JSON Web Token Application

# Controllers

**Token Controller**

The Token controller is called using basic authentication. For demonstration purposes, authentication is done using the logged on user name (HttpContext.Current.Request.LogonUserIdentity.Name). The Token Controller uses the BasicAuthentication filter for authentication. Once authenticated, the Token Controller uses the business logic in the JwtToken model to generate and provide the JSON Web Token.

**Data Controller**

The Data controller is used to provide requested data. In this instance, dummy data of “value1” and “value2” are provided in JSON format. The Data controller uses the JwtAuthentication filter. The JwtAuthentication filer requires that the token generated by the Token controller to be passed in the header with the key value of token.

# Filters

**BasicAuthentication**

The BasicAuthentication filter requires the username and password be supplied as credentials. The BasicAuthentication filter uses the business logic in the ApiSecurity model to authenticate the user against the virtual database.

**JwtAuthentication**

The JwtAuthentication filter uses the token provided by the token controller to authorize the user. The JwtAuthentication filter validates the token and ensures the signature matches. The JwtAuthentication filter also ensures that the token has not expired.

# Models

**ApiSecurity**

The ApiSecurity model is used during authentication. This model validates the user against the known users within the VirtualDatabase.

**JwtToken**

The JwtToken model is used to generate the JSON web token. This model is also used to generate a comparison token during authorization to ensure token integrity.

**VirutalDatabase**

For portability and simplified demonstration, a “Virtual Database” class was created to house data to be called by the application. This data would typically be stored in an actual database. This Virtual Database contains data such as Users, User Access Control List, and the Private Key used for signing.

**InputSanitizer**

The InputSanitizer is used to encode incoming input. This is useful for prevention of SQL Injection types of attacks.

