

Fwk.Security

El bloque Fwk.Security presenta un conjunto de componentes al desarrollador que están relacionados a funcionalidades de autorización y autenticación para las aplicaciones. Permite manejo completo de usuarios, roles y reglas de negocio mediante

El bloque Fwk.Security colabora con el desarrollo de la siguiente manera:

- ✓
- ✓ Reducción de escritura de código, basado en bloques probados y estándar
- ✓ Provee de practicas comúnmente utilizada en la mayoría de las cuestiones de seguridad

Componentes comunes en el bloque:

Servicios de seguridad completamente desarrollados

- ✓ Obtener usuarios
- ✓ Autorizar usuarios
- ✓ Autenticar usuarios
- ✓ Obtener Roles y reglas
- ✓ etc.

Componentes basados en standards de Microsoft

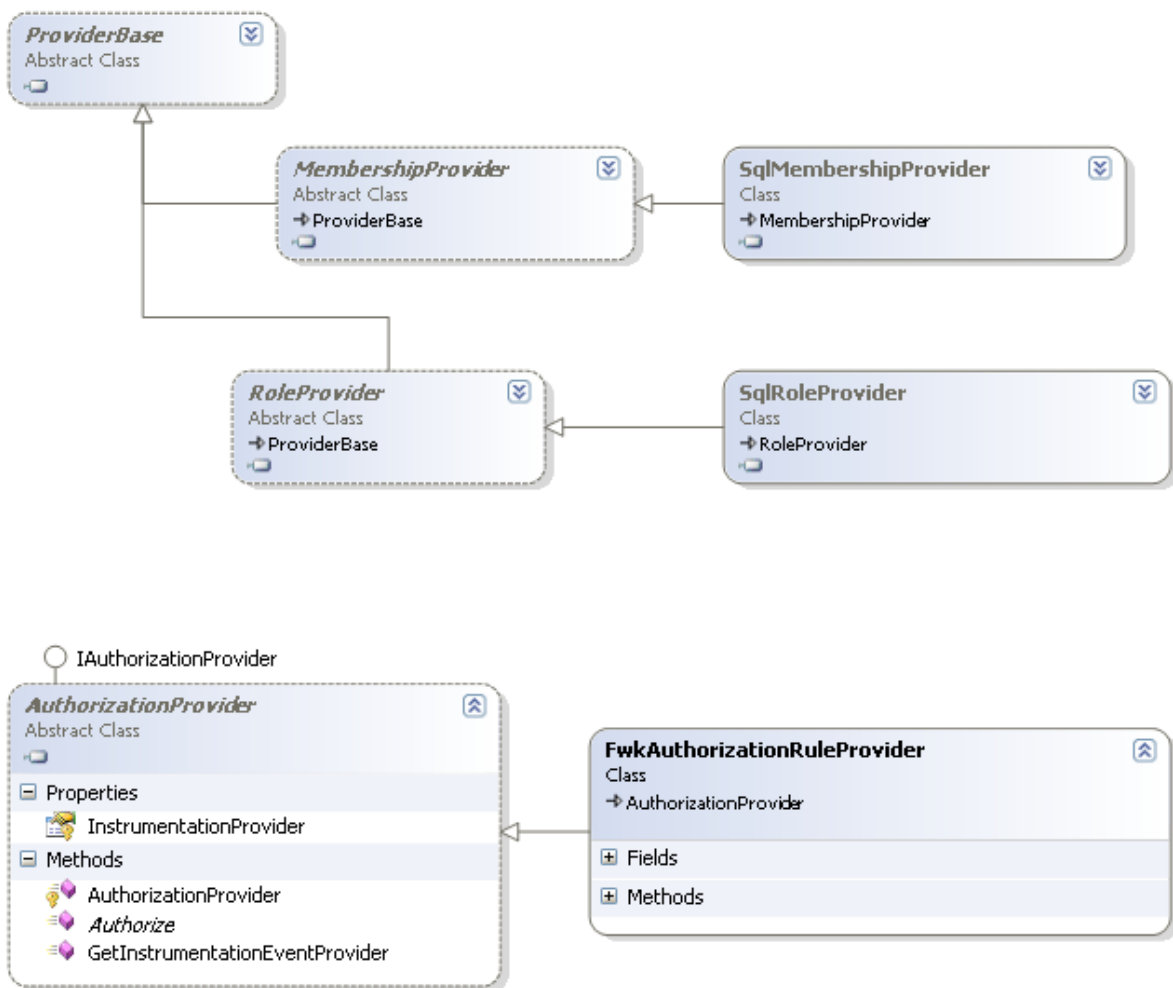
- ✓ FwkMemberships: Con funcionalidades para manejo de Memberships, RolManager Rules
 - Managing Users by Using Membership
 - Managing Authorization Using Roles
- ✓ LDAP : Seguridad basada en Active Directory
 - Fwk.Security.ActiveDirectory: Conjunto de clases helper que abstraen la complejidad de consultas y transacciones LDAP contra controladores de dominio de Active Directory
 - Servicio web Centralizad Security : Servicio que concentra aspectos mas comunes de seguridad basada en Active directory. Utiliza los componentes Fwk.Security.ActiveDirectory
- ✓ Fwk.UI
 - Componente de Logging Win 32 basado en usuarios cutomizados
 - Componente de Logging basado en usuarios cutomizados y Active y Directory
- ✓ Fwk.UI.SecurityController: Componente que enmascara las llamadas a los servicios de seguridad. Comúnmente utilizado por aplicativos clientes web o win 32.-

