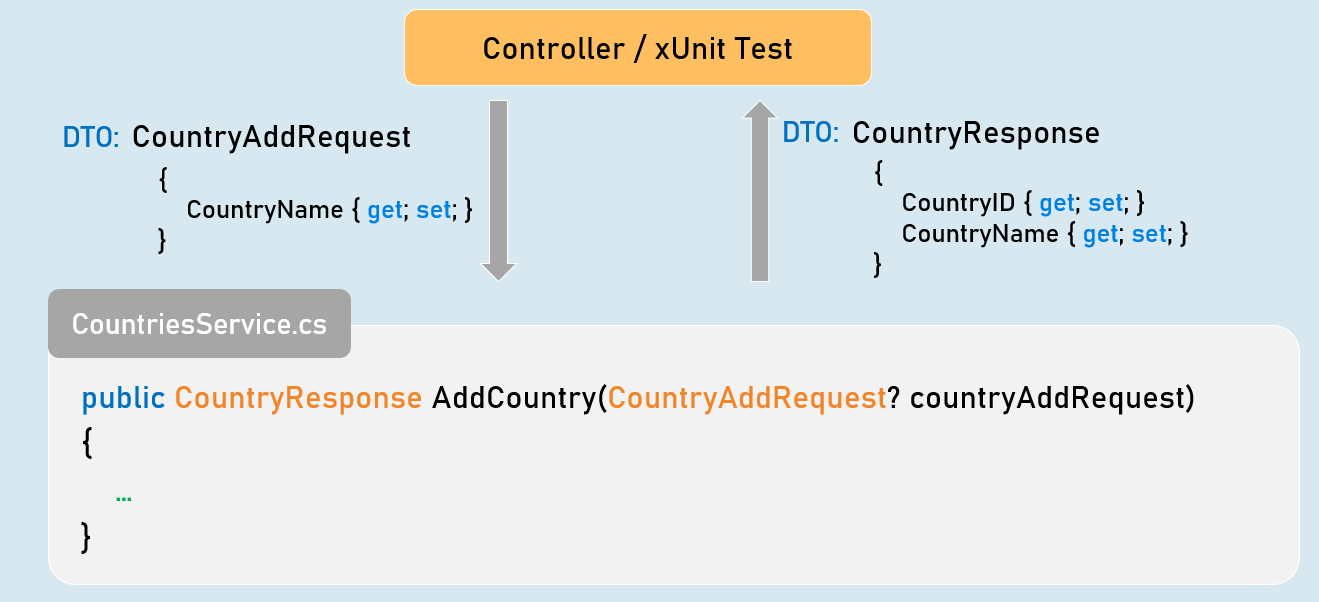
**Introduction to xUnit**

xUnit is the free, open source unit testing tool for .NET Framework.

* Easy and extensible.
* Best to use with a mocking framework called "Moq".

**Add Country - xUnit Test**



public CountryResponse AddCountry(CountryAddRequest? countryAddRequest)

{

//Check if "countryAddRequest" is not null.

//Validate all properties of "countryAddRequest"

//Convert "countryAddRequest" from "CountryAddRequest" type to "Country".

