

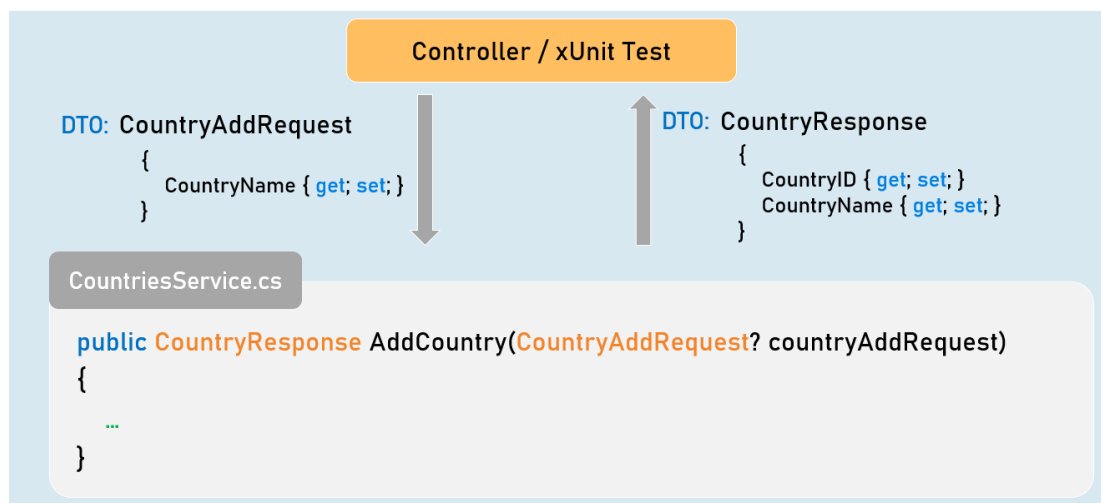
Section Cheat Sheet (PPT)

Introduction to xUnit

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

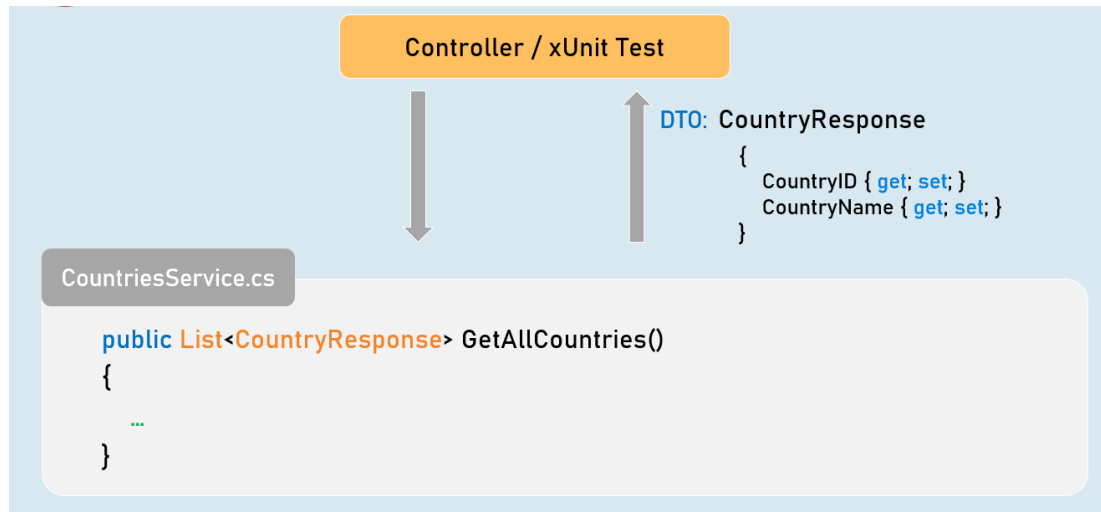
- Easy and extensible.
- Best to use with a mocking framework called "Moq".

