Building commercetools
customizations and extensions

Table of Contents

Introduction

> *"The most important capability of any ecommerce system for a large company is customizability. The CDOs [Chief Digital Officer] we spoke with were largely not interested in turn-key solutions. These entities prefer to have complete creative control over the environment their customers encounter."*
>
> – [The New Chiefs of Commerce in the Digital Enterprise](#), page 10

commercetools is a dynamically extensible, cloud-native commerce solution. It allows retailers to sculpt a solution that fits their unique needs today, and is flexible to support their evolving business strategy tomorrow.

In this whitepaper, we:

1. Describe the various ways you can customize and extend functionality on the commercetools platform;
2. Provide a flowchart to help you decide which technique is best for your use case;
3. Share tutorial links to help you get started with each customization option.

Customizations versus Extensions

Before we get started, it's important to understand the difference between customizations and extensions in the context of commercetools.

Customizations are achieved by leveraging features built into commercetools to enhance its functionality and customize it based on your unique needs. You can leverage commercetools' Custom Fields and Custom Objects to perform this level of customization and we cover both in this paper.

Extensions are achieved by integrating solutions that run externally from the commercetools platform. Extension solutions can come in many forms: custom microservices you build yourself, commercial solutions you've licensed, open source projects you want to leverage, and, of course, commercetools Marketplace solutions. We briefly touch on the Marketplace in this paper but all sorts of extensions are possible because of the API-based design of commercetools: it allows for a rich ecosystem of loosely coupled solutions to co-exist and interoperate.