

# React Burger Tutorial:

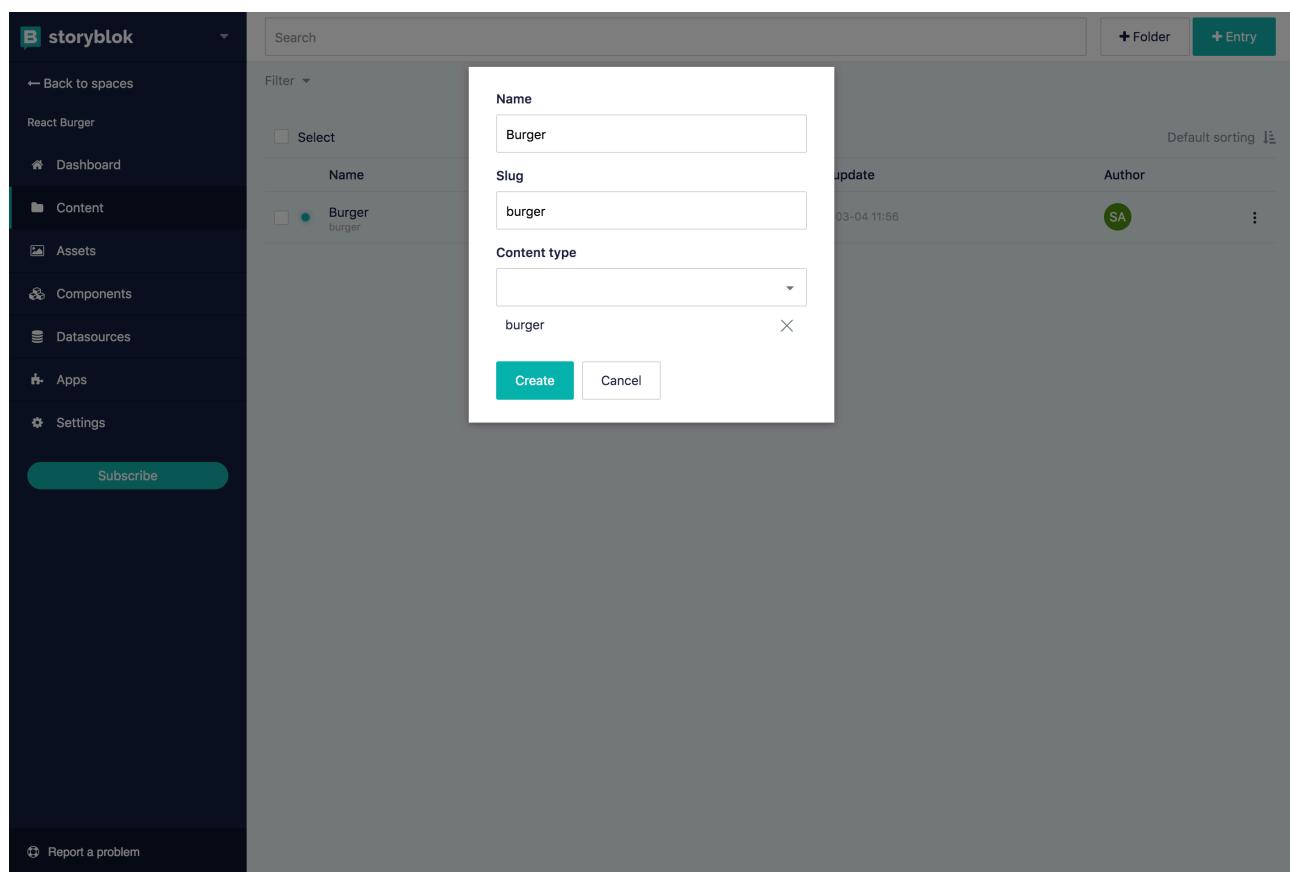
## Create Content In Storyblok:

### 1. The Burger:

In Content section (left column), add new entry (top right, green button).

Set Name, Slug, and Content type to **burger**.

These would typically be set to the name of the content you would like to create (The name of a page for example).



### 2. The Slices/Ingredients:

You will be creating 5 “blocks” which will dictate the data that will be provided to your components in your React App.

Click on “Add block” in the right column.

