A Design Pattern for Caching Frequently Used Data in XML

Paul D. Sheriff

BUSINESS SOLUTIONS ARCHITECT, FAIRWAY TECHNOLOGIES, INC.

www.fairwaytech.com   psheriff@fairwaytech.com

Module Goals

Read from SQL Server

- Cache data into XML file

Always read from XML file

- Unless data on server has changed

Detect changes

- Between server and XML file

Read and Store

Read From SQL Server

Use Entity Framework to get data

Serialize EF collection

Write serialized data to XML file

Demo

Get data from SQL Server, store locally

Read data locally

Detect Changes

Detect Changes

Must have a "last modified date" field on your table

Check for larger date on server

Check for different number of rows

Demo

Detect changes

Summary

Cache frequently used data in XML

Quicker to read from local file

Detect changes

- Use a date field
- Different # of rows

Course Summary

XML is a nice data transport format

- Easy to read and extend

Use LINQ to XML

- Use XPath if you have to

Serialize .NET objects as XML

Cache data locally for performance

Sample Code

The sample code for this course may be found at http://fairwaytech.com/pluralsight

I hope you enjoyed this course!

Paul D. Sheriff

Business Solutions Architect, Fairway Technologies, Inc.

www.fairwaytech.com **psheriff@fairwaytech.com**