**When To Use FirstOrDefault with Linq Expression**

[FirstOrDefault](https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.firstordefault) is a Linq expression that returns the first element of a sequence or a default value (which is usually null) when no element is found. [First](https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.first) is similar, but throws an error when no element is found. FirstOrDefault is usually preferred to First because it does not throw an error unless you are sure that First is the better option.

FirstOrDefault can be used instead of [Where](https://docs.microsoft.com/en-us/dotnet/api/system.linq.enumerable.where) when the first element is always used. The difference is that FirstOrDefault returns an element while Where returns a list. Then, you need to get the first element in the list. For example, the condition is filtering the primary key and therefore it returns only one record. If that is the case, FirstOrDefault is better than Where because FirstOrDefault will return null when there is no record. It would be easier to do null check.

Let’s see the example that illustrates that FirstOrDefault is more elegant than Where when we need to do null check on the filtered list.

As you can see, using Where would be too awkward. Where returns a list with one element. To check if the list is empty or not, you have to check the count. To return a list, you need to pick the first element.

On the other hand, FirstOrDefault returns an element, not a list with one element. The refactored code is more elegant.

<script src="https://gist.github.com/mydatahack/0b620d4b567c9bb41309e09157c3237f.js"></script>

Pretty cool.