Objects

<dl>
<dt>initialize : <code>object</code></dt>
<dd> initializes all materialcss javascript
</dd>
<dt>regServiceWorker : <code>object</code></dt>
<dd> registers service worker into browser sessions
</dd>
<dt>firestore : <code>object</code></dt>
<dd> creating firestore connection
</dd>
<dt>vueObjects : <code>object</code></dt>
<dd> creating vue objects
</dd>
<dt>serviceWorker : <code>object</code></dt>
<dd> onfiguration and initialisation of the service worker (sw)
</dd>
</dl>

Functions

<dl>
<dt>clickFunctions()</dt>
<dd> Method to initialize all click functions
</dd>
<dt>clickFunctions()</dt>
<dd> Method to toggle dialog
</dd>
</dl>

initialize : <code>object</code>

initializes all materialcss javascript

Kind: global namespace

- initialize : <code>object</code>
    - .AutoInit()
    - .fixed-action-btn()
    - .datepicker()

initialize.AutoInit()

Method to auto-initialize all javascript elements

Kind: static method of <code>initialize</code>

initialize.fixed-action-btn()

Method to disable hover element of button

Kind: static method of <code>initialize</code>

initialize.datepicker()

Method to enable autoclose after a date is picked in datepicker

Kind: static method of <code>initialize</code>

regServiceWorker : <code>object</code>

registers service worker into browser sessions

Kind: global namespace

- regServiceWorker : <code>object</code>
    - .serviceWorker(serviceWorker)
    - .register()
    - .update()

regServiceWorker.serviceWorker(serviceWorker)

Kind: static method of <code>regServiceWorker</code>

Param	Type	Description
serviceWorker	<code>object</code>	detection test to make sure service workers are supported before trying to register one

regServiceWorker.register()

function to register the service worker for this site

Kind: static method of <code>regServiceWorker</code>
Throws:

- <code>error</code> catch rejected promise

Param	Type
service-worker.js	<code>javascript</code>

regServiceWorker.update()

updates reg. and cache with button click

Kind: static method of <code>regServiceWorker</code>

firestore : <code>object</code>

creating firestore connection

Kind: global namespace

- firestore : <code>object</code>
    - .config : <code>list</code>
    - .firestore : <code>object</code>
    - .firestoreConnection() ⇒ <code>object</code>
    - .Vueuse(VueFirestore)
    - .initializeApp(config)

firestore.config : <code>list</code>

force Vue to use VueFirestore.js for compatibility

Kind: static property of <code>firestore</code>

firestore.firestore : <code>object</code>

Kind: static constant of <code>firestore</code>

firestore.firestoreConnection() ⇒ <code>object</code>

Kind: static method of <code>firestore</code>
Returns: <code>object</code> - firestore

firestore.Vueuse(VueFirestore)

force Vue to use VueFirestore.js for compatibility

Kind: static method of <code>firestore</code>

Param	Type
VueFirestore	<code>object</code>

firestore.initializeApp(config)

Kind: static method of <code>firestore</code>

Param	Type
config	<code>list</code>

vueObjects : <code>object</code>

creating vue objects

Kind: global namespace

- vueObjects : <code>object</code>
    - .vueObjects()
    - .Vue()
    - .collection(collection, document)
    - .Vue()
    - .collection(collection, document)
    - .Vue()
    - .collection(collection, document)
    - .update(collection, document)

vueObjects.vueObjects()

creates vue objects to use in

Kind: static method of <code>vueObjects</code>

vueObjects.Vue()

vue object for coupons today

Kind: static method of <code>vueObjects</code>
Properties

Name	Type	Description
data	<code>object</code>	default data object
cafe	<code>number</code>	number of cafe coupons today
cake	<code>number</code>	number of cake coupons today
beverage	<code>number</code>	number of beverage coupons today
create	<code>object</code>	get data from database on creation

vueObjects.collection(collection, document)

get data from database (firestore)

Kind: static method of <code>vueObjects</code>
Throws:

- <code>error</code>

Param	Type	Description
collection	<code>string</code>	name
document	<code>string</code>	name

vueObjects.Vue()

vue object for coupons nextDays

Kind: static method of <code>vueObjects</code>
Properties

Name	Type	Description
data	<code>object</code>	default data object
cafe	<code>number</code>	number of cafe coupons nextDays
cake	<code>number</code>	number of cake coupons nextDays
beverage	<code>number</code>	number of beverage coupons nextDays
create	<code>object</code>	get data from database on creation

vueObjects.collection(collection, document)

get data from database (firestore)

Kind: static method of <code>vueObjects</code>
Throws:

- <code>error</code>

Param	Type	Description
collection	<code>string</code>	name
document	<code>string</code>	name

vueObjects.Vue()

vue object for wishes state

Kind: static method of <code>vueObjects</code>
Properties

Name	Type	Description
data	<code>object</code>	default data object
cafe	<code>number</code>	default state of cafe wish
cake	<code>number</code>	default state of cake wish
beverage	<code>number</code>	default state of beverage wish
create	<code>object</code>	get data from database on creation

vueObjects.collection(collection, document)

get data from database (firestore)

Kind: static method of <code>vueObjects</code>
Throws:

- <code>error</code>

Param	Type	Description
collection	<code>string</code>	name
document	<code>string</code>	name

vueObjects.update(collection, document)

method to push data to backend (eg. update data)

Kind: static method of <code>vueObjects</code>
Throws:

- <code>error</code>

Param	Type	Description
collection	<code>string</code>	name
document	<code>string</code>	name

serviceWorker : <code>object</code>

onfiguration and initialisation of the service worker (sw)

Kind: global namespace

- serviceWorker : <code>object</code>
    - .cacheName : <code>string</code>
    - .filesToCache : <code>array</code>
    - .install()
    - .fetch()
    - .activate()

serviceWorker.cacheName : <code>string</code>

Providing a cache name allows to version files, or separate data from the app shell.

Kind: static property of <code>serviceWorker</code>

serviceWorker.filesToCache : <code>array</code>

Array of all files that will be cached

Kind: static property of <code>serviceWorker</code>

serviceWorker.install()

open the cache with caches.open() and provide a cache name. call cache.addAll(), which takes an array and adds every item to the cache. caution: cache.addAll() is atomic!

Kind: static method of <code>serviceWorker</code>

serviceWorker.fetch()

caches.match() evaluates the web request that triggered the fetch event, and checks to see if it's available in the cache. It then either responds with the cached version, or uses fetch to get a copy from the network. Here it checks if network can be reached and if not, serves either a custom offline page, or the last cache available. The response is passed back to the web page with event.respondWith().

Kind: static method of <code>serviceWorker</code>

serviceWorker.activate()

To ensures that service worker updates its cache whenever any of the app shell files change. In order for this to work, you'd need to increment the cacheName variable at the top of your service worker file.

Kind: static method of <code>serviceWorker</code>

clickFunctions()

Method to initialize all click functions

Kind: global function

clickFunctions()

Method to toggle dialog

Kind: global function