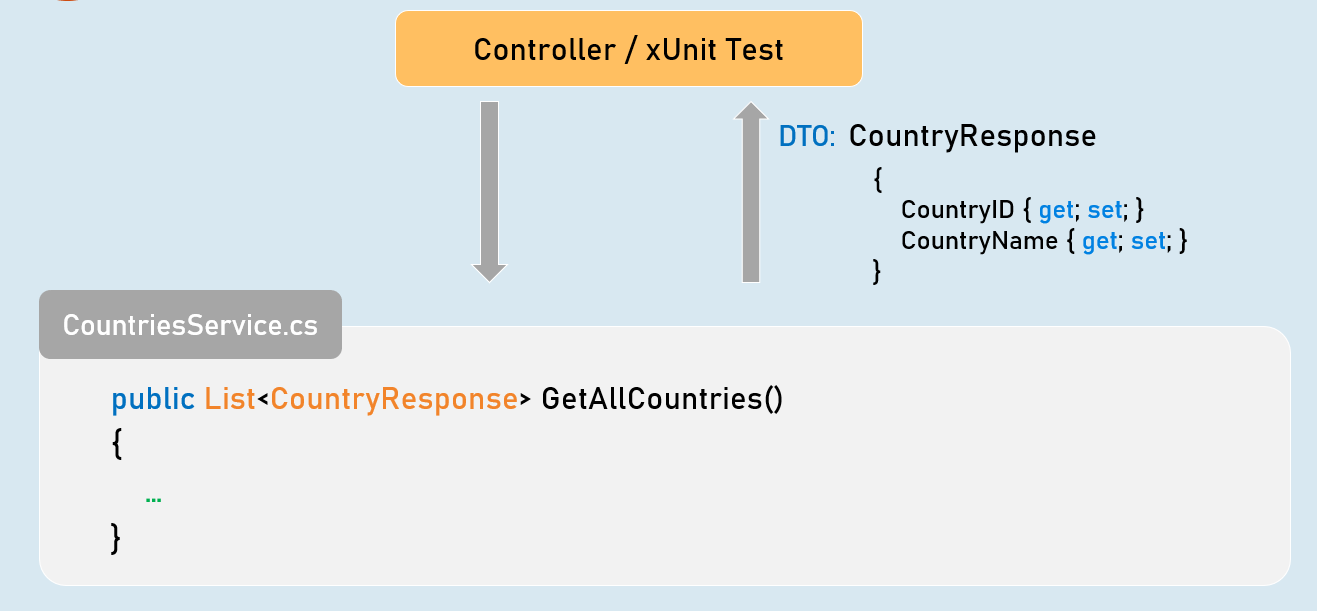
//Generate a new CountryID

//Then add it into List<Country>

//Return CountryResponse object with generated CountryID

}

**Get All Countries - xUnit Test**



public List<CountryResponse> GetAllCountries()

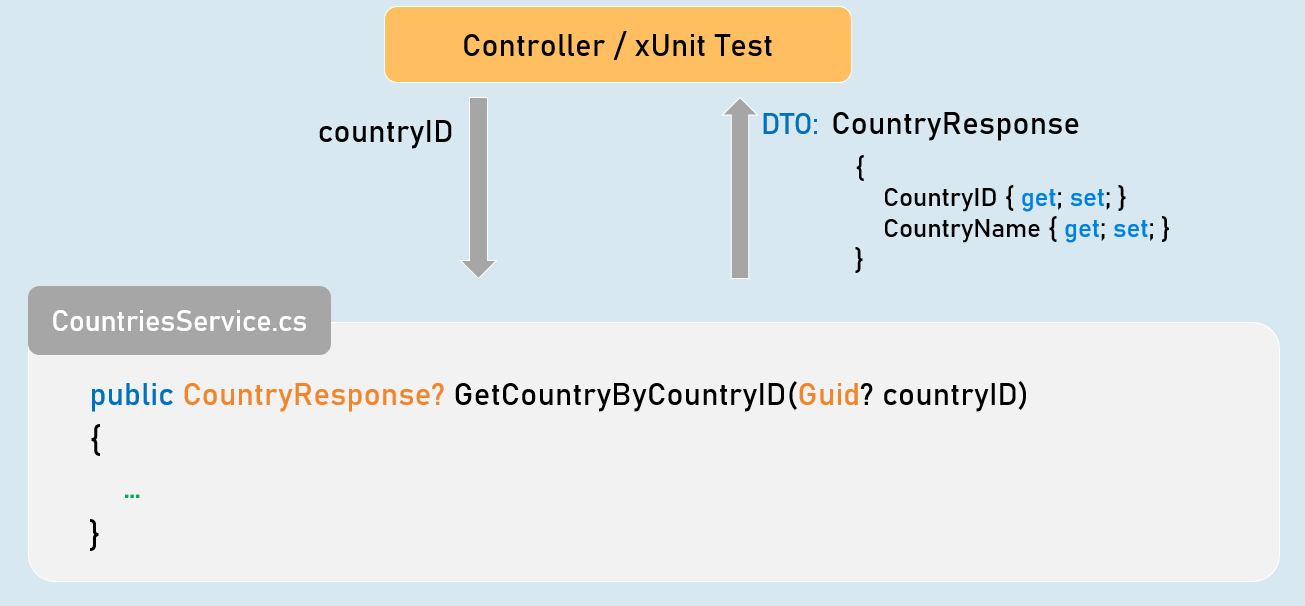
{

//Convert all countries from "Country" type to "CountryResponse" type.

