Article topic: How to implement and refactor a service?

What is the current status?

 a.Service is in the core layer and is not configuarable.

 b.Service is quite limited.

 c.For layers above the core, they have to public some properties for accessing services.

 d.The Cache service is at the composer level.

What is the benifts and short comings?

What is the chaning process and final result?

  a. Add one service layer.

  b. Moving service from Core to service layer.

  c. Moving the cache service to service layer

  d. Define some prodefined services and other contract.

  e. Implement some default ones.

  f. Implement one service provider which is configurable.

  g. Provide ways to define services.

What is the remaining issues?

  a. Robustness

  b. Expandibility.

1. Refactor the architect, add one service layer.

2. Setup the infrastructure of redo/undo

3. Add limited support for redo/undo.

1.  Refactor the code.

   a. Remove cache service interface from Core project and IDataSource.

   b. Add one service layer, which defines the IServices and IServiceProvider. This layer is a new layer between core and other layers.

   c. Implement one default ServiceManager which derive from IServiceProvider, by which external apps can register, unregister and query services.

   d. Defines interfaces for CacheService, UndoRedoService and CopyPasteService, and also one default implementation of the services.

   e. Add one provider for services and config it to Vdc Composer.

2.In shared project.

   a. Implement concrete copy/paste items (which are converted from original Actions.cs).

   b. Impelement some concrete undo/redo items.

   c. Integrate the items with the UIs, to support copy/paste, redo/undo.

3. In Datasource project, change the way of getting services.

4. In composer

   a. parse the service provider and setup the services.

   b. integrate redo/undo.

   e. Add redo/undo resources.

5. In unit test, change way of getting services.

Article abstract:

    A practice of refactoring: seperate mixed services to a configurable service layer

Content:

    Refactoring is important in software development. This article is based on some refactoring work I've done to the VDC project. It walks thourgh the refactoring work, including the situation before the refactoring, the reason for refactoring, how & why I choose the refactor way, the final choice with its benifits and shortcomings, from my personal point of view. The result is far from perfect, but it is better than before. Hope my experience can give some value to our daily development.