

Creating a Real-world C# Application



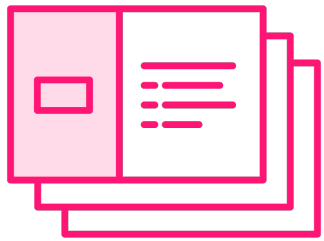
Filip Ekberg

Principal Consultant & CEO

@fekberg | fekberg.com



Creating a Layered Architecture



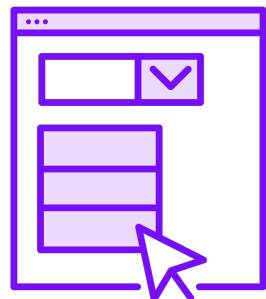
Domain

Models representing the data and types used in the domain. Expressed in a similar language as the business would use.



Infrastructure

Interaction with external systems such as databases, services and the file system.



UI

The definition of the UI and its components. This may be divided into multiple different projects as well if the UI components are shared among multiple applications.



Records are great for data transmission objects and domain models.

They will NOT work with EF Core.



Connecting the Model and Business Logic

```
record Bug (..) : TodoTask(..);

class BugRepository
{
    public void Save(Bug bug) { /* Database code to save the bug goes here */ }
}

class BugViewModel
{
    BugRepository repository;

    public string Title { get; set; } // UI automatically populates this property with a value!

    public void SaveCommand()
    {
        // Construct a domain model instance based on data from the UI
        // Without knowing about the UI layer!
        Bug bugToSave = new (Title, ..);

        repository.Save(bugToSave);
    }
}
```



Using the Repository

```
class BugRepository : IRepository<Bug>
{
    public Task<Bug> GetAsync(Guid id)
    { /* return a bug from the database */ }
}
```

```
IRepository<Bug> repository = ...;
```

```
Bug result = await repository.GetAsync(...);
```



Using the TodoInMemoryRepository

```
class BugViewModel
{
    IRepository<Bug> repository;

    public BugViewModel(IRepository<Bug> repository)
    {
        this.repository = repository;
    }

    public void SaveCommand()
    {
        repository.Save(...);
    }
}

var viewModel = new BugViewModel(new TodoInMemoryRepository<Bug>());
```



Writing test code as you introduce models and business logic is a good way to approach this task!



ICommand and RelayCommand

“A command whose sole purpose is to relay its functionality to other objects by invoking delegates. “

Example:

```
ICommand saveCommand = new RelayCommand(() => {  
    repository.Save(...);  
});
```

UI binds to saveCommand to trigger the action.

