Formulatrix

Software Engineering

Object-oriented Test

READ THESE FIRST:

- **Use English in all your answers.**
- **There is only 1 question in this test.**
- **Write your solution in C#**
- **Make sure your solution is compilable as a class library, NOT as an executable.**
- **You do not need to use any external references. But, you could if you would like to, but please indicate all sources of your references.**
- **You must be able to explain all your answers during the following interview. This also applies if you copy your answers from any external references.**
- **Use in-memory storage to demonstrate your solution**

The following requirements are optional. These serve as additional challenges.

- **This library will be used in a multi-threaded environment**
- **Design your solution in such a way that, in the future, it would take the least effort to:**
  - **Support more item content data type while remain strong-type (hint: do not use Object type).**
  - **Replace the current in-memory storage to any other options (i.e. database, file based, etc).**

A Formulatrix repository manager is a library that can be used to store and retrieve JSON string or XML string. A unique string is used to indicate the item being stored.

QUESTION:

- Implement a repository manager with the following specifications:

    - The following public API must be implemented. Please notice the signatures.

    - *// Store an item to the repository.*
      // Parameter itemType is used to differentiate JSON or XML.
      // 1 = itemContent is a JSON string.
      // 2 = itemContent is an XML string.
      void Register(**string** itemName, **string** itemContent, **int** itemType)

    - *// Retrieve an item from the repository.*
      string Retrieve(**string** itemName)

    - *// Retrieve the type of the item (JSON or XML).*
      int GetType(**string** itemName)

    - *// Remove an item from the repository.*
      void Deregister(**string** itemName)

    - *// Initialize the repository for use, if needed.*
      // You could leave it empty if you have your own way to make the repository ready for use
      // (e.g. using the constructor).
      void Initialize()

    - In **Register**, a validation must be performed against the **itemContent** based on the given **itemType**. You must include the validation process in your code, but you do not have to implement the logic itself.

    - A registered item should be protected from an overwrite.

    - If you use **Initialize**, it must be used just once after the repository instance is created.

- Write a **document** that explains the test scenarios that could be used to verify the above public API of this class. Or, alternatively, write the unit tests.

END OF DOCUMENT