

# Public JS API

## Screenshot\_2020-04-29\_at\_19.1 Getting started

This article contains information regarding access to Public JS API and examples of its usage.

Table of contents:

1. [API methods](#)
  1. [cart: TcApiCart](#)
  2. [localization: TcApiLocalization](#)
  3. [customer: TcApiCustomer](#)
2. [Examples](#)

## Screenshot\_2020-04-29\_at\_19.1 API methods

The API contains three main types of objects: **cart**, **localization** and **customer**. They are accessible through the **window.tcApi**. The object **window.tcApi** returns Promise with **TcApi** class instance.

### **cart: TcApiCart**

**getTotalValue(): Promise<number>** - method returns total value of cart.

**getProductsValue(): Promise<number>** - method returns total value of all products in cart.

**getProductsQuantity(): Promise<number>** - method returns quantity of products in cart (product \* quantity).

**getGiftsQuantity(): Promise<number>** - method returns quantity of gifts in cart.

**totalProducts(): Promise<number>** - method returns number of product rows in cart.

**totalGifts(): Promise<number>** - method returns number of gift rows in cart.

**getProductsSummary(): Promise<ProductSummary>** - method returns **ProductsSummary** object

```
interface ProductSummary {
  totalValue: number;
  productsValue: number;
  totalProductsQuantity: number;
  totalGiftsQuantity: number;
  totalProducts: number;
  totalGifts: number;
}
```

## localization: TcApiLocalization

**getLocale(): string** - current env locale. For example: **pl**, **cs** or **sk**.

**getCurrency(): string** - current locale currency. For example **zł** in PL environment.

**getCurrencyIso(): string** - current locale currency in **ISO 4217** standard. For example **PLN**.

**getDecimalDigits(): number** - number of decimal digits in prices.

**getMinGiftPrice(): number** - minimum price for a gift.

**getMobileRegex(): string** - mobile phone number regex pattern.

**getMobileDefaultPrefix(): string** - mobile phone number prefix. For example: **+48**.

**getMobilePlaceholder(): string** - mobile phone number placeholder. For example:  
**+48000000000**

**getPostalCodePlaceholder(): string** - postal code placeholder. For example: **00-000**.

**getPostalCodeRegex(): string** - postal code regex. For example: **^([0-9]{2}-[0-9]{3})\$**.

## customer: TcApiClientustomer

**isLoggedIn(): Promise<boolean>** - method returns authentication state of current customer.

**getFirstName(): Promise<string|null>** - method returns first name of customer if logged in.

**getState(): Promise<CustomerState>** - method return **CustomerState** object.

```
interface CustomerState {
  loggedIn: boolean;
  firstName: string|null;
}
```

# Screenshot\_2020-04-29\_at\_19.1

## Examples

Example usage of TC public JS API in CMS Blocks and CMS Pages.

Methods returning a Promise type may return the target value with a slight delay. This fact should be taken into account when designing views to avoid potential issues related to Cumulative Layout Shift or empty HTML elements.

### Fetch and display current cart value in span element

```
<script>
window.tcApi().then(async (api) => {
  if (api.localization === null) { /* Localization object can be null */
    return;
  }

  const cartValue = await api.cart.getTotalValue(); /* Example: 14.99 */
  const currency = api.localization.getCurrency(); /* Example: "zł" */
  const cartValueElement = document.querySelector('.cart-value');
  if (!cartValueElement) {
    return;
  }
  cartValueElement.innerHTML = `${cartValue} ${currency}`;
});
</script>
```

The value of your cart is </span>

### Check if user is logged in and display the appropriate banner based on their status

```
<script>
window.tcApi().then(async (api) => {
  const {loggedIn, firstName} = await api.customer.getState();

  const loggedInElement = document.querySelector('.logged');
  const unloggedElement = document.querySelector('.unlogged');
  if (!loggedInElement || !unloggedElement) {
    return;
  }
}
```

```
if (loggedIn && firstName) {
  loggedElement.style.display = 'block';
  unloggedElement.style.display = 'none';
  loggedElement.innerHTML = `Welcome ${firstName}`;
} else {
  loggedElement.style.display = 'none';
  unloggedElement.style.display = 'block';
}
});
</script>
```

```
<div class="logged" style="display: none;"></div>
<div class="unlogged" style="display: none;">Welcome guest!</div>
```

---

Revision #2

Created 27 November 2024 21:05:59 by Tau Ceti

Updated 27 November 2024 21:18:38 by Tau Ceti