

Creating Interfaces to Add Extensibility



Jeremy Clark

DEVELOPER BETTERER

@jeremybytes www.jeremybytes.com



How



Create a repository interface

Implement the interface

- Web service

- Text file

- SQL database

Remove code duplication

Focus on important functionality



Different Data Sources



Web service



Text file



SQL database



Document database



Cloud service



Azure functions

Repository Pattern

Mediates between the domain and data mapping layers using a collection-like interface for accessing domain objects.

Fowler, et al. *Patterns of Enterprise Application Architecture*. Addison-Wesley, 2003.

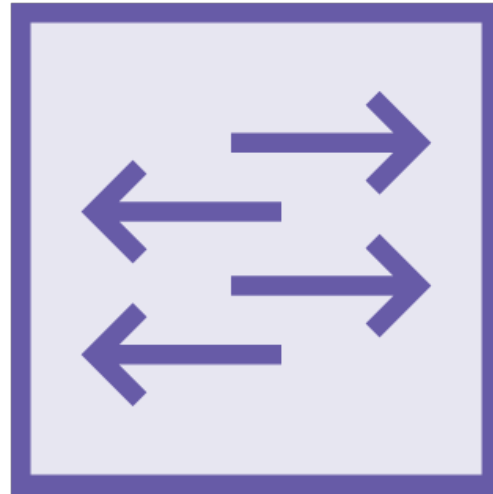


Repository Pattern

Separates our application from the data storage technology



Application



Repository



Data store

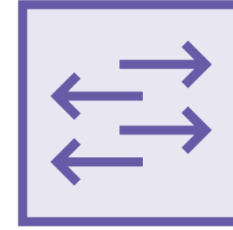
Repository Pattern



Application



Interface



Service repository



CSV repository



SQL repository



Web service



Text file



SQL database



CRUD Repository

Create

Read

Update

Delete



```
public interface IPersonRepository
{
    void AddPerson(Person newPerson);
    IEnumerable<Person> GetPeople();
    Person GetPerson(int id);
    void UpdatePerson(int id,
        Person updatedPerson);
    void DeletePerson(int id);
}
```

◀ Create

◀ Read

◀ Read

◀ Update

◀ Delete



Demo



Implementing an interface

Multiple repositories

- Service
- Text file
- SQL database

Remove duplication



Repository Factory

```
IPersonRepository GetRepository(string repositoryType) {  
    IPersonRepository repository = null;  
  
    switch (repositoryType) {  
        case "Service": repository = new ServiceRepository();  
            break;  
        case "CSV": repository = new CSVRepository();  
            break;  
        case "SQL": repository = new SQLRepository();  
            break;  
    }  
    return repository;  
}
```



Demo



Add factory method

Remove references to specific
repositories

Application only knows the interface



No References to Specific Repositories

```
private void PopulateListBox(string repositoryType)
{
    ClearListBox();

    IPersonRepository repository =
        RepositoryFactory.GetRepository(repositoryType);

    var people = repository.GetPeople();

    foreach (var person in people)
        PersonListBox.Items.Add(person);

    ShowRepositoryType(repository);
}
```



How



Create a repository interface

Implement

- Web service

- Text file

- SQL database

Remove code duplication

Focus on important functionality