//Return all CountryResponse objects

}

**Get Country by Country ID - xUnit Test**



public CountryResponse? GetCountryByCountryID(Guid? countryID)

{

//Check if "countryID" is not null.

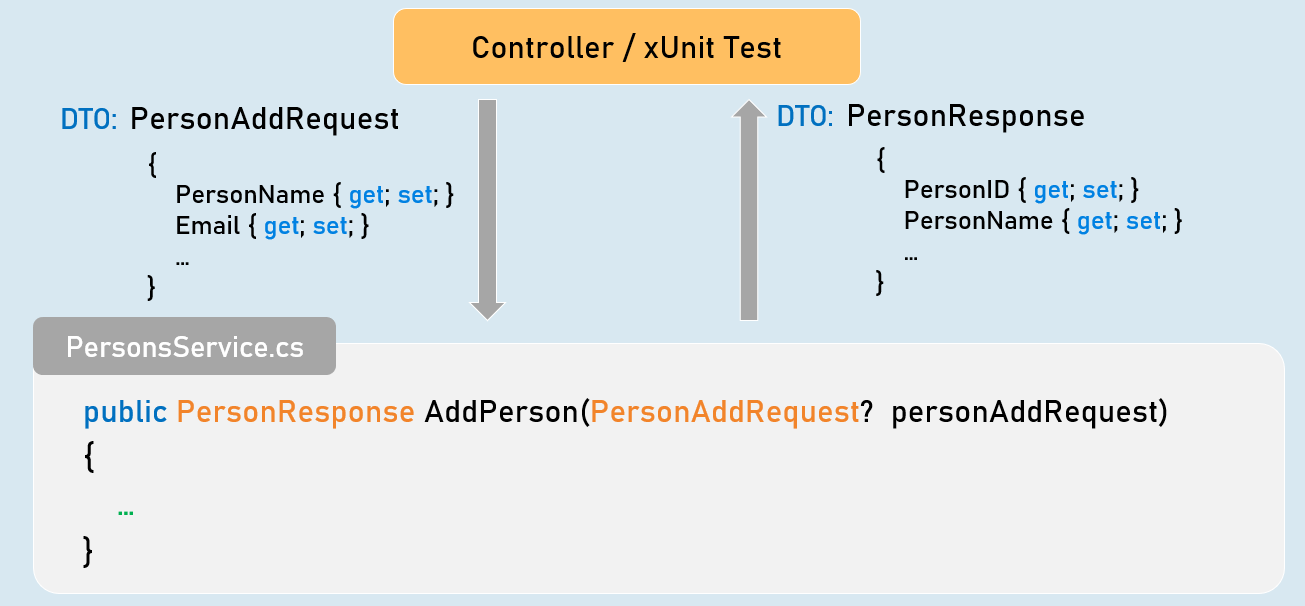
//Get matching country from List<Country> based countryID.

//Convert matching country object from "Country" to "CountryResponse" type.

//Return CountryResponse object

}

**Add Person - xUnit Test**



public PersonResponse AddPerson(PersonAddRequest? personAddRequest)

{

//Check if "personAddRequest" is not null.

//Validate all properties of "personAddRequest".

//Convert "personAddRequest" from "PersonAddRequest" type to "Person".

