Storing API Data Locally

James Millar
FREELANCE SOFTWARE DEVELOPER

@jamesmillar www.james-millar.co.uk

Overview

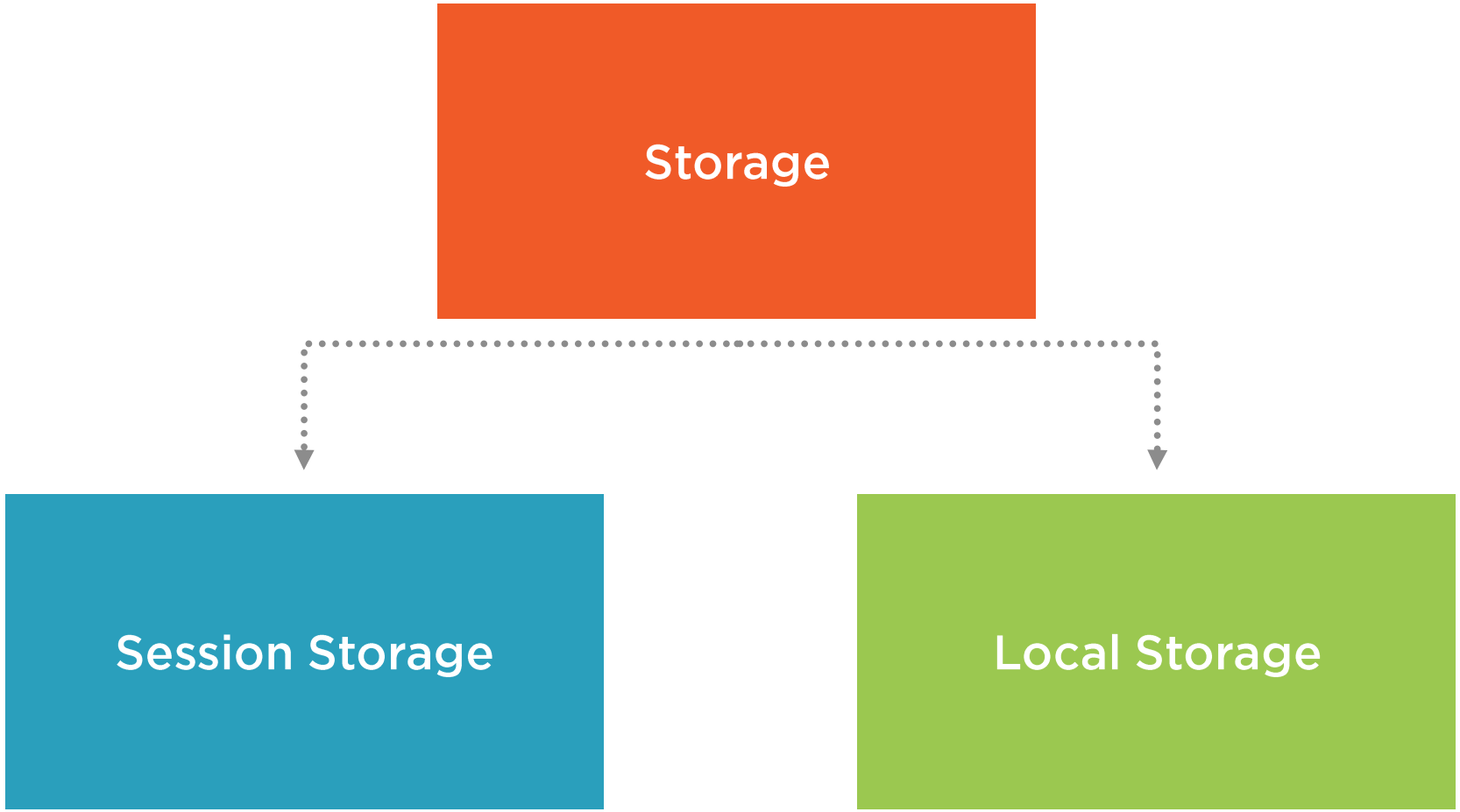
Explore the Storage API

Create shopping cart

Storing API data

- Complex objects
- Storing in local storage
- Retrieving from local storage

Storage API



Storage Interface

```
interface Storage {

readonly attribute unsigned long length;

DOMString? key(unsigned long index);

getter DOMString? getItem(DOMString key);

setter undefined setItem(DOMString key, DOMString value);

deleter undefined removeItem(DOMString key);

undefined clear();

};
```

Check Support for Storage API

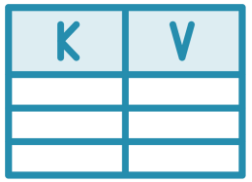
```
if (typeof(Storage) !== "undefined") {
  // local storage and session storage are available
}

else {
  // no support for the storage API
}
```

Working with Storage

Wiredbraincoffee.com

Cart

Local Storage



Javascript

```
localStorage.setItem(key,value)

localStorage.setItem(key,value)

localStorage.getItem(key)

localStorage.removeItem(key)
```

