

Bank Web API


Example of ASP.Net Core 3.1, MVC, C#, API, JWT, EF Core, LINQ, jQuery, Bootstrap, MSTest, Selenium

Database : SQL Server Express LocalDB, Database Design, PK, FK, Index, Paging

Function : Create account, Show details, Deposit, Withdraw, Transfer, Fee, Transaction List

Source code : <https://github.com/Ligerking007/BankWebAPI>

Bank System					
API ▾ TRANSACTION ▾ CUSTOMER ACCOUNT ▾					
Customer Account List					
Show	10 ▾	entries	Search: <input type="text"/>		
No ↑↓	Account Name ↑↓	Account No. ↑↓	Balance ↑↓	Created Date ↑↓	Created By ↑↓
1	Somchai Jaiyai	00000000000000000004	0.00	25/06/2021 09:53	System
2	Somchai Jaiyai2	00000000000000000003	0.00	25/06/2021 09:51	System
3	Suchat JaiDee	00000000000000000002	0.00	25/06/2021 09:07	System
4	Jakapan Kanta	00000000000000000001	179.80	25/06/2021 08:42	System
Showing 1 to 4 of 4 entries				Previous	1 Next



Swagger

Supported by SMARTBEAR

Select a definition

Bank Web API

Bank Web API

v1

OAS3

/swagger/v1/swagger.json

Authorize

AuthenAPI

POST

/api/AuthenAPI/GetToken

Get token for authentication and call API

TransactionAPI

POST

/api/TransactionAPI/CreateAccount

Create account by full name and Id card or passport

POST

/api/TransactionAPI/GetAccount

Get account detail by account id

POST

/api/TransactionAPI/DepositMoney

Deposit money to destination account no

POST

/api/TransactionAPI/WithdrawMoney

Withdraw money from source account no

POST

/api/TransactionAPI/TransferMoney

Transfer money from source account no to destination account no

POST

/api/TransactionAPI/GetTransactionListById

Get all transaction by account id

POST

/api/TransactionAPI/GetTransactionList

Get all transaction by grid account id

POST

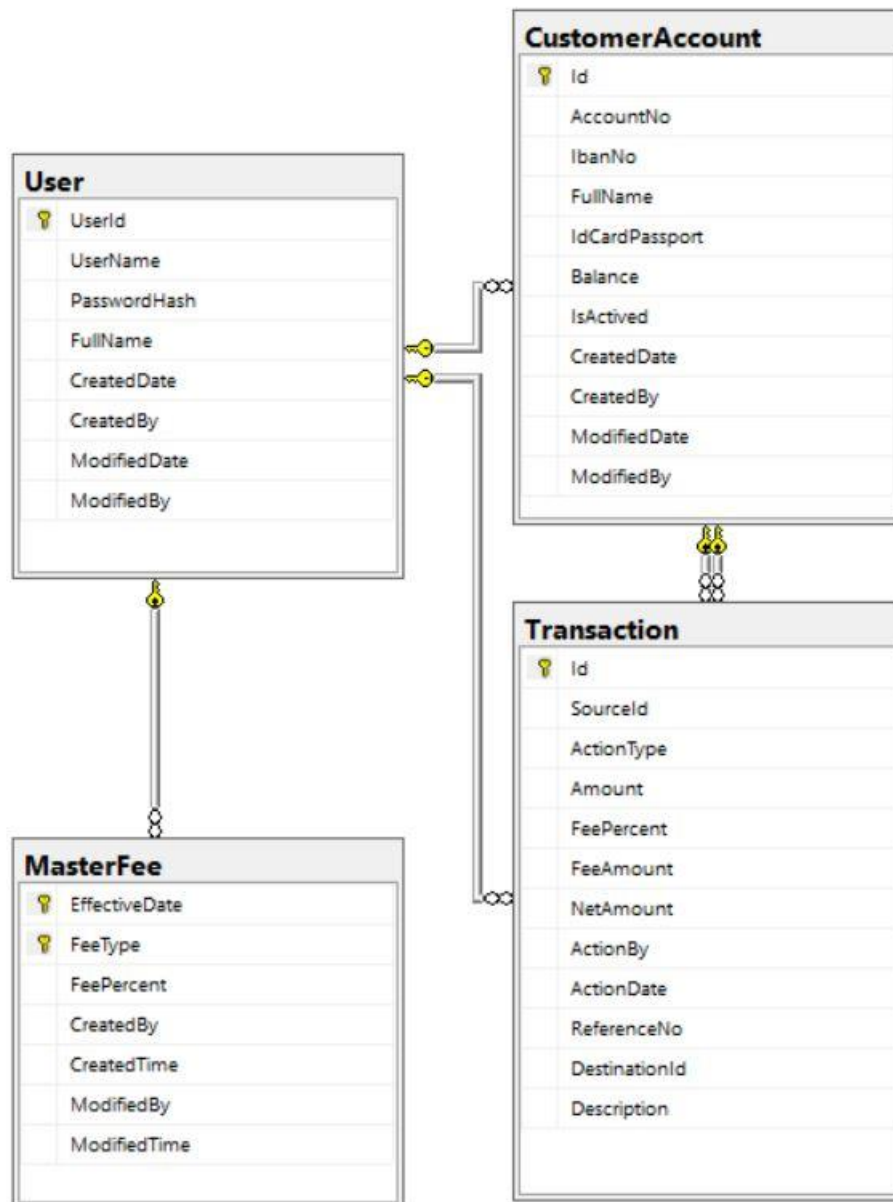
/api/TransactionAPI/GetTransactionListCount

