



Building commercetools customizations using Custom Objects

Table of Contents

Introduction	3
What are Custom Objects?	3
Implementation Example	4
Use Case	4
Problem	4
Solution	4
Implementation Steps	4
Step A: Create a commercetools project.	5
Step B: Load sample data.	5
Step C: Create a new Type to define the Custom Field.	6
Step D: Create a Cart.	7
Create	7
Read	9
Update	10
Delete	10
Step E: Create a new Type to define the cart Custom Field.	11
Step F: Create a Cart.	12
Step G: Update the Cart to add Features used.	13
Step H: Test: Query to find all carts that used a specific feature.	15
Step I: Install the Sunrise SPA Front End.	17
Step J: Extend the front end to use our Custom Objects.	18
Step K: Provide you with resources if you need help.	23
Additional Help	24



Introduction

“The most important capability of any ecommerce system for a large company is customizability. The CDOs 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.

There are many powerful extensibility features built into commercetools that handle a wide variety of use cases. For an overview of them, see [Building commercetools customizations - Overview](#).

In this whitepaper we will do a deep dive on one powerful technique for customizing commercetools: Custom Objects.

What are Custom Objects?

[Custom Objects](#) allow you to create a new resource that you would like to store within commercetools and integrate with other aspects of the system.

“Custom objects are a way to store arbitrary JSON-formatted data on the commercetools platform. It allows you to persist data that does not fit the standard data model. This frees your application completely from any third-party persistence solution and means that all your data stays on the commercetools platform”

– [Custom Objects](#), Platform Documentation, commercetools

For instance, you may have an entity that is specific to your business domain and does not correspond to an existing commercetools resource. In this case, you can define the entity within commercetools using a Custom Object.

You can also use Custom Objects in combination with [Custom Fields](#). For example, you could extend the `customer` resource with a Custom Field that references a complex data type stored in a custom object. For a detailed tutorial on Custom Fields, see [Building commercetools customizations - Custom Fields](#).

Bottom line, Custom Objects provide a lot of power. In fact they're so powerful

