

Virtualmind Test

Pre-Deployment Instructions

Milena Lewitzki Dujmusic

Step 1: Setup Database

For this test, the selected database solution is SQL Server 2016. All attached scripts follow T-SQL patterns and rules. Microsoft SQL Server Management Studio is recommended as database manager to perform any database-related action described in this document, such as running scripts.

1. Open a database manager instance and log in to the server where the database will be hosted with the user that will have access to the application's database.
Creating a new user to manage only this database is encouraged.
2. On the database manager instance run attached script *Script_Database_DropAndCreate.sql* to create the database.
3. On the database manager instance run attached script *Script_InitialMigration.sql* to create Users table and Migrations History.
4. On the database manager instance run attached script *Script_PopulateUsers.sql* to populate Users table.

Step 2: Update config file

1. On project source code, folder *VirtualmindTest.WebAPI*, there is a file named *Web.config*. Open it on any available code editor.
2. Find the following section:

```
<connectionStrings>
  <add name="VirtualmindTestContext" connectionString="Data
Source=SERVER; Initial Catalog=VirtualmindTest; Persist Security Info=True;
User ID=DBUser; Password=DBPassword"
providerName="System.Data.SqlClient" />
</connectionStrings>
```

3. Replace values on parameters DataSource, User ID and Password:
 - a. DataSource: replace "SERVER" with the local network IP address, network name or the URL of the server where the SQL Server database was hosted on Step 1.
 - b. User ID: replace "DBUser" with the username of the user configured on Step 1.1.
 - c. Password: replace "DBPassword" with the password of the user configured on Step 1.1.
4. Save changes and close the file.

Notes:

Web.config is the file that should be used to deploy the application. If any other file is preferred, the contents of Web.config should be copied to that file before deployment. Any changes other than connection string values needed to release the application have already been saved to Web.config file.

No additional security or encryption measures have been applied to protect connection data, since it is not part of the goals of this test. Nevertheless, security measures are always encouraged.

Step 3: Publish

Build and publish the application to the desired server with your preferred deployment tool. Take note of the domain address of the server, and subsite URL segments if any, to be able to reach the application through that address later.

Step 4: Services details

Once the database has been setup and the application has been deployed, the services detailed in this section should be accessible. Through this step, words “server” and “site” will appear to resemble the address of the site where the application will be hosted.

Cotizacion

Dolar [GET]: <http://server/site/MyRestfulApp/Cotizacion/Dolar>

Pesos [GET]: <http://server/site/MyRestfulApp/Cotizacion/Pesos>

Real [GET]: <http://server/site/MyRestfulApp/Cotizacion/Real>

User

Note: no additional security or encryption measures have been applied to protect user data, since it is not part of the goals of this test. Nevertheless, security measures are always encouraged.

Get [GET]: <http://server/site/MyRestfulApp/User/Get>

Get/id [GET]: <http://server/site/MyRestfulApp/User/Get/id>

where *id* is the ID number of the desired user.

Create [POST]: <http://server/site/MyRestfulApp/User/Create>

with example body (JSON):

```
{
  "Id":null,
  "Nombre":"Nombre",
  "Apellido":"Apellido",
  "Email":"test@test.com",
  "Password":"test"
}
```

Update [POST]: <http://server/site/MyRestfulApp/User/Update>

with example body (JSON):

```
{
  "Id":2,
  "Nombre":"Nombre2_u",
  "Apellido":"Apellido2_u",
  "Email":"test2_u@test2u.com",
  "Password":"test2Updated"
}
```

Delete/Id [DELETE]: <http://server/site/MyRestfulApp/User/Delete/id>

where *id* is the ID number of the user to delete.