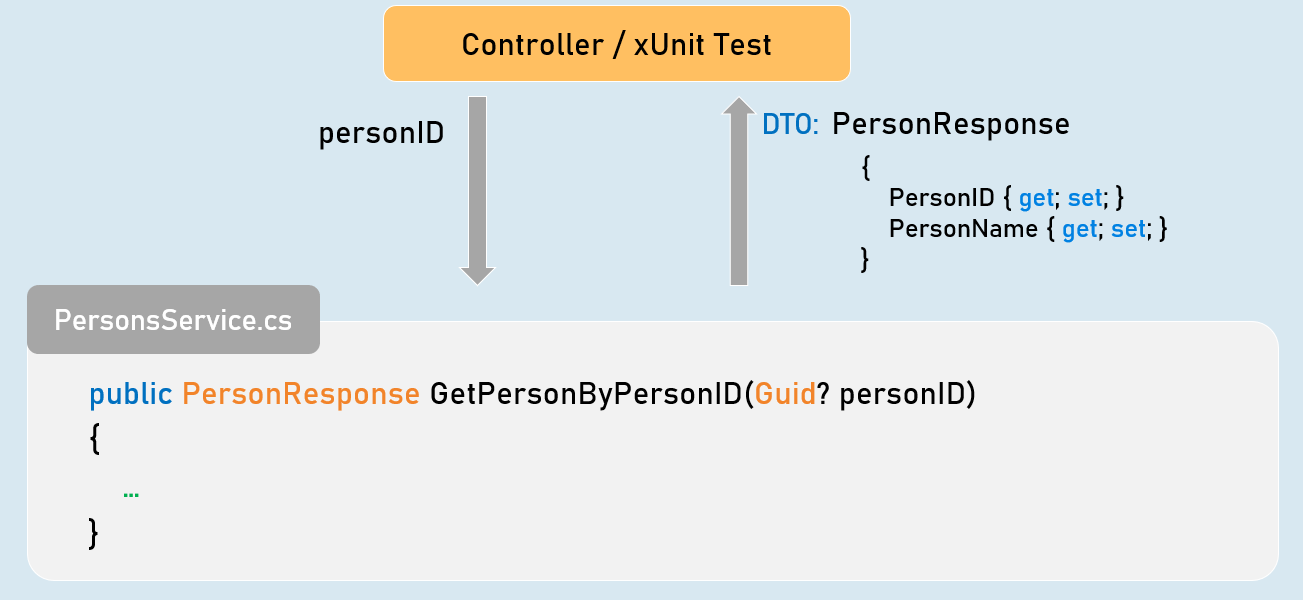
//Generate a new PersonID.

//Then add it into List<Person>.

//Return PersonResponse object with generated PersonID.

}

**Get Person by Person ID - xUnit Test**



public PersonResponse GetPersonByPersonID(Guid? personID)

{

//Check if "personID" is not null.

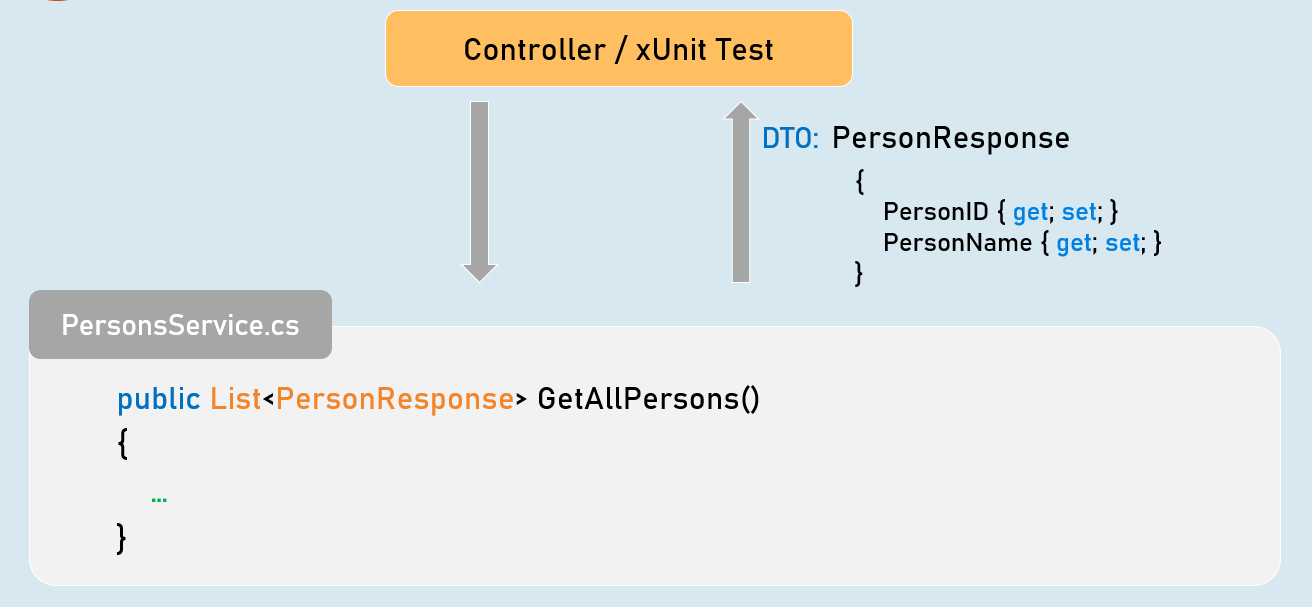
//Get matching person from List<Person> based personID.

