Interface_and_Data_Provider_ Updates_to_support_Article_Editing

- Updating the IArticleDataProvider interface to allow editing of articles
- Updating the FakeArticleDataProvider to match the updated interface
- Updating the ArticleManager to use the new interface methods

Updating IArticleDataProvider.cs

```
using System.Collections.Generic;

namespace MyBlog.src
{/// <summary>
///An interface to an article provider.
/// </summary>
    interface IArticleDataProvider
    {   /// <summary>
        /// Initializes this provider.
        /// </summary>
        void Initialize();

        /// <summary>
        /// Gets an article by the provided identifier.
        /// </summary>
        /// <param name="id">The identifier to get the article by.</param>
        /// <returns></returns>
        ArticleInfo GetArticleById(int id, bool getNext);
        /// <summary>
        /// Gets the latest article
        /// </summary>
        /// <returns></returns>
        ArticleInfo GetLatestArticle();
/// <summary>
        /// Gets the article ids and titles.
        /// </summary>
        /// <returns></returns>
        IEnumerable<ArticleInfo> GetArticleIdsAndTitles();

        /// <summary>
        /// Updates the article
        /// </summary>
        /// <param name="id">The identifier</param>
        /// <param name="title">The title</param>
        /// <param name="content">The content</param>
        /// <returns>True if successful; otherwise false.</returns>
        bool UpdateArticle(int id, string title, string content);

        /// <summary>
        /// Creates a new article.
        /// </summary>
        /// <param name="title">The title.</param>
        /// <param name="content">The content.</param>
        /// <returns>The id of the newly created article.</returns>
        int CreateArticle(string title, string content);
    }
}
```