Add Country - xUnit Test



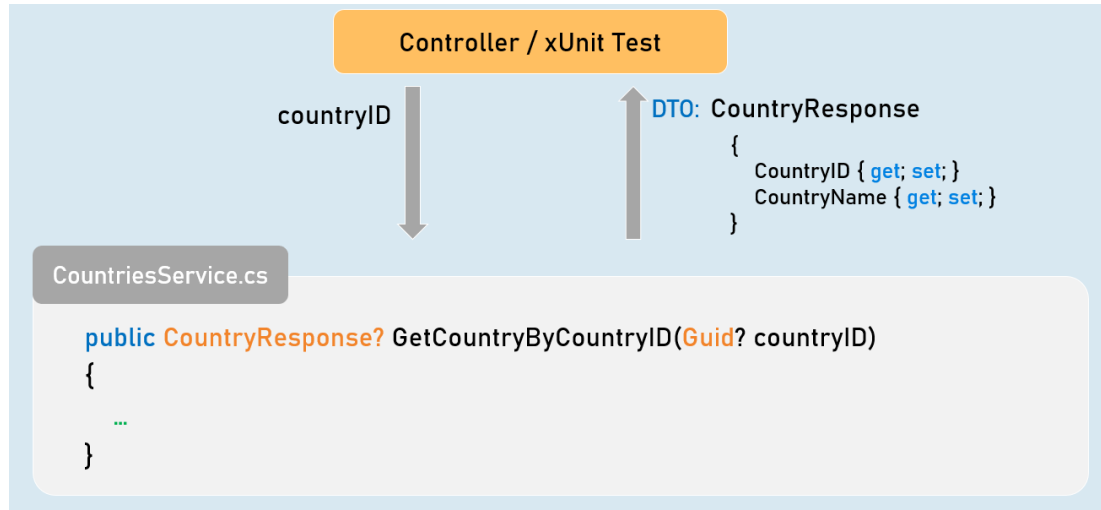
```
1 | public CountryResponse AddCountry(CountryAddRequest? countryAddRequest)
2 | {
3 |     //Check if "countryAddRequest" is not null.
4 |     //Validate all properties of "countryAddRequest"
5 |     //Convert "countryAddRequest" from "CountryAddRequest" type to
   | "Country".
6 |     //Generate a new CountryID
7 |     //Then add it into List<Country>
8 |     //Return CountryResponse object with generated CountryID
9 | }
```

Get All Countries - xUnit Test



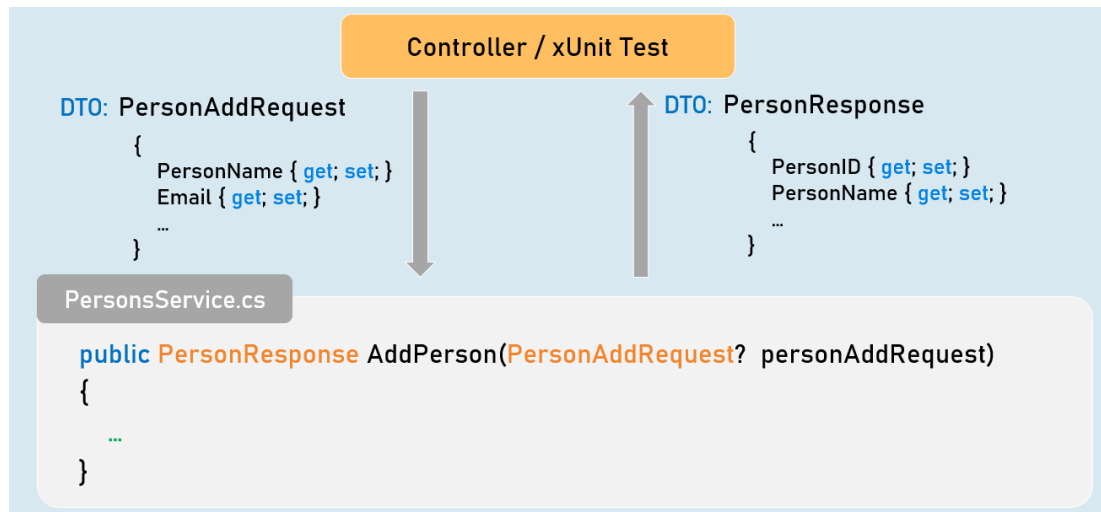
```
1 | public List<CountryResponse> GetAllCountries()
2 | {
3 |     //Convert all countries from "Country" type to "CountryResponse" type.
4 |     //Return all CountryResponse objects
5 | }
```

