



# Building commercetools customizations using API Extensions and the Google Cloud Platform

# Table of Contents

---

<b>Introduction</b>	<b>3</b>
<b>What are API Extensions?</b>	<b>3</b>
<b>Implementation Example</b>	<b>4</b>
Use Case	4
Problem	4
Solution	5
Implementation Steps	5
Step A: Create a commercetools Project.	5
Step B: Create commercetools Discounts.	5
Step C: Set up a GCP Project.	9
Set up a Google Cloud Account	9
Set up a Google Cloud Project	10
Install the Google Cloud SDK	10
Authenticate using Google Cloud SDK	10
Step D: Create a GCP Cloud Function.	10
Step E: Configure and Deploy the Cloud Function.	13
Step F: Set up commercetools API Extension.	14
Step G: Test commercetools/GCP Integration.	15
Step H: Provide you with resources if you need help.	16
<b>Additional Help</b>	<b>17</b>



# 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: API Extensions.

## What are API Extensions?

[API Extensions](#) allow you to modify the response of a commercetools API call to customize behavior.

*“The commercetools platform provides default data structures and default behavior that is useful for many of the customers. However, each project has its unique requirements. Similar to data structures that can be customized with Custom Types and Product Types, additional behavior can be added. For behavior that needs to be executed before the API call succeeds, API Extensions can be used.”*

– [API Extensions](#), Platform Documentation, commercetools

You may have unique business rules that must be taken into account for cart validation, cart discounts, shipping cost calculations, etc. API Extensions provide a mechanism for these business rules to be applied *within the context of a commercetools API's execution*. When commercetools receives an API call for an endpoint you've extended, it first performs standard commercetools behaviors on the resource and then calls your code for further business rule processing. Once it gets your updates back, the result is persisted.