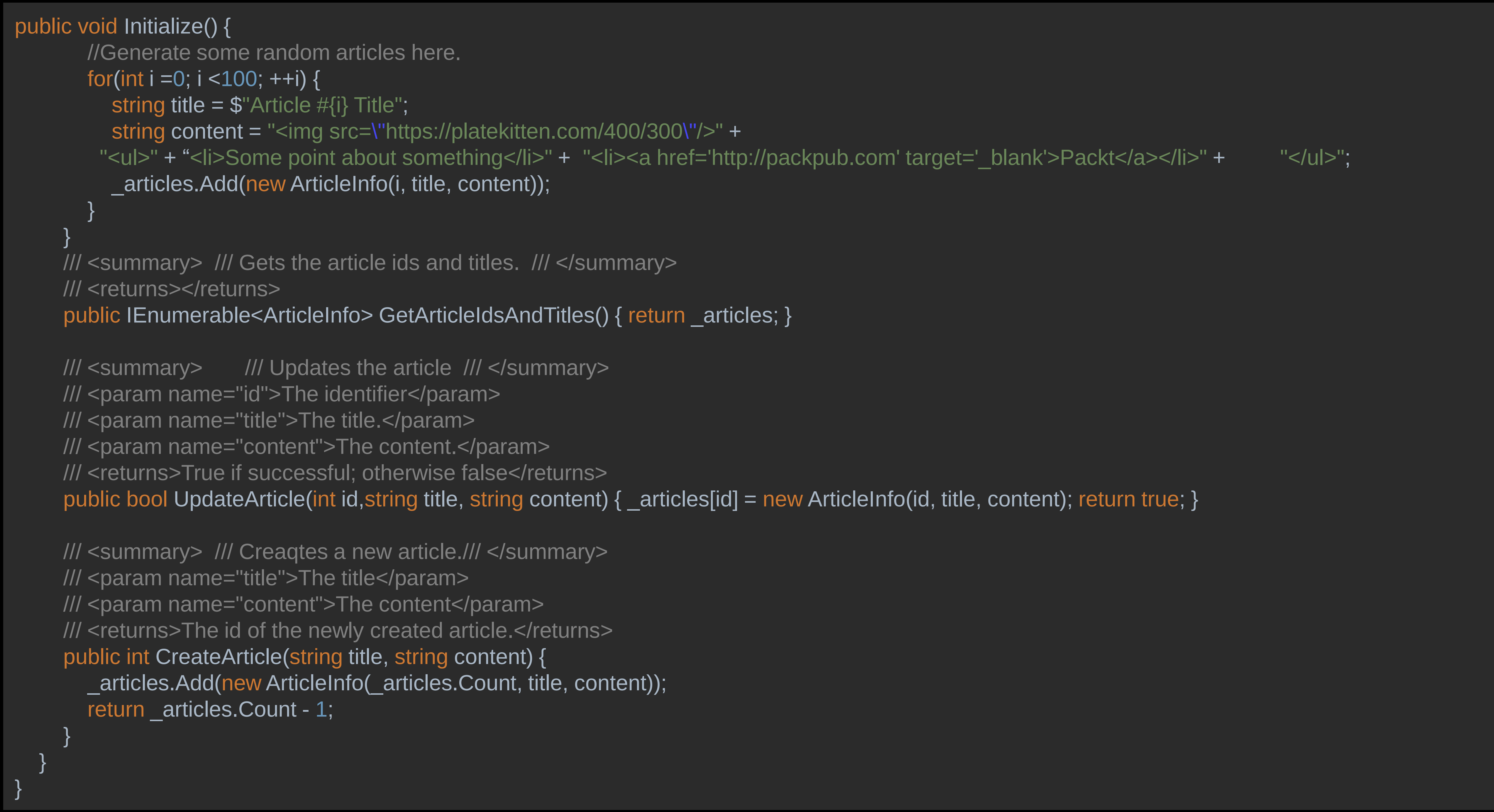
Updating FakeArticleDataProvider.cs

```
namespace MyBlog.src {
    /// <summary>
    /// Provides fake, hard-coded data for use with the blog.
    /// </summary>
    public class FakeArticleDataProvider : IArticleDataProvider {

        private List<ArticleInfo> _articles = new List<ArticleInfo>();

        public ArticleInfo GetArticleById(int id , bool getNext = true) {
            //Note: In a DB, we would want to check for the highest article id *under* this id.
            int actualId;
            if(getNext) { actualId = (id - 1) % _articles.Count(); }
            else { actualId = id; }
            return _articles[actualId];
        }

        public ArticleInfo GetLatestArticle() {
            return _articles.Last();
        }
```



```
public void Initialize() {
            //Generate some random articles here.
            for(int i =0; i <100; ++i) {
                string title = $"Article #{i} Title";
                string content = "<img src=\"https://platekitten.com/400/300\"/>" +
              "<ul>" + “<li>Some point about something</li>" +  "<li><a href='http://packpub.com' target='_blank'>Packt</a></li>" +          "</ul>";
                _articles.Add(new ArticleInfo(i, title, content));
            }
        }
        /// <summary>  /// Gets the article ids and titles.  /// </summary>
        /// <returns></returns>
        public IEnumerable<ArticleInfo> GetArticleIdsAndTitles() { return _articles; }

        /// <summary>       /// Updates the article  /// </summary>
        /// <param name="id">The identifier</param>
        /// <param name="title">The title.</param>
        /// <param name="content">The content.</param>
        /// <returns>True if successful; otherwise false</returns>
        public bool UpdateArticle(int id,string title, string content) { _articles[id] = new ArticleInfo(id, title, content); return true; }

        /// <summary>  /// Creaqtes a new article./// </summary>
        /// <param name="title">The title</param>
        /// <param name="content">The content</param>
        /// <returns>The id of the newly created article.</returns>
        public int CreateArticle(string title, string content) {
            _articles.Add(new ArticleInfo(_articles.Count, title, content));
            return _articles.Count - 1;
        }
    }
}
```

Updating ArticleManager.cs

```
namespace MyBlog.src
{
    /// <summary>  /// Manages articles in the system  /// </summary>
    public class ArticleManager
    {
        private static IArticleDataProvider _provider;

        public static void Initialize() {  //TODO: Hook up the data provider.
            _provider = new FakeArticleDataProvider();
            //Initialize the data provider.
            _provider.Initialize();
        }
        /// <summary> /// Gets the article with the provided identifier. If is is not found, the one with the next higest identifier is returned
        /// </summary>
        /// <param name="id">The article identifier.</param>
        /// <returns></returns>
        public static ArticleInfo GetArticleById(int id, bool getNext) {
            return _provider.GetArticleById(id,getNext);
        }

        /// <summary> /// Gets the latest article in the system /// </summary>
        /// <returns></returns>
        public static ArticleInfo GetLatestArticle() { return _provider.GetLatestArticle(); }

        public static IEnumerable<ArticleInfo> GetArticleIdsAndTitles() { return _provider.GetArticleIdsAndTitles(); }

        public static void UpdateArticle(ArticleInfo article) { _provider.UpdateArticle(article.Id, article.Title, article.Content); }

        public static int CreateArticle(string title, string content) { return _provider.CreateArticle(title, content); }
    }
}
```