# Installing the NextJS Starter App with the CLI

Installing and Configuring the NextJS Starter App

About Starter Apps
In Contentstack, starter apps are pre-built applications that demonstrate the capabilities of Contentstack and provide a starting point for developers learning about and building web applications.

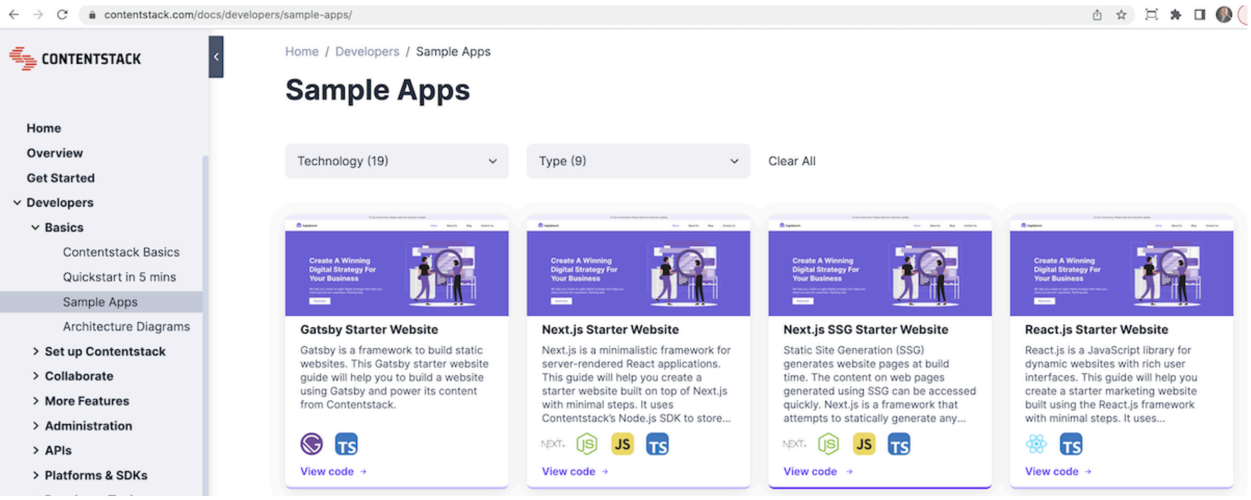
Starter apps are designed to be fully functional, customizable, and extensible, providing developers with a set of best practices and common patterns for building web applications that consume content from Contentstack.

Starter apps in Contentstack are available for a variety of programming languages and frameworks, including NextJS, React, Angular, and Vue.js. Each starter app includes a set of pre-built content types, entries, assets, and environments.

Using starter apps in Contentstack provides several benefits, including:

- Rapid prototyping: Starter apps provide a quick way to prototype and test new ideas, allowing developers to create a fully functional application in a fraction of the time it would take to build from scratch.
- Best practices: Starter apps are built using best practices and industry-standard patterns, providing developers with a solid foundation for building scalable, maintainable, and high-performing applications.
- Customization: Starter apps can be fully customized to meet the specific needs of the application being developed, including branding, layout, functionality, and integrations with other systems.
- Extensibility: Starter apps can be extended and allowing developers to add new features and functionality as needed.

Overall, starter apps in Contentstack provide a powerful and flexible way for developers to jumpstart their projects, reduce development time and costs, and ensure high-quality, scalable, and maintainable applications.



Sample Apps

**Gatsby Starter Website**

Gatsby is a framework to build static websites. This Gatsby starter website guide will help you to build a website using Gatsby and power its content from Contentstack.

**Next.js Starter Website**

Next.js is a minimalistic framework for server-rendered React applications. This guide will help you create a starter website built on top of Next.js with minimal steps. It uses Contentstack's Node.js SDK to store...

**Next.js SSG Starter Website**

Static Site Generation (SSG) generates website pages at build time. The content on web pages generated using SSG can be accessed quickly. Next.js is a framework that attempts to statically generate any...

**React.js Starter Website**

React.js is a JavaScript library for dynamic websites with rich user interfaces. This guide will help you create a starter marketing website built using the React.js framework with minimal steps. It uses...

## Three ways to get started with the StarterApps

When starting to develop with Contentstack, we can install and use an existing NextJS Starter App called Logoipsum and this can be set up in several ways:

1. The first approach involves using an automated bootstrap CLI script to generate a new Stack or load content into an existing Stack. This method includes all the default content and downloads the NextJS app. We will be using this option for the course.

2. The second approach involves installing a Starter App via the Marketplace. This method is useful when you want to try using Contentstack as an Author or Editor and publish content without accessing the frontend code. However, we will not be using this approach in the course.

3. Manually import the content with the CLI and download the StarterApp from the Github repo and manually configure the application.  This does the same thing as step 1, but requires more time.