Get Country by Country ID - xUnit Test



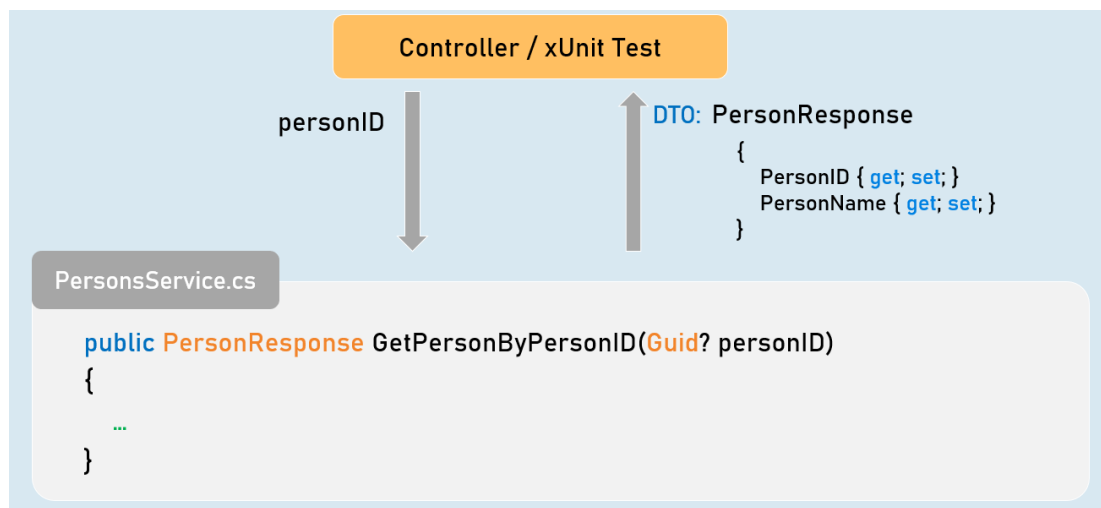
```
1 | public CountryResponse? GetCountryByCountryID(Guid? countryID)
2 | {
3 |     //Check if "countryID" is not null.
4 |     //Get matching country from List<Country> based countryID.
5 |     //Convert matching country object from "Country" to "CountryResponse"
   | type.
6 |     //Return CountryResponse object
7 | }
```

