

Relewise Developer Exercises

For any of these exercises, it is completely up to you to choose any Host Environment, to run the code. Whether it is Web, Console, Unit Test, or even through External IDE's like LINQPad is not important for the exercises.

If you are stuck – don't worry – just try the next exercise – or see if tools like ChatGPT can help you further.

Exercise 1: Download Raw Product Data and Map to Relewise Products

- Write a C# Class, that implements the `IJob` -interface (see *Appendix-section*)
- The `Execute(...)` -method should
 - o Download data from this URL
 - <https://cdn.relewise.com/academy/productdata/raw>
 - o Parse the raw data and map each row into an instance of `Relewise.Client.DataTypes.Product` (available on NuGet: *Install-Package Relewise.Client*)
 - The minimum properties that you should map are:
 - `ProductId` should map to `string Id`
 - `ProductName` should map to `Multilingual DisplayName`
 - `List Price` should map to `MultiCurrency ListPrice`
 - `Sales Price` should map to `MultiCurrency SalesPrice`
 - o The method, which returns a string, should return a message containing a count of how many products were mapped.

Exercise 2: Download Json Product Data and Map to Relewise Products

- Write a C# Class, that implements the `IJob` -interface (see *Appendix-section*)
- The `Execute(...)` -method should
 - o Download data from this URL
 - <https://cdn.relewise.com/academy/productdata/customjsonfeed>
 - o Deserialize the JSON data into an array of a local Class type, and map each instance of this local type into an instance of `Relewise.Client.DataTypes.Product` (available on NuGet: *Install-Package Relewise.Client*)
 - The minimum properties that you should map are:
 - `productId` should map to `string Id`
 - `productName` should map to `Multilingual DisplayName`
 - `listPrice` should map to `MultiCurrency ListPrice`
 - `salesPrice` should map to `MultiCurrency SalesPrice`
 - o The method, which returns a string, should return a message containing a count of how many products were mapped.

Exercise 3: Download Google Shopping Feed and Map to Relewise Products

- Write a C# Class, that implements the `IJob` -interface (see *Appendix-section*)
- The `Execute(...)` -method should
 - o Download data from this URL
 - <https://cdn.relewise.com/academy/productdata/googleshoppingfeed>
 - o The data represents a Google Shopping Feed (<https://support.google.com/merchants/answer/7052112?hl=en>), where you need to map each `<item>`-element into an instance of `Relewise.Client.DataTypes.Product` (available on NuGet: *Install-Package Relewise.Client*)
 - The minimum properties that you should map are:
 - `<g:id>` should map to `string Id`
 - `<title>` should map to `Multilingual DisplayName`
 - `<g:price>` should map to `MultiCurrency ListPrice`
 - `<g:sale_price>` should map to `MultiCurrency SalesPrice`
 - o The method, which returns a string, should return a message containing a count of how many products were mapped.

Appendix: Various C# types needed for the assignments

```
public interface IJob
{
    Task<string> Execute(
        JobArguments arguments,
        Func<string, Task> info,
        Func<string, Task> warn,
        CancellationToken token);
}

public class JobArguments
{
    public JobArguments(
        Guid datasetId,
        string apiKey,
        IReadOnlyDictionary<string, string> jobConfiguration)
    {
        DatasetId = datasetId;
        ApiKey = apiKey;
        JobConfiguration = jobConfiguration;
    }

    public Guid DatasetId { get; }
    public string ApiKey { get; }

    public IReadOnlyDictionary<string, string> JobConfiguration { get; }
}
```