

Index

General information	2
Initialization	3
Embedding script:.....	3
Initialization:.....	3
GDPR.....	4
Opt-in / opt-out	4
Handling form errors	5
Others	6
Initialization options (optional):	6
Events	7
Misc methods	8
Lead state	8
Misc	9

General information

New Leadz-script provides greater stability, it's feature-rich and it's easier to implement that old-one script.

Implementation of BUTTONS and FORMS hasn't changed – export and embed it as usual.

Initialization

Script url: <https://api.leadz.de/api/PublicScripts/leadz.js>

Embedding

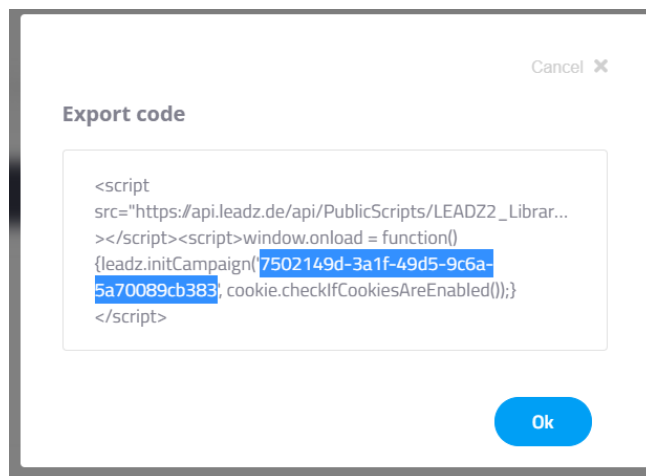
```
<script src="https://api.leadz.de/api/PublicScripts/leadz.js"></script>
```

Initialization

```
<script>
  (function () {
    leadz.initialize(
      "https://api.leadz.de/",
      "<CAMPAIGN ID>"
    );
  })();
</script>
```

<CAMPAIGN ID> has to be replaced by proper campaign id.

Campaign Id can be found in **Leadz -> Campaign -> Export**



Anyway, please DON'T use the whole export code (it provides obsolete script). Use scripts provided in this document.

It should doesn't matter if initialization will be executed in jquery "\$(document).ready()" function, "window.onload" or just like in the example.

GDPR

After initialization, default Leadz-script state is **"TRACKING-DISABLED"**. It's because user has to accept cookies first. In order to enable tracking method:

```
leadz.enableTracking()
```

has to be used.

Example:

```
<button onclick="leadz.enableTracking()">I accept cookies.</button>
```

After enabling tracing, proper cookie will be created and it won't be necessary anymore to enable tracing until the cookie expires.

Opt-in / opt-out

In order to opt-in / opt-out two methods may be used:

```
<button onclick="leadz.enableTracking()">Opt-in</button>  
<button onclick="leadz.disableTracking()">Opt-out</button>
```

After using each of the options, proper cookie will be created and selection will be remembered until cookie expires or becomes deleted.

Handling form errors

In order to better handling form errors- standard Leadz form script can be extended by adding proper error callback. In order to do this, standard “onsubmit” event has to be changed:

```
<form data-leadz-form-id="fa8c03d7-efc1-eb11-a7ad-00155d31294a" onsubmit="leadz.sendForm(event, onFormError)">
```

OnFormError is a method that accepts one argument – error.

Error API:

```
/** <summary>Error code. */
code: any;
/** <summary>Error message. */
message: string;
/** <summary>Error details. */
details: string;
/** <summary>Validation errors if exists. */
validationErrors: Array<ValidationErrorInfo>

/** ValidationErrorInfo */
/** <summary>Validation error message. */
message: string
/** <summary>Relate invalid members (fields/properties). */
members: Array<string>
```

Example:

```
<body>
  <form data-leadz-form-id="fa8c03d7-efc1-eb11-a7ad-00155d31294a" onsubmit="leadz.sendForm(event, handleFormError)">
    ...
  </form>
  <script>
    function handleFormError(error) {
      if (error && error.message)
        alert(error.message);
    }
  </script>
</body>
```

Others

Initialization options (optional):

Optional initialization options may be provided to initialization method:

```
/** Get / set debug flag. */
debug?: boolean;
/** Defines if lastly clicked button should be executed on login.
 */
callLastClickedAfterLogin?: boolean;
/** Defines if leadz items should be refreshed on DOM change. */
refreshItemsOnDomChange?: boolean;
/** Defines if leadz should fetch data from SalesCommunicator */
fetchScData?: boolean;
/** Defines how long should be "trackingEnabled" cookie expiration
date. */
trackingEnabledCookieExpirationDays?: number;
```

Usage sample:

```
leadz.initialize(
  "http://localhost:6234/",
  "a5833278-fcd1-4fc2-91c2-d768dbb64318",
  {
    debug: true,
    callLastClickedAfterLogin: true,
    trackingEnabledCookieExpirationDays: 365,
  }
);
```

Events

In order to react for some actions, events can be used.

```
/** Raised when Leadz finish initialization successfully. */
onInitialized?: () => void;
/** Raised when Lead logged-in successfully. */
onLoggedIn?: () => void;
/** Raised when Lead logged-out successfully. */
onLoggedOut?: () => void;
/** Raised when Lead tries to use Leadz button or form in case when trackingEnabled is false. */
onTrackingDisabledUseAttempt?: (leadzItem: IForm | IButton) => void;
/** Raised when Leadz tracking state changes. */
onTrackingStateChanged?: (trackingState: boolean) => void;
/** Raised when Lead activates successfully. */
onActivated?: () => void;
/** Raised when Lead doesn't activate properly. */
onActivationFailed?: (error: any) => void;
/** Raised when unhandled error occurs. */
onError?: (error: any) => void;
```

Usage example (use after initialization):

```
leadz.events.onTrackingDisabledUseAttempt = () => {
  alert("Leadz disabled!");
};
```

Misc methods

Lead state

Returns currently logged-in lead data

```
/** Login state. */
isLoggedIn: boolean;
/** Lead name. */
name?: string;
/** Lead surname. */
surname?: string;
/** Lead email address. */
emailAddress?: string;
/** Unique lead identifier. */
visitorIdentifier: string;
/** Lead campaign language. */
language?: string;
/** Lead language provided in registration in alpha2 code eg. "de" */
countryAlpha2?: string;
/** Lead current score */
globalScore: number;
/** SalesCommunicator lists to which lead belongs. */
scLists: Array<number>;
```

Example:

```
if (leadz.state.isLogged)
    alert("Hello " + leadz.state.name);
```

Misc

```
/** Returns opt-out state. */
leadz.isOptOut
/** Logging-out the Lead. */
leadz.logout()
```