//Convert matching person object from "Person" to "PersonResponse" type.

//Return PersonResponse object

}

**Get All Persons - xUnit Test**



public List<PersonResponse> GetAllPersons()

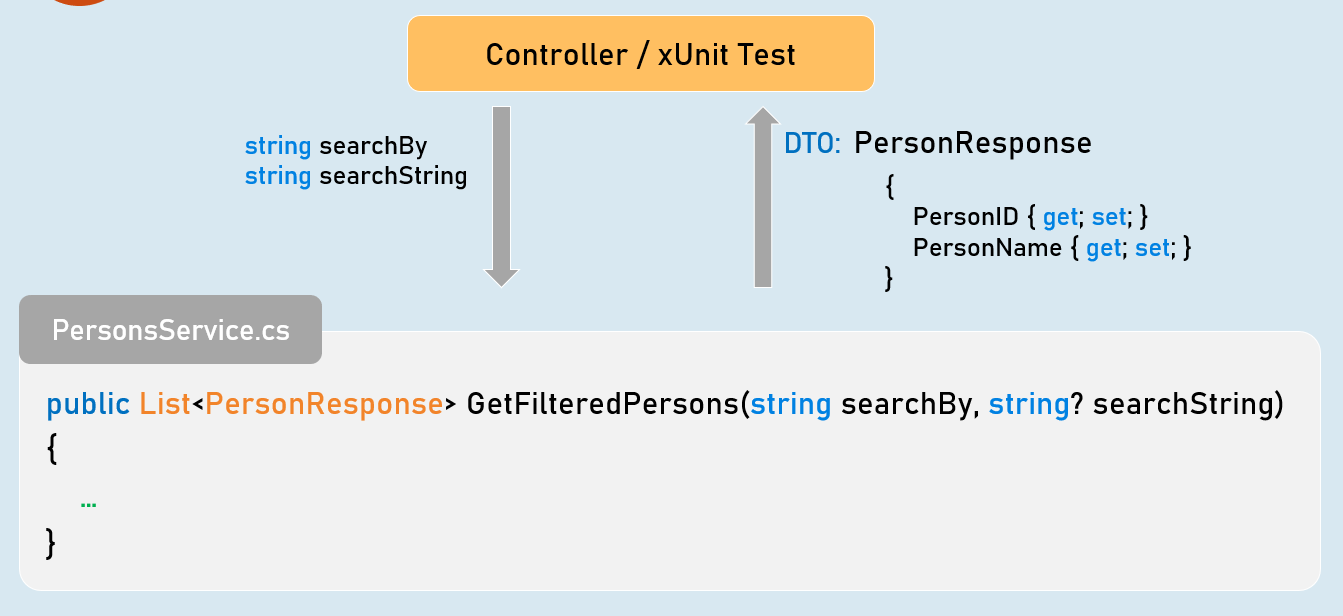
{

//Convert all persons from "Person" type to "PersonResponse" type.

//Return all PersonResponse objects

}

**Get Filtered Persons - xUnit Test**



public List<PersonResponse> GetFilteredPersons(string searchBy, string? searchString)

{

//Check if "searchBy" is not null.