Get transaction count number by account id (for paging)

POST

/api/TransactionAPI/GetCustomerAccountList

Get all account details



Database Design, PK, FK, Index

Bank System

API ▾ TRANSACTION ▾ CUSTOMER ACCOUNT ▾

👤 ▾

Create customer account

Customer name

New Normal

ID Card / Passport

1559900111126

Cancel

Create

Bank System

API ▾ TRANSACTION ▾ CUSTOMER ACCOUNT ▾

👤 ▾

Deposit money

SourceNo

Jakapan Kanta - Rp5,000,179.70 ▾

Amount

Rp

2000

Description (optional)

2K

Cancel

Deposit

Bank System

API ▾ TRANSACTION ▾ CUSTOMER ACCOUNT ▾

👤 ▾

Withdraw money

SourceNo

Jakapan Kanta - Rp5,002,177.70 ▾

Amount

Rp

50000

Description (optional)

50k

Cancel

Withdraw

Bank System

API TRANSACTION CUSTOMER ACCOUNT

Transfer money

SourceNo

Jakapan Kanta - 00000000000000000001 - ₪5,002,177.70

DestinationNo

00000000000000000004

New Normal - 00000000000000000004 - ₪0.00

Amount

₪ 89000

Description (optional)

89k

Cancel

Transfer

Bank System

API TRANSACTION CUSTOMER ACCOUNT

Jakapan Kanta

Account details

ID Card / Passport

1559900111123

Current balance

₪179.80

Account number

00000000000000000001

IBAN number

NL40ABNA6579776978

Date of creation

25/06/2021 08:42

Last transactions

Show 10 entries

Search:

No	Action Type	Source No	Destination No	Amount	Fee Percent	Fee Amount	Net Amount	Action Date	Account
1	Withdraw	00000000000000000001		20.00	0.00	0.00	20.00	29/06/2021 18:16	Sy
2	Deposit		00000000000000000001	200.00	0.10	0.20	199.80	25/06/2021 09:11	Sy

