# Introduction to the REST API and DTOs

In this lesson we will define the REST API for our Catalog microservice.

## Catalog REST API

The REST API defines the operations exposed by the microservice. Since the Catalog microservice owns the items that players will be able to purchase, here are the operations that it should support:

- Get items, which retrieves all the items currently stored in the service
- Get item by id, which retrieves the details of a specific item
- POST item, which creates an item in the service
- PUT item, which updates a specific item
- And DELETE item, which deletes the specified item

As part of the definition of the REST API we will also need to define the relevant Data Transfer Objects.

## Data Transfer Objects

What is a Data Transfer Object? A Data Transfer Object, also known as DTO is an object that carries data between processes. For example:

Let’s go back to our client and the catalog service. When the client needs to retrieve the details for a specific item it will send a GET request to the service with the item id, and the service will in turn return the item details. This payload returned by the service is what we call the DTO.

The DTO represents the contract between the microservice and the client. Modifying this contract has significant implications. Image what would happen if we decide to rename the Price to ItemPrice? Our clients would break and they would need to get updated right away to ensure no disruption to our customers.

As a general rule you need to think carefully about each of your DTOs to ensure you minimize the need to update them later as much as possible.