint8Array.subarray(1, -1);  
7  
8 alert( new TextDecoder().decode(binaryString) ); // Hel
```

Раздел

[Бинарные данные и файлы](#)

Навигация по уроку

TextEncoder

Комментарии

Поделиться



[Редактировать на GitHub](#)

TextEncoder

[TextEncoder](#) поступает наоборот – кодирует строку в бинарный массив.

Имеет следующий синтаксис:

```
1 let encoder = new TextEncoder();
```



Поддерживается только кодировка «utf-8».

Кодировщик имеет следующие два метода:

- **encode(str)** – возвращает бинарный массив `Uint8Array`, содержащий закодированную строку.
- **encodeInto(str, destination)** – кодирует строку (`str`) и помещает её в `destination`, который должен быть экземпляром `Uint8Array`.

```
1 let encoder = new TextEncoder();
2
3 let uint8Array = encoder.encode("Hello");
4 alert(uint8Array); // 72,101,108,108,111
```



Проводим [курсы по JavaScript и фреймворкам](#).



Комментарии

перед тем как писать...