Add Person - xUnit Test



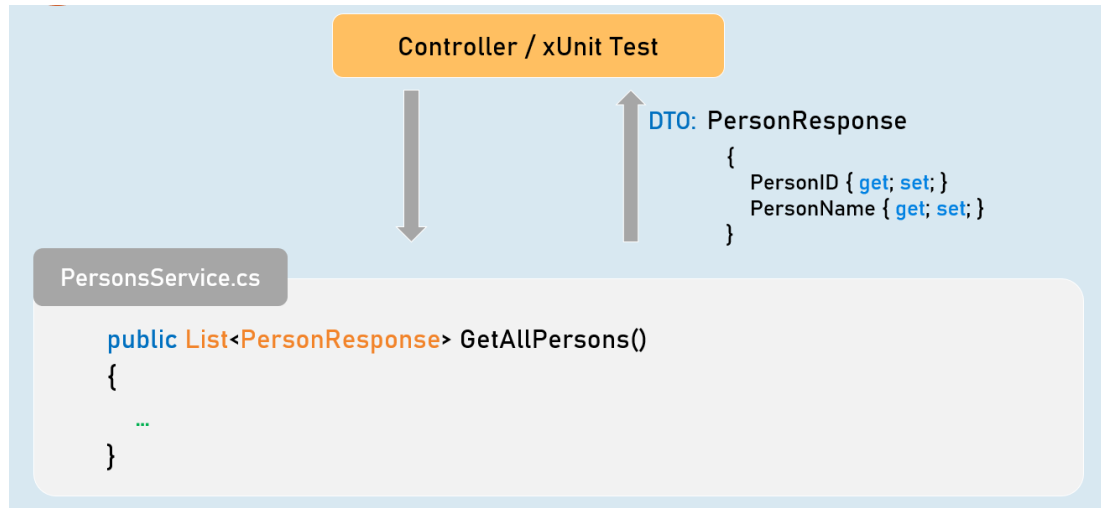
```
1 | public PersonResponse AddPerson(PersonAddRequest? personAddRequest)
2 | {
3 |     //Check if "personAddRequest" is not null.
4 |     //Validate all properties of "personAddRequest".
5 |     //Convert "personAddRequest" from "PersonAddRequest" type to "Person".
6 |     //Generate a new PersonID.
7 |     //Then add it into List<Person>.
8 |     //Return PersonResponse object with generated PersonID.
9 | }
```

Get Person by Person ID - xUnit Test



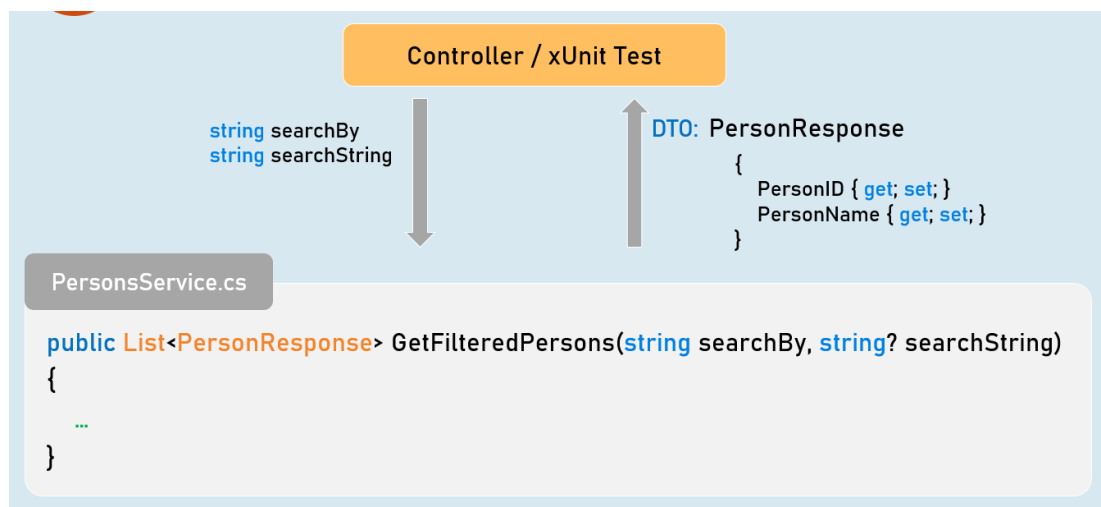
```
1 | public PersonResponse GetPersonByPersonID(Guid? personID)
2 | {
3 |     //Check if "personID" is not null.
4 |     //Get matching person from List<Person> based personID.
5 |     //Convert matching person object from "Person" to "PersonResponse" type.
6 |     //Return PersonResponse object
```

Get All Persons - xUnit Test



```
1 | public List<PersonResponse> GetAllPersons()
2 | {
3 |     //Convert all persons from "Person" type to "PersonResponse" type.
4 |     //Return all PersonResponse objects
5 | }
```