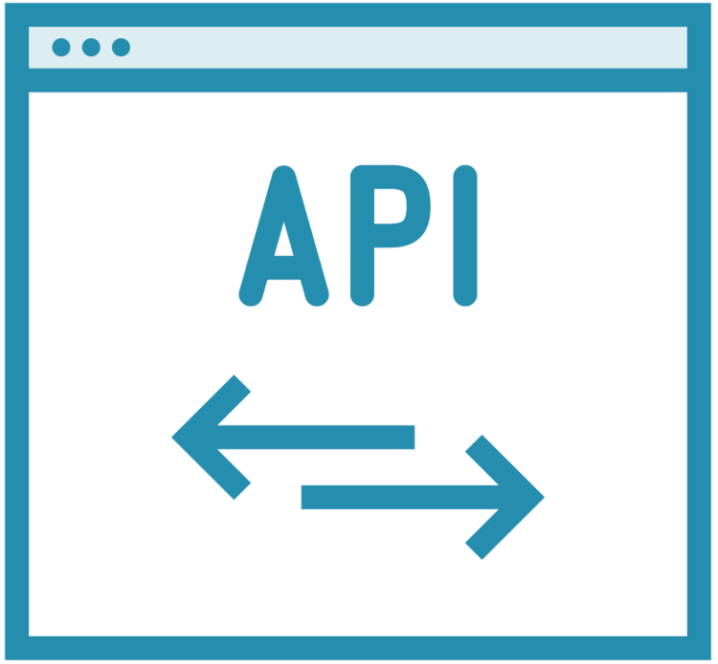
Demo

Implement the shopping cart

- Local storage

Working with a REST API

Storing REST API Data



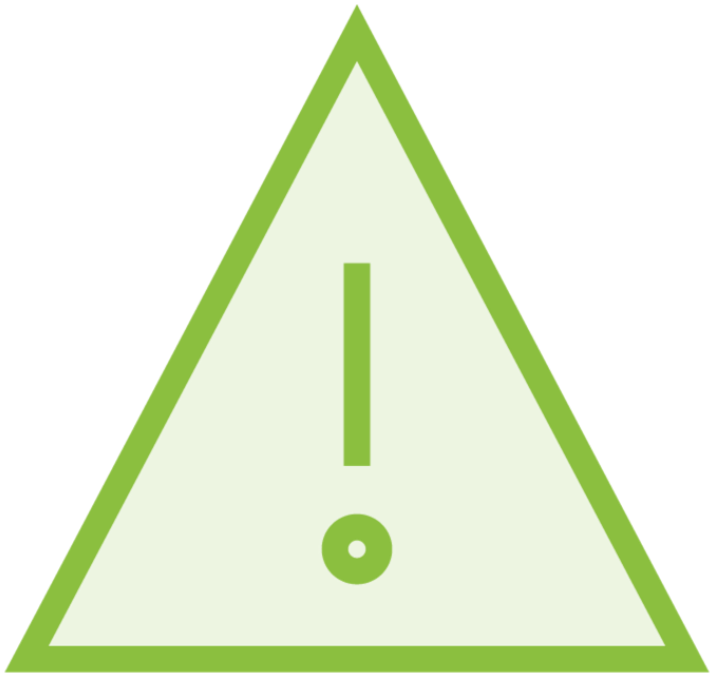
Data dependent

Reduces outbound network traffic

More responsive

Data stored as strings

Storage API Issues



- Functions are synchronous
- No access from web workers
- Vulnerable to XSS attacks
- Don’t store sensitive data

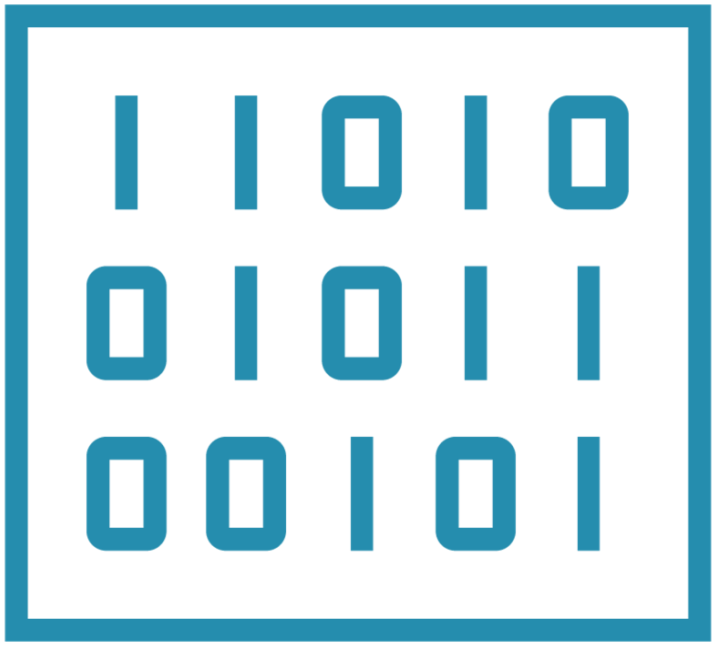
Saving Data as Strings

```
saveCart: function (object) {

    var stringified = JSON.stringify(object);
    localStorage.setItem(cartId, stringified);
    return true;

}
```

Reading Data

```
getCart: function () {

    return JSON.parse(localStorage.getItem(cartId));
}
```

Storing Complex Data



Challenging

- XML
- Images
- Files

Convert data with a parser

Consider performance

Use alternative storage for complex data

Demo

Retrieve product data from API

Store in local storage

Demo

Get product data from local storage

Summary

Explored the storage API

Implemented the shopping cart

Storing data

- Advantages
- Disadvantages

Store API data and display it