Introduction

We built Storyblok with a robust and flexible plugin system to give our customers the power to extend Storyblok and the visual editor.

There are three different ways in which you can extend Storyblok: Custom Applications, Field-types or Tools.

Field-types

Field-types allow you to extend the visual editor with custom fields like a color picker or a google maps location selector.

Custom Applications

Custom Applications are the biggest of all extensions types and are the right choice if you want to extend Storyblok with some complex functionality like a Dashboard.

They will have their own location in the Space sidebar and a big window to build your own functionality similar to the popular Tasks app.

Tools

Tools are a different way to extend your visual editor. They allow you to improve your productivity by adding some functionality to the editor. Two examples for Tools are importing and exporting content or a custom clipboard.
Getting Started

To create a custom sidebar application you need to join the Storyblok partner program.

Become a Partner

Creating an Extension

Once you’re in the partner portal click on Apps to create a new extension.

Installing an Extension

Custom Applications and Tools are always specific to a Storyblok space. To get access to these extensions, you first need to install them in one of your spaces.

App Authentication

In order to give your local extensions (apps and tools) access to the Storyblok space, we need to authenticate them inside of Storyblok. This will happen the first time you open a story once you installed the tool or app.
CheatSheet - Storyblok Extensions

Custom Apps
Custom Application

Storyblok allows you to create custom apps with their own big window. This is great for creating Dashboards or complex functionality in Storyblok.

HINT: You can start with the preconfigured example Nuxt app: https://github.com/storyblok/storyblok-workflow-app

Creating an App in Storyblok

Once you’re in the partner portal click on Apps to create a new Sidebar app.

CREATE

Authenticating your sidebar app

In order to use your app inside of the Storyblok App you need to fill the following .env variables inside your project from your Storyblok App settings in order to authenticate the application.

BASE_URL=http://localhost:3000
CONFIDENTIAL_CLIENT_ID= “copy from Storyblok”
CONFIDENTIAL_CLIENT_SECRET= “copy from Storyblok”
CONFIDENTIAL_CLIENT_REDIRECT_URI= “{ngrok url}/auth/callback”

Client id
8fk63RqBz7ooHRi40pb3Q==

Secret
7t7+AcVqYzek6yf4bCpubbLhyXkyHf1GLKr6

Create a NGROK Tunnel

To expose the local application to the Storyblok App start a ngrok tunnel to your localhost.

ngrok http 3000

Then fill the created tunnel URL into your app settings in the partner portal.

Live
Url to your app
https://90327b19f0ee.ngrok.io/auth/connect/storyblok

OAuth2 callback url
https://90327b19f0ee.ngrok.io/auth/callback
CheatSheet - Storyblok Extensions

Field-types
Field-types

Storyblok allows you to create custom field-types to enhance the Storyblok editor. An example for such a field-types would be a custom icon picker.

HINT: Learn all about field-types in our cookbook. storyblok.com/docs/plugins/field-type

Creating a field-type in Storyblok

Once you’re in the partner portal click on Field-types to create a new custom field-type.

Field-type editor

Storyblok includes its own editor to create and inspect new field-types. Custom field-types are created as Vue components with a few helpers.

```javascript
const Fieldtype = {
  mixins: [window.Storyblok.plugin],
  template: `<input v-model="model.example" />`,
  methods: {
    initWith() {
      return {
        plugin: 'example_plugin',
        example: 'Hello world!'
      }
    },
  },
}
```

Local Development Mode

If you want to create more complex field-types you have the option to develop in a local development environment and still preview the output in the Storyblok Editor. Start with the example repository:

```
$ git clone https://github.com/storyblok/storyblok-fieldtype
$ cd storyblok-fieldtype
$ npm install
$ npm run dev
```

Follow these steps to develop locally:

1. Change URL from https to http
2. Open the Plugin.vue file and change the plugin name to the one you created in Storyblok
3. Click “Enable local dev mode” in the Storyblok Editor to preview the field-type
4. Make sure your app name in Plugin.vue is correct, then run `npm run build`
5. Copy the content of export.js into the Storyblok editor to import your field-type and use it
Expose values to the Storyblok Api

Once your field-type is ready, you can use a Vue watcher with the $emit method to expose a specific value to Storyblok.

```javascript
const Fieldtype = {
  watch: {
    'model': {
      handler: function (value) {
        this.$emit('changed-model', value);
      },
      deep: true
    }
  }
}
```

HINT: You can also track changes of a story inside the field-type. Read: storyblok.com/faq/plugin-track-storyitem-changes

You will see what your field-type outputs inside the Storyblok field-type editor:

```
{ "plugin": "rating-star", "value": 0, "_uid": "1aa839ac-e295-4b6b-af31-fa9454bf644" }
```
Tools
Setup

Storyblok tools are specific to stories and will appear in your Visual Editor when editing a story.

Authenticating your sidebar app

In order to use your tool inside of the Storyblok App you will need to add your clientID and clientSecret to the .env file. Furthermore you will need an ngrok tunnel and add this url to the .env as well.

```
CONFIDENTIAL_CLIENT_ID= “copy from Storyblok”
CONFIDENTIAL_CLIENT_SECRET= “copy from Storyblok”
CONFIDENTIAL_CLIENT_REDIRECT_URI= “{ngrok url}/ auth/callback”
```

Create a NGROK Tunnel

To expose the local application to the Storyblok App start a ngrok tunnel to your localhost.

```
ngrok http 3000
```

Then fill the created tunnel URL into your App settings in the partner portal.

HINT: Follow this detailed tutorial to learn about the authentication: storyblok.com/tp/how-to-create-a-oauth2-authentication-flow-with-koa

Creating a Tool App in Storyblok

Once you’re in the partner portal click on Apps to create a new tool app and select App type Tool.

HINT: You can start with the preconfigured example Nuxt tool: github.com/storyblok/storyblok-tool-example

HINT: You can start with the preconfigured example Nuxt tool: github.com/storyblok/storyblok-tool-example

HINT: You can start with the preconfigured example Nuxt tool: github.com/storyblok/storyblok-tool-example

HINT: You can start with the preconfigured example Nuxt tool: github.com/storyblok/storyblok-tool-example

HINT: You can start with the preconfigured example Nuxt tool: github.com/storyblok/storyblok-tool-example

HINT: You can start with the preconfigured example Nuxt tool: github.com/storyblok/storyblok-tool-example

HINT: You can start with the preconfigured example Nuxt tool: github.com/storyblok/storyblok-tool-example

HINT: You can start with the preconfigured example Nuxt tool: github.com/storyblok/storyblok-tool-example

HINT: You can start with the preconfigured example Nuxt tool: github.com/storyblok/storyblok-tool-example

HINT: You can start with the preconfigured example Nuxt tool: github.com/storyblok/storyblok-tool-example

HINT: You can start with the preconfigured example Nuxt tool: github.com/storyblok/storyblok-tool-example

HINT: You can start with the preconfigured example Nuxt tool: github.com/storyblok/storyblok-tool-example

HINT: You can start with the preconfigured example Nuxt tool: github.com/storyblok/storyblok-tool-example

HINT: You can start with the preconfigured example Nuxt tool: github.com/storyblok/storyblok-tool-example

HINT: You can start with the preconfigured example Nuxt tool: github.com/storyblok/storyblok-tool-example

HINT: You can start with the preconfigured example Nuxt tool: github.com/storyblok/storyblok-tool-example

HINT: You can start with the preconfigured example Nuxt tool: github.com/storyblok/storyblok-tool-example

HINT: You can start with the preconfigured example Nuxt tool: github.com/storyblok/storyblok-tool-example