Get Filtered Persons - xUnit Test



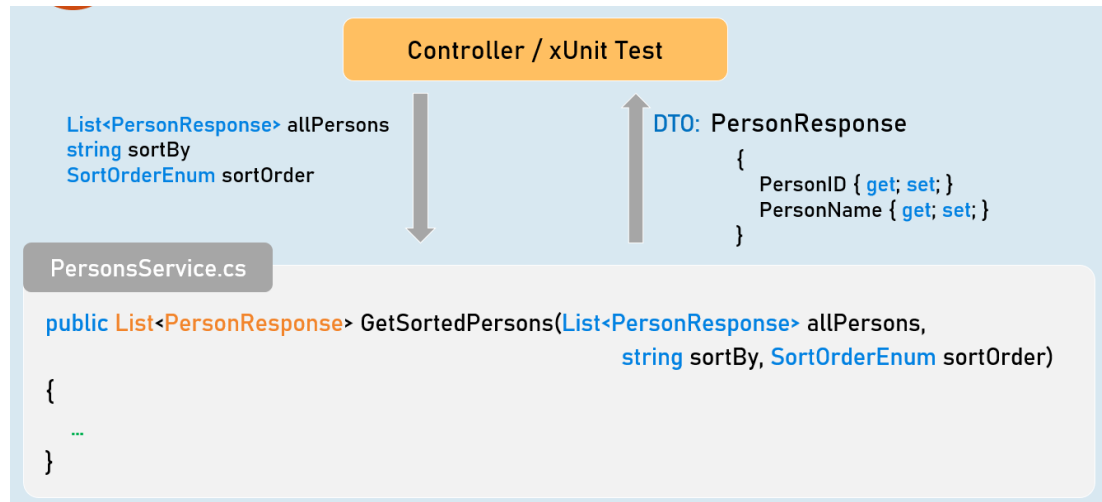
```
1 | public List<PersonResponse> GetFilteredPersons(string searchBy, string?
   | searchString)
2 | {
3 |     //Check if "searchBy" is not null.
4 |     //Get matching persons from List<Person> based on given searchBy and
   | searchString.
```

```

5 | //Convert the matching persons from "Person" type to "PersonResponse"
   | type.
6 | //Return all matching PersonResponse objects
7 | }

```

Get Sorted Persons - xUnit Test

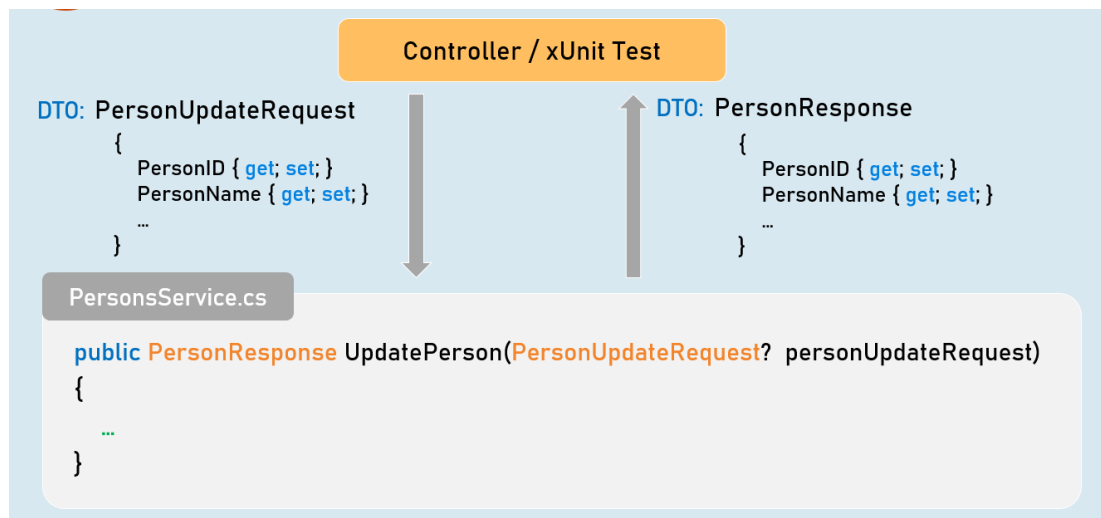


```

1 | public List<PersonResponse> GetSortedPersons(List<PersonResponse>
   | allPersons,
2 | string sortBy, SortOrderEnum sortOrder)
   | {
3 |     //Check if "sortBy" is not null.
4 |     //Get sorted persons from "allPersons" based on given "sortBy" and
   | "sortOrder".
5 |     //Convert the sorted persons from "Person" type to "PersonResponse"
   | type.
6 |     //Return all sorted PersonResponse objects
7 | }
8 |