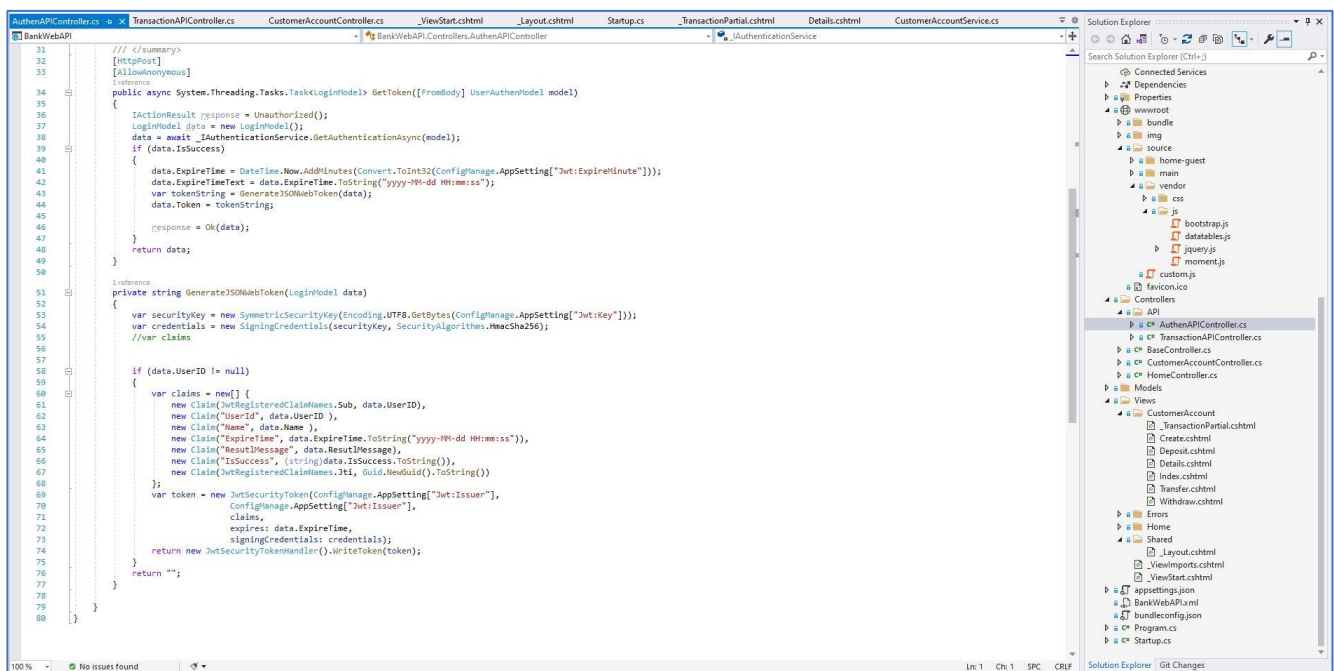
Showing 1 to 2 of 2 entries

Previous

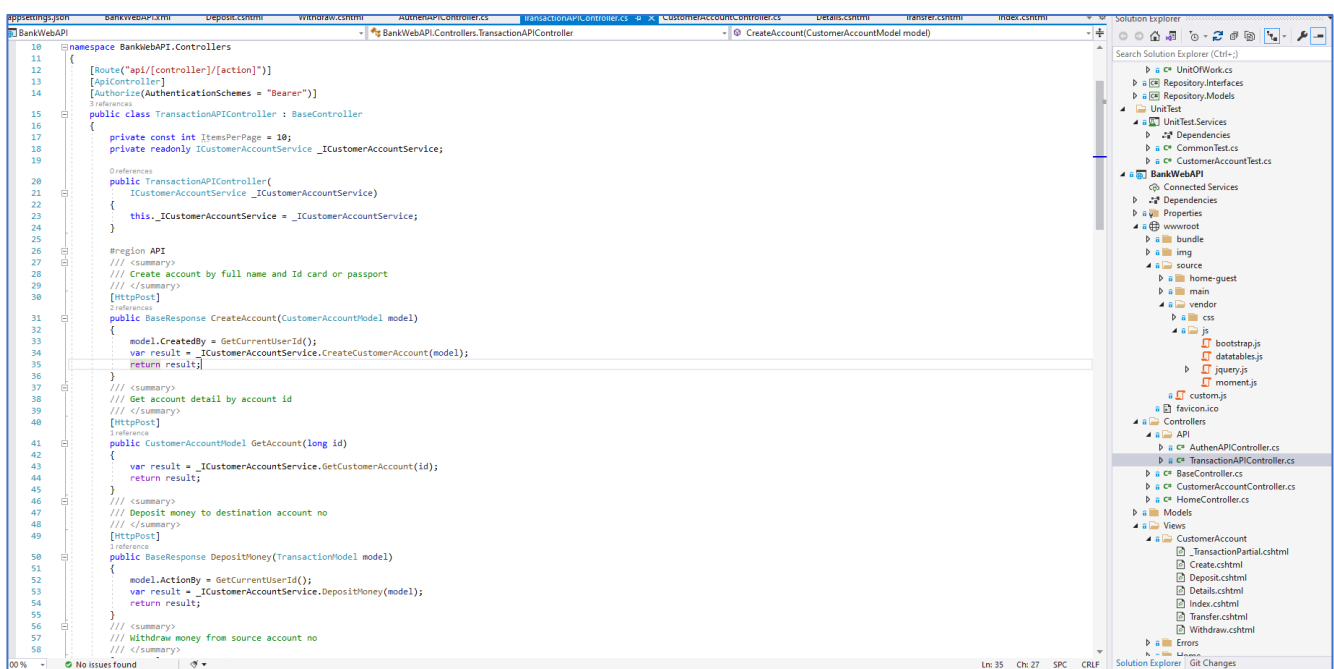
1

Next

