

Shopware 6 Theme entwickeln – JavaScript

Shopware 6 Theme entwickeln – JavaScript

[expand title="mehr lesen..."]

You can add JavaScript to theme to interact with the DOM, make some API calls or change the behavior of the storefront.

By default, Shopware 6 look inside the `<plugin root>/src/Resources/app/storefront/src` folder of your plugin to load a `main.js` file.

You can simple put your own JavaScript code in here. Shopware 6 support the [ECMAScript 6](#) and will transpile your code to ES5 for legacy browser support.

The recommended way to add JavaScript is to write a JavaScript Plugin for the storefront.

Writing a JavaScript plugin

Storefront JavaScript plugins are vanilla JavaScript ES6 classes that extend from our Plugin base class. For more background information on JavaScript classes, take a look [here](#). To get started create a `src/Resources/app/storefront/src/example-plugin` folder and put an `example-plugin.plugin.js` file in there. Inside that file create and export a `ExamplePlugin` class that extends the

base Plugin class:

```
// src/Resources/app/storefront/src/example-plugin/example-plugin.plugin.js
```

```
import Plugin from 'src/plugin-system/plugin.class';
```

```
export default class ExamplePlugin extends Plugin {  
}
```

Each Plugin has to implement the `init()` method. This method will be called when your plugin gets initialized and is the entrypoint to your custom logic. In your case you add an callback to the `onScroll` event from the window and check if the user has scrolled to the bottom of the page. If so we display an alert. Your full plugin now looks like this:

```
// src/Resources/app/storefront/src/example-plugin/example-plugin.plugin.js
```

```
import Plugin from 'src/plugin-system/plugin.class';
```

```
export default class ExamplePlugin extends Plugin {  
  init() {  
    window.onscroll = function() {  
      if ((window.innerHeight + window.pageYOffset) >=  
document.body.offsetHeight) {  
        alert('seems like there\'s nothing more to see  
here.');      }  
    };  
  }  
}
```

Registering your plugin

Next you have to tell Shopware that your plugin should be loaded and executed. Therefore you have to register your plugin in the `PluginManager`. Open up `main.js` file inside your `src/Resources/app/storefront/src` folder and add the

following example code to register your plugin in teh PluginManager.

```
// src/Resources/app/storefront/src/main.js

// import all necessary storefront plugins
import ExamplePlugin from './example-plugin/example-plugin.plugin';

// register them via the existing PluginManager
const PluginManager = window.PluginManager;
PluginManager.register('ExamplePlugin', ExamplePlugin);
```

You also can bind your plugin to an DOM element by providing an css selector:

```
// src/Resources/app/storefront/src/main.js

// import all necessary storefront plugins
import ExamplePlugin from './example-plugin/example-plugin.plugin';

// register them via the existing PluginManager
const PluginManager = window.PluginManager;
PluginManager.register('ExamplePlugin', ExamplePlugin, '[data-example-plugin]');
```

In this case the plugin just gets executed if the HTML document contains at least one element with the data-scroll-detector attribute. You could use `this.el` inside your plugin to access the DOM element your plugin is bound to.

Loading your plugin

You bound your plugin to the css selector „`[data-example-plugin]`“ so you have to add DOM elements with this attribute on the pages you want your plugin to be active. Therefore create an `Resources/views/storefront/page/content/` folder and create an `index.html.twig` template. Inside this template extend `ExamplePlugin` from

the `@Storefront/storefront/page/content/index.html.twig` and overwrite the `base_main_inner` block. After the parent content of the blog add an `template` tag that has the `data-example-plugin` attribute. For more information on how template extension works, take a look [here](#).

```
{%                                                     sw_extends
'@Storefront/storefront/page/content/index.html.twig' %}

{% block base_main_inner %}
    {{ parent() }}

    <template data-example-plugin></template>
{% endblock %}
```

With this template extension your plugin is active on every content page, like the homepage or category listing pages.

More about Storefront JavaScript: [Storefront JavaScript reference documentation](#)

[/expand]