## Install the Starter App

### Create the Stack

To get started we will need to create a new Stack in Contentstack for our NextJS Starter app.  To create our Starter App using the Contentstack CMS, we need to follow a few steps. Firstly, we need to create a new Stack and then use a CLI command to import the Starter app content and code. It is crucial to ensure that you have an Empty Stack ready for the StarterApp before proceeding. If you have the role of Organization Owner in Contentstack, then the CLI can create the Stack for you automatically using your login credentials.

Watch the following video to explain how to install the StarterApp with the CLI

## What's inside the Starter App

The Starter app Stack contains all the Content Types, Entries, Assets, Environments, Tokens, and Roles for a full website.  The following video will explain the Starter app and where everything is located:

# Architecture

## Big Picture Architecture Diagram

### Backend

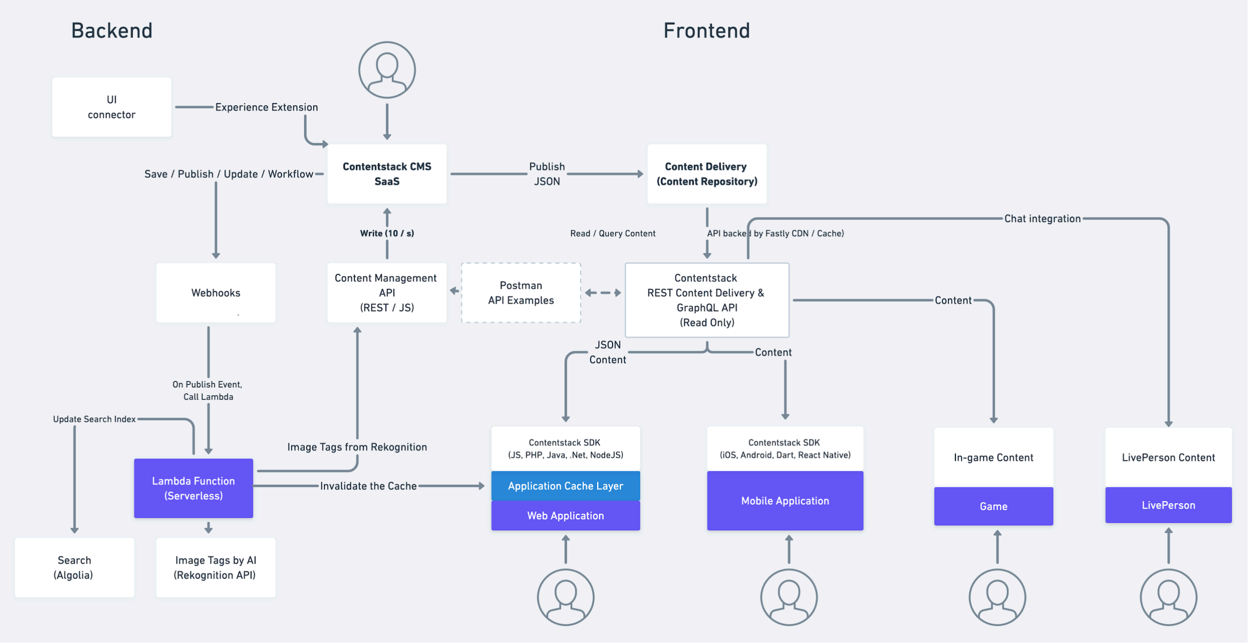
The Backend of Contentstack is the CMS with the ability to create and publish content.  The focus here is on editor ease-of-use by providing a simple, powerful and flexible UI for content creation and management.  In addition to the out of the box functionality, we also provide an ability to extend the interface by creating UI

Extensions using our App SDK and connect to 3rd-party applications such as DAM providers, Analytics, E-Commerce platforms and more.

## Frontend

The focus on the frontend is high-performance for your website and web application. APIs and SDKs are provided to the developer for using in their applications to perform read-only queries to the content.  The Content Delivery APIs use an internal CDN to cache all queries to the platform which results in higher performance for all your web applications.  This is also where we find the Images API for delivering image assets.

Watch this video that explains the big picture architecture:



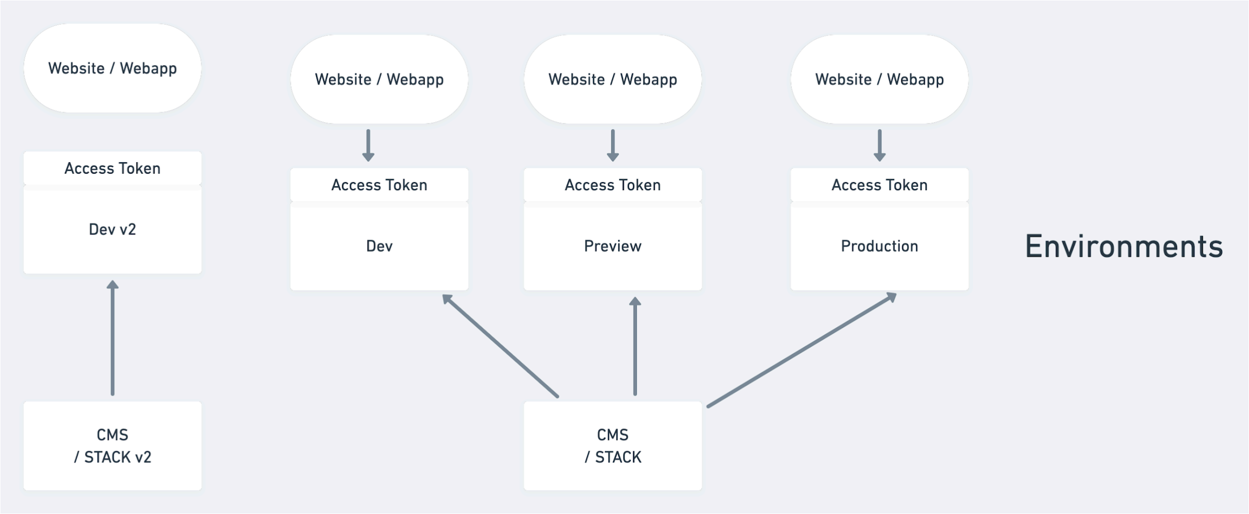
Backend
Frontend
UI connector
Experience Extension
Contentstack CMS SaaS
Save / Publish / Update / Workflow
Publish JSON
Content Delivery (Content Repository)
Write (10 / s)
Read / Query Content
API backed by Fastly CDN / Cache)
Chat integration
Webhooks
Content Management API (REST / JS)
Postman API Examples
Contentstack REST Content Delivery & GraphQL API (Read Only)
Content
JSON Content
Content
On Publish Event, Call Lambda
Update Search Index
Image Tags from Rekognition
Lambda Function (Serverless)
Invalidate the Cache
Contentstack SDK (JS, PHP, Java, .Net, NodeJS)
Application Cache Layer
Web Application
Contentstack SDK (iOS, Android, Dart, React Native)
Mobile Application
In-game Content
Game
LivePerson Content
LivePerson
Search (Algolia)
Image Tags by AI (Rekognition API)

# Environments

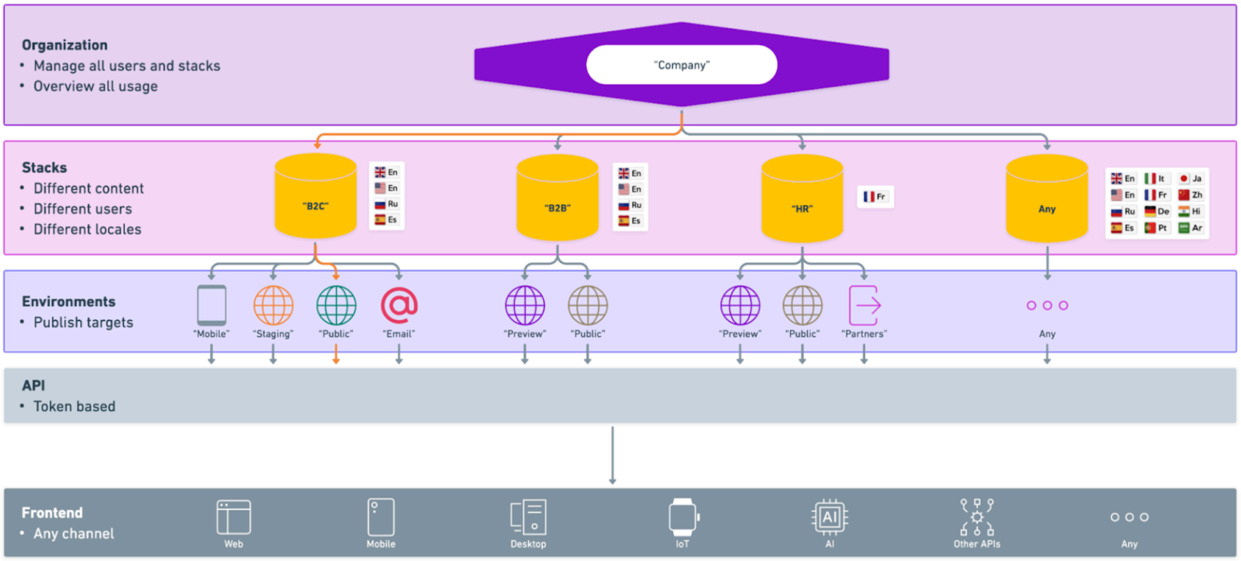
- One Stack can publish to multiple Environments.  Think of Environments as “publish destinations”.
- Each Environment has one Delivery Token, otherwise known as an Access Token.  That token is unique per Environment.
- Each Stack has an API key.  There is only 1 API key per Stack and it is unique.
- Most API calls will require both an API token and a Delivery / Access Token.

Watch this video to understand how we use Environments in Contentstack:

The following diagram for Environments illustrates these concepts:



## Multi-Stack Architecture Diagram



**Organization**

- Manage all users and stacks
- Overview all usage

**Stacks**

- Different content
- Different users
- Different locales

**Environments**

- Publish targets

**API**

- Token based

**Frontend**

- Any channel