Create the following blocks:

- Top Bun
- Vegetables
- Cheese
- Patty
- Bottom Bun

It should look like this:

The screenshot shows the Webpacker interface for a 'Burger' project. On the left, there's a sidebar with navigation links like 'Dashboard', 'Create', 'My projects', 'Logout'. The main area has a header 'Burger' and a sub-header 'burger'. Below the header, there's a URL input field 'http:// Your server address /burger'. The main content area is divided into sections: '1. Download a preconfigured project' (with a note about preconfiguration and an API token), 'Preferred by our users' (listing Nuxt.js, Next.js, and React/Gatsby with download and tutorial buttons), 'Other technologies' (listing Plain JS, Vue.js, Node.js, PHP, and Ruby), and '2. Start your server' (with a URL input field and a 'Go' button). On the right, there's a 'Content' tab and a configuration panel for the 'Burger' project. This panel includes tabs for 'Content', 'Status', 'Config', and 'Tools'. Under 'Content', there's a tree view of project components: 'Body' (with 'Select all'), 'Top Bun', 'Vegetables', 'Cheese' (with 'cheese.svg'), 'Patty' (with 'patty.svg'), and 'Bottom Bun'. A red box highlights this configuration area.

Each block will be configured with one or more “Key Values”, when you create a block, you will be asked to enter them:

This screenshot shows the same 'Burger' project setup as above, but with a focus on configuring a specific block. A red box highlights the 'Define schema' section in the configuration panel. In the 'Schema (3)' tab, there's a search bar 'Enter a key value' and a '+ Add' button. Below it, there's a note: 'Examples: news\_items, body, columns, title, description or call\_to\_action\_btn.' To the right, there's a 'General' tab with two items: 'TopBun' and 'TopBunColour'. The 'TopBun' item has a small icon, a label, and edit/delete buttons. The 'TopBunColour' item has a minus sign, a label, and edit/delete buttons.

Use these key values for each block:

- Top Bun:

**TopBun**, type: text

The screenshot shows the 'Define 'TopBun'' schema editor. On the left, there's a preview of a landing page with sections for downloading a preconfigured project and starting a server. On the right, the schema configuration is detailed:

- Type:** Text
- Maximum characters:** (empty input)
- Required:** (unchecked)
- Regex validation:** (empty input)
- Description:** (empty input)
- Display name:** Top Bun
- Default value:** (empty input)
- Enable RTL (Right to left):** (unchecked)
- Renaming:** (button to change field name)

**TopBunColour**, type: Single-Option, options: (see screenshot below)

The screenshot shows the 'Define 'TopBun'' schema editor. The configuration for the 'TopBunColour' field includes a table of color options:

Value	Color
= red	#ea4435
= green	#34a854
= blue	#4285f3
= transparent	none

Other configuration fields include:

- Type:** Single-Option
- Source:** Self
- Required:** (unchecked)
- Description:** (empty input)
- Display name:** Top Bun Colour
- Default value:** (empty input)

- Vegetables:

**Vegetables**, type: Multi-Assets

The screenshot shows the configuration interface for a 'Vegetables' asset. On the left, there's a preview area with a header 'Burger burger' and a URL input field 'http:// Your server address /burger'. Below this is a section titled '1. Download a preconfigured project' which includes a note about a preconfigured package and an API token. It lists 'Preferred by our users' projects like Nuxt.js, Next.js, and React/Gatsby, each with a download and tutorial button. Under 'Other technologies', it lists Plain JS, Vue.js, Node.js, PHP, and Ruby. A '2. Start your server' section has a URL input field and a 'Go' button. On the right, the 'Define 'Vegetables'' panel shows the schema configuration: Type is set to 'Multi-Assets', and the 'Edit field' tab is selected. Other tabs include 'Schema (1)', 'Config', and 'Save schema'. The 'Description' field contains 'These are the vegetables!', 'Display name' is 'Vegetables', and 'Renaming' is set to 'Change field name'.

- Cheese:

**Cheese**, type: Image

The screenshot shows the configuration interface for a 'Cheese' asset. The layout is identical to the 'Vegetables' page, with a preview area on the left and a schema definition panel on the right. The schema configuration for 'Cheese' includes a 'Type' dropdown set to 'Image', and the 'Edit field' tab is selected. Other tabs include 'Schema (1)', 'Config', and 'Save schema'. The 'Default asset folder' section is present but empty. The 'Description' field is empty, 'Display name' is 'Cheese', 'Default value' is empty, and 'Renaming' is set to 'Change field name'.

- Patty:

**Patty**, type: Image

The screenshot shows the 'Burger' UI interface. On the left, there's a sidebar with a back arrow and the text 'Burger'. The main content area has a header 'http:// Your server address /burger'. Below this, there are two sections: '1. Download a preconfigured project' and '2. Start your server'. The first section contains a note about a preconfigured package and links to download and view tutorials for Nuxt.js, Next.js, and React/Gatsby. The second section has a text input for the server URL and a 'Go' button. To the right, a 'Define 'Patty'' panel is open, showing a schema with one item: 'General' containing a key-value pair 'Patty' with a file icon.

