

Dataset manipulation

A scriptable variable can access the datasets that are a part test to perform data manipulation. Datasets data can be accessed from the **SessionContext**. **GetDatasource(string)** method that returns a System.Data.DataTable object. This is useful in situations where it necessary to emulate javascript actions that the browser would otherwise perform. See the example below.

Example

The following example returns base64 encoded username and password that comes from a dataset. In this example we'll implement VU binding, so each VU use its own credentials from the dataset. There are a few steps performed here:

1. Get the dataset called *Credentials* which has columns *Username* and *Password*. Note: This dataset must exist in the test.
2. Retrieve the row number that corresponds to the VU.
3. Return the base64 encoded value of a combination of username and password column for the determined row.

Base64 encode example

```
public class Base64EncodeCredentials : IExternalVariable
{
    public Base64EncodeCredentials()
    {
        //Do not edit this section.
    }

    /// <summary>
    /// Return true if the variable will be evaluated once per VU iteration. Otherwise will be evaluated on
every parameter.
    /// </summary>
    bool IExternalVariable.EvaluateOnIteration
    {
        get
        {
            return true;
        }
    }

    /// <summary>
    /// Return a combination of server defined prefix and current timestamp.
    /// </summary>
    /// <param name="session">The session context object of the request consuming the variable.</param>
    /// <returns>The source variable value.</returns>
    string IExternalVariable.GetValue(SessionContext context)
    {
        DataTable dt = context.GetDataSource("Credentials");
        if (dt != null)
        {
            int row = context.VUNumber; //Use this option when number of VUs <= number of rows in UserData table
            //int row = context.VUNumber % dt.Rows.Count; //Use this option when number of VUs > number of rows
in UserData table
            if (row < dt.Rows.Count)
                return base64Encode(dt.Rows[row]["Username"].ToString(), dt.Rows[row]["Password"].ToString());
        }
        return null;
    }

    /// <summary>
    /// Base64 encode username and password.
    /// </summary>
    /// <param name="username">Username string</param>
    /// <param name="password">Password string</param>
    /// <returns>Base64 string of username : password.</returns>
    string base64Encode(string username, string password)
    {
        string userpass = username + ":" + password;
        return Convert.ToBase64String(Encoding.ASCII.GetBytes(userpass));
    }
}
```