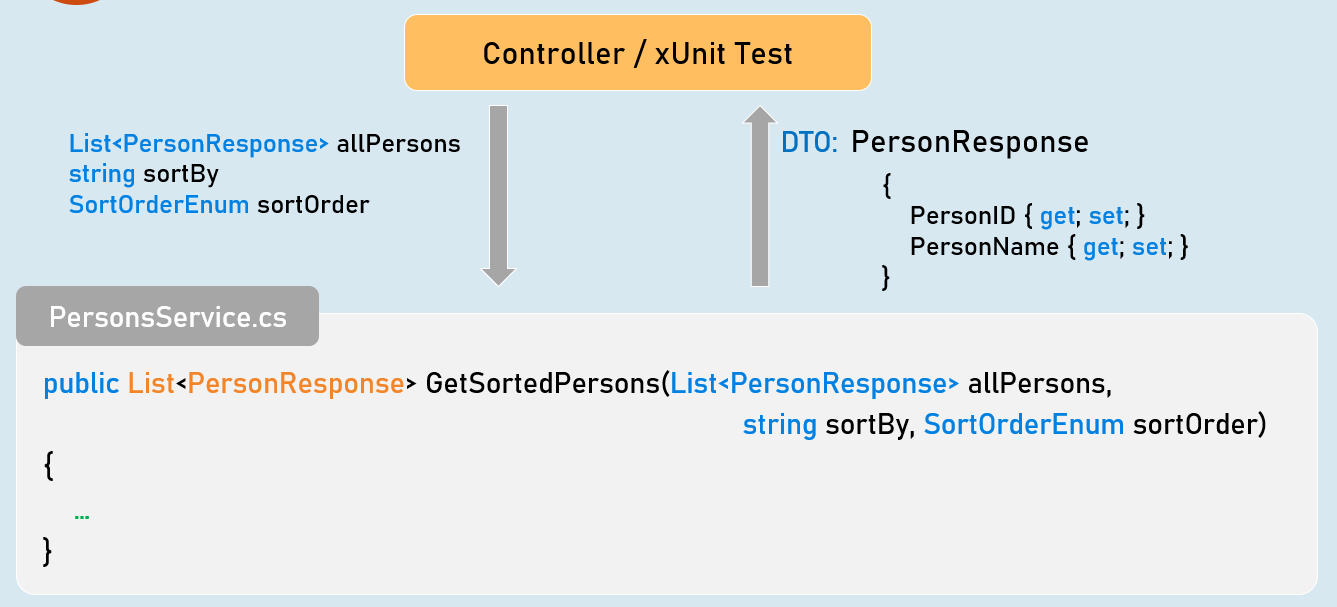
//Get matching persons from List<Person> based on given searchBy and searchString.

//Convert the matching persons from "Person" type to "PersonResponse" type.

//Return all matching PersonResponse objects

}

**Get Sorted Persons - xUnit Test**



public List<PersonResponse> GetSortedPersons(List<PersonResponse> allPersons,

string sortBy, SortOrderEnum sortOrder)

{

//Check if "sortBy" is not null.

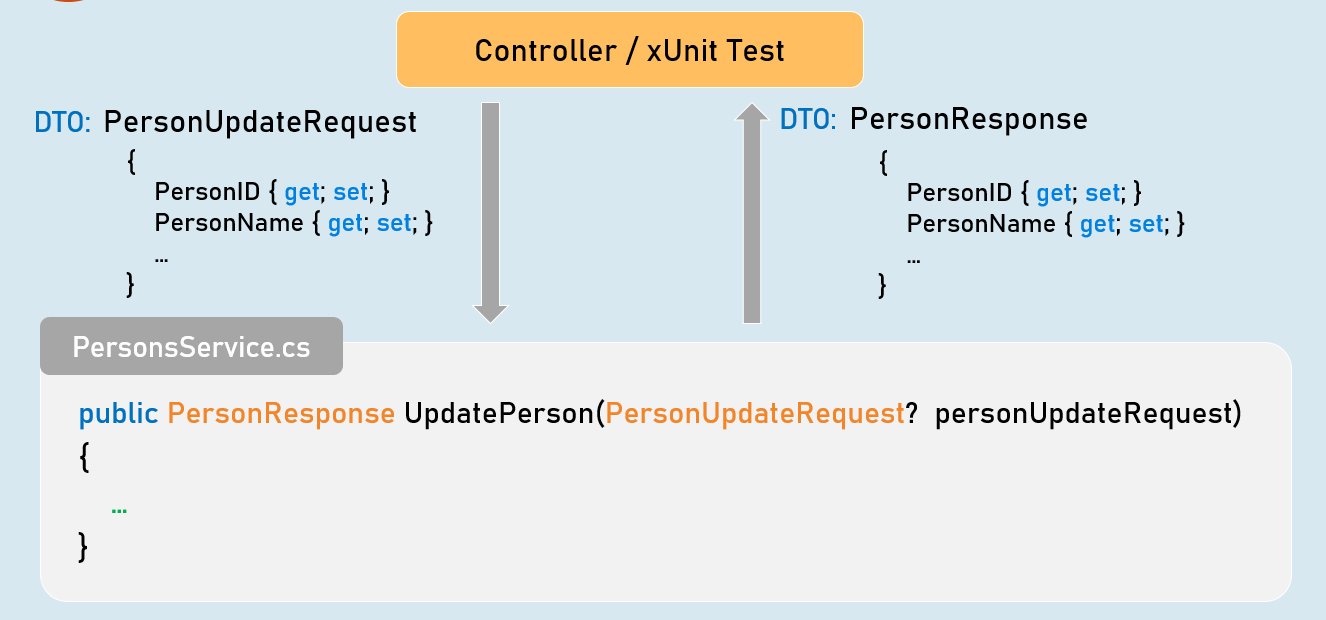
//Get sorted persons from "allPersons" based on given "sortBy" and "sortOrder".