- Bottom Bun:

**BottomBun**, type: Text

The screenshot shows the 'Burger' UI interface. The layout is identical to the 'Patty' screen, with sections for downloading projects and starting a server. The 'Define 'BottomBun'' panel on the right shows a schema with two items: 'General' containing 'BottomBun' and a collapsed section 'BottomBunColour'.

## **BottomBunColour**, type: Single-Option

The screenshot shows the Webflow interface for creating a preconfigured project. On the left, there's a preview of the landing page with download links for Nuxt.js, Next.js, and React/Gatsby. On the right, the 'Edit field' tab for 'BottomBun' is open, showing it's a 'Single-Option' type with three color choices: red (#ea4435), green (#34a854), and blue (#4285f3). A 'Required' checkbox is also present.

### **3. Entering our data in the blocks/components:**

All the files are included in the project you cloned from the GitHub repository (<https://github.com/Digital-E/reactburger.git>)

The file path is: ./src/reactburgerimages/

Locate them and keep them handy for this part, we'll be needing them.

We are going to add data to each of the blocks we created:

- Top Bun

#### **Top Bun**

Add the .svg code into the text field. We are adding the .svg code as is, so we can then change the colour of the .svg in our React app by changing the fill.

#### **Top Bun Colour**

Chooses the colour of the bun (this can be changed at any time).

1. Download a preconfigured project  
The downloaded package comes preconfigured with your api token: `hE4BLg0qed3XBNGJcALtrAtt`

Preferred by our users

- NUXTJS
- NEXT.js
- React/Gatsby

Other technologies

- Plain JS
- Vue.js
- Node.js
- PHP
- Ruby

2. Start your server  
Start the server of the downloaded package and insert your server url here:

`http://` Example: localhost:8080 or yourwebsite.com `Go`

FAQ

Content Status Config Tools

Top Bun `<svg xmlns="http://www.w3.org/2000/svg" v`

Top Bun Colour green

## - Vegetables

### Vegetables

Upload the onions, tomato, and salad.  
These will be uploaded as files, as we won't be changing their colour.

1. Download a preconfigured project  
The downloaded package comes preconfigured with your api token: `hE4BLg0qed3XBNGJcALtrAtt`

Preferred by our users

- NUXTJS
- NEXT.js
- React/Gatsby

Other technologies

- Plain JS
- Vue.js
- Node.js
- PHP
- Ruby

2. Start your server  
Start the server of the downloaded package and insert your server url here:

`http://` Example: localhost:8080 or yourwebsite.com `Go`

FAQ

Content Status Config Tools

Vegetables `<img alt="onions.svg" />` `<img alt="tomato.svg" />` `<img alt="salad.svg" />`

onions.svg Description  
tomato.svg Description  
salad.svg Description

Upload new Asset manager

These are the vegetables!

## - Cheese

### Cheese

The screenshot shows the Burger interface with the project 'Cheese' selected. The left panel displays a landing page with sections for downloading preconfigured projects (Vue.js/Nuxt.js, React/Next.js, React/Gatsby) and starting a server. The right panel shows the project details for 'Cheese', including its schema and assets.

**Content** Status Config Tools

**Cheese**

Burger > Cheese

cheese.svg

//a.storyblok.com/f/77303/300x94/01ff9t

Upload new Asset manager

1 items out of schema

- Patty

## Patty

The screenshot shows the Burger interface with the project 'Patty' selected. The left panel displays a landing page with sections for downloading preconfigured projects (Vue.js/Nuxt.js, React/Next.js, React/Gatsby) and starting a server. The right panel shows the project details for 'Patty', including its schema and assets.

**Content** Status Config Tools

**Patty**

Burger > Patty

patty.svg

//a.storyblok.com/f/77303/300x114/c414r

Upload new Asset manager

FAQ

- Bottom Bun

## Bottom Bun

Burger

burger

Save Publish

Content Status Config Tools

Bottom Bun Define schema

Copy Burger > Bottom Bun

Bottom Bun

<svg xmlns="http://www.w3.org/2000/svg" v

Bottom Bun Colour

green

1. Download a preconfigured project

The downloaded package comes preconfigured with your api token: hE4BLg8qed3XBNGJcALtrAtt

Preferred by our users

NUXTJS Vue.js/Nuxt.js  Tutorial

NEXT React/Next.js  Tutorial

React/Gatsby  Tutorial

Other technologies

Plain JS  Vue.js  Nodejs  PHP  Ruby 

2. Start your server

Start the server of the downloaded package and insert your server url here:

http:// Example: localhost:8080 or yourwebsite.com Go

FAQ

---

We're Done!

Now click on Save, and Publish!