```

Update Person - xUnit Test

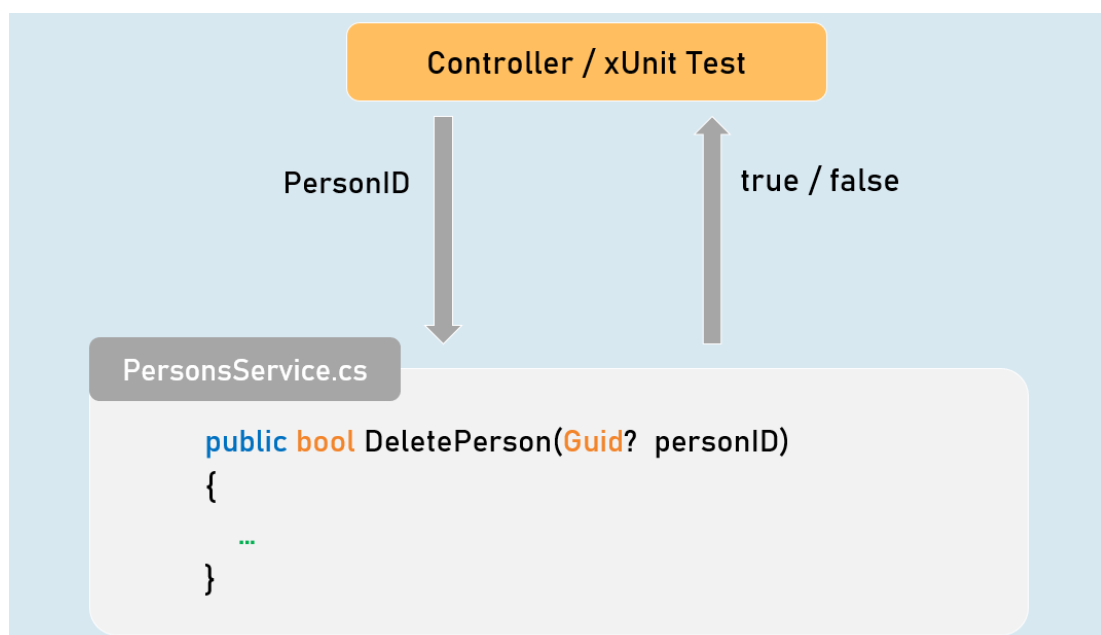


```

1 | public PersonResponse UpdatePerson(PersonUpdateRequest?
   | personUpdateRequest)
2 | {
3 |     //Check if "personUpdateRequest" is not null.
4 |     //Validate all properties of "personUpdateRequest"
5 |     //Get the matching "Person" object from List<Person> based on PersonID.
6 |     //Check if matching "Person" object is not null
7 |     //Update all details from "PersonUpdateRequest" object to "Person"
   | object
8 |     //Convert the person object from "Person" to "PersonResponse" type
9 |     //Return PersonResponse object with updated details
10| }

```

Delete Person - xUnit Test



```

1 | public bool DeletePerson(Guid? personID)
2 | {

```

```
3 | //Check if "personID" is not null.
4 | //Get the matching "Person" object from List<Person> based on PersonID.
5 | //Check if matching "Person" object is not null
6 | //Delete the matching "Person" object from List<Person>
7 | //Return Boolean value indicating whether person object was deleted or
   | not
8 | }
```