

Project 4 – Credit Card Gateway Web API

This project will give you experience working with a RESTful Web API. In this project, you will create a Web API that provides a service used by retailers to manage credit card accounts and transactions. You will create a Web Application (ASPX pages) that will be the Web API's Client, and a Web API using ASP.NET Core. **Important: only the Web API can access and use the database.**

You must publish your Web API to the **Project4WS** folder since this folder is setup to execute ASP.NET Core Web APIs. Otherwise, your Web API will not work. Publish your Web Application (project with the ASPX pages) to the **Project4** folder.

Requirements:

You need to create a Web Application that utilizes a Web API for a credit card processing gateway. The Web API will be responsible for working with the database, and it will be responsible for verifying, charging, and searching for credit card accounts. The ASPX page should allow the user to enter credit card information, verification information, and purchase information, and record the credit card transaction. The ASPX page (Web API Consumer) should not contain any database connection code; it should rely on the Web API to handle all database operations.

Database Requirements:

1. You need to design and implement a complete data model for the required functionality. You will be graded on your data model implementation.

Class Library Requirements:

1. You need to use component-based software design to create classes for this project that perform the necessary functionalities.

Web Application Requirements:

1. The Web Application needs to allow the user to enter credit card information, billing information, purchase information, and submit the transaction for approval.
 - a. You do not need to create a program where the user buys products. This can be implemented very simply by allowing the user to enter the credit card information, billing information, and a dollar amount to charge the credit card.
2. The Web Application needs to allow the user to add a credit card account and update an existing credit card account. An employee of a company will need the ability to create credit cards for their customers, make changes to the data in the account, and apply payments to the account.

3. The Web Application should allow the user to retrieve all transactions for a specific credit card account.
4. Build a GUI that allows the user to perform these tasks and use the Web Service API to perform the actual operations.

Web API Requirements:

Your Web API will contain at least the following methods:

1. The Web API must contain a Post action method that receives credit card account information including any verification information, billing information, and purchase information. This information must be passed to the Web API using an object or objects containing this information. These object(s) should be defined as custom user-defined classes that you create. This method will be responsible for processing the transaction and recording the transaction in the database. After processing the transaction, the method should return the transaction's status.
 - a. The Web API will return the status of the transaction as an array containing information about the transaction. It should return whether the transaction was accepted or decline, some error number indicating the issue, the date and time of the transaction.
 - b. The Web API client will use this information to display the outcome of the transaction.
2. The Web API must contain another Post action method to add a credit card account to the database. This method is used when creating new credit card accounts.
 - a. You need to build in some security like a unique verification code or API key, so others cannot simply add credit card accounts fraudulently. See Requirement #6.
3. The Web API must contain a Put action method to update an existing credit card account in the database.
 - a. This method is needed for the company to deactivate a credit card that was stolen or cancelled/closed
 - b. This method can also used to apply payments to an account.
4. The Web API needs a Get action method to retrieve all credit card transactions for a specific credit card account.
5. The Web API may require other methods to complete other important operations. This is for you to determine.
 - a. For example, you should not write all the logic in the first few methods. You should separate the functionality by creating a private method to verify the account, another private method to charge an account, etc... This can be also be done using component-based software design, so the logic can be reused in other applications.
6. Secure all Web API methods through the use of an API Key. The API Key is used to allow access to a Web API's method. Remember, a Web API method can be called by any application including applications you did not write. This means someone in class or over the Internet can call your Web API methods and ruin the data in your database. We want to stop unauthorized access to the Web API methods. Therefore, you must write the Web API methods to accept their

normal input parameters plus an extra input parameter that will act as a password (API Key). The code in your Web API method should check this password input parameter and only execute the code in the method if the password/API Key is correct and exit the method immediately when the password/API Key is incorrect.

Design principles:

- a. Retailers like Walmart and Target will use the Web Application to perform credit card transactions and manage credit card accounts for their store. A financial institute like Chase or Citibank would normally create the Web API to allow their retail partners the ability to manage credit card accounts. This Web API can be used by applications written by the retailers to manage their credit cards. Think of this perspective when designing the application.
- b. Provide a consistent and logical navigation system. The user should never have to use the browser's Back and Forward buttons to move between pages.
- c. The user should be presented with an opening screen that presents the various transactions with links to respective pages to perform the selected transaction.
- d. Make your presentation clear to the user, providing on-screen instructions wherever needed both for data entry and error correction. If required data is omitted or entries are incorrect, the user should not have to re-enter data that is already correct.
- e. Create a good data model and implement the data model by creating the necessary tables in the database. The tables I listed above are just for explanation purposes. You are free to make a completely different set of tables since I expect you to implement your own data model. You will be graded on the implementation of your data model.
- f. You need to use a proper naming convention for all controls and in your code. I expect you to properly name your classes, variables, functions, etc...
- g. Your programs should not crash for any reason; it's poor design to have a program crash. Make sure to implement exception handling in appropriate places that can cause errors and handle them gracefully.
- h. You must use stored procedures for all database operations.
- i. Perform server-side input validation where necessary.
- j. **You must use component-based software design. This means writing as much code in classes and functions of classes instead of in the GUI.**