FwkMembership

Este componente actúa como proxy entre las Memberships, RolManager y la Security Application Blocks (rules)

FwkMembership se configura por medio del modelo de proveedores (providers). Los providers hacen extensible el desarrollo de código ya que facilitan diseñar sus propios proveedores.- La Memberships, RolManager y el modelo de autorización siguen el patrón de providers.

La Fwkmemberships internamente está desarrollada para trabajar con providers ya definidos y se pueden ver en el siguiente grafico:



SqlMembershipProvider

Administra el almacenamiento de información de suscripciones de usuario para una aplicación en una base de datos de SQL Server.

Dada esta cadena de conexión

```

<add name=" LocalSqlServer "
    connectionString="data source=.\SQLEXPRESS;Integrated Security=SSPI;AttachDBFilename=|DataDirectory|aspnetdb.mdf;User
Instance=true"
    providerName="System.Data.SqlClient" />
    
```

Podríamos tener este proveedor:

```
<membership>
  <providers>
    <add
      name="customers"
      type="System.Web.Security.SqlMembershipProvider, ..."
      connectionStringName="customersAmex "
      enablePasswordRetrieval="false"
      enablePasswordReset="true"
      requiresQuestionAndAnswer="true"
      applicationName="customersAmex "
      requiresUniqueEmail="false"
      passwordFormat="Hashed"
      maxInvalidPasswordAttempts="5"
      minRequiredPasswordLength="7"
      minRequiredNonalphanumericCharacters="1"
      passwordAttemptWindow="10"
      passwordStrengthRegularExpression=""
    />
  </providers>
</membership>
```

SqlRoleProvider

Las clases Roles y RolePrincipal que se utilizan dentro de FwkMembership, emplean SqlRoleProvider para proporcionar servicios de administración de funciones para una aplicación usando una base de datos de SQL Server

Configuración:

```
<system.web>

<roleManager defaultProvider="customersAmex "
  enabled="true"
  cacheRolesInCookie="true"
  cookieName=".ASPROLES"
  cookieTimeout="30"
  cookiePath="/"
  cookieRequireSSL="true"
  cookieSlidingExpiration="true"
  cookieProtection="All" >
  <providers>
    <add
      name="customersAmex "
      type="System.Web.Security.SqlRoleProvider"
      connectionStringName=" customersAmex "
      applicationName="customersAmex " />
    </providers>
  </roleManager>
</system.web>
```

FwkAuthorizationRuleProvider

Este proveedor se encarga de brindar todo lo necesario para aplicar reglas de autorización (reglas para usuarios y roles). FwkAuthorizationRuleProvider almacena toda su información en la una base de datos de SQL y trabaja en conjunto a los proveedores anteriores, por lo tanto es necesario tener una base de datos aspnetdb embebida o externa donde alojar las reglas de una determinada aplicación.

Configuración de seguridad

Ver documento **Aquitectura Tecnológica Security Services.pdf**

Instalación de base de datos de seguridad:

Ver documento Fwk.Security.InstallDatabase.pdf

Fwk ActiveDirectory (LDAP Support)

El framework de seguridad soporta autenticación contra un controlador de dominios LDAP.