//Convert the sorted persons from "Person" type to "PersonResponse" type.

//Return all sorted PersonResponse objects

}

**Update Person - xUnit Test**



public PersonResponse UpdatePerson(PersonUpdateRequest? personUpdateRequest)

{

//Check if "personUpdateRequest" is not null.

//Validate all properties of "personUpdateRequest"

//Get the matching "Person" object from List<Person> based on PersonID.

//Check if matching "Person" object is not null

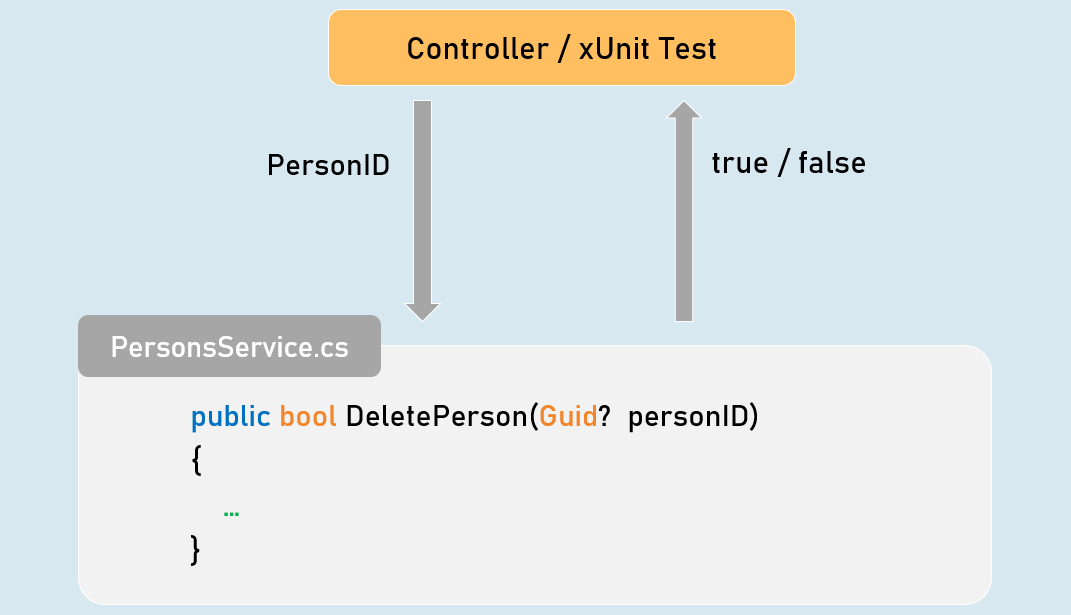
//Update all details from "PersonUpdateRequest" object to "Person" object

//Convert the person object from "Person" to "PersonResponse" type

//Return PersonResponse object with updated details

}

**Delete Person - xUnit Test**



public bool DeletePerson(Guid? personID)

{

//Check if "personID" is not null.

//Get the matching "Person" object from List<Person> based on PersonID.

//Check if matching "Person" object is not null

//Delete the matching "Person" object from List<Person>

//Return Boolean value indicating whether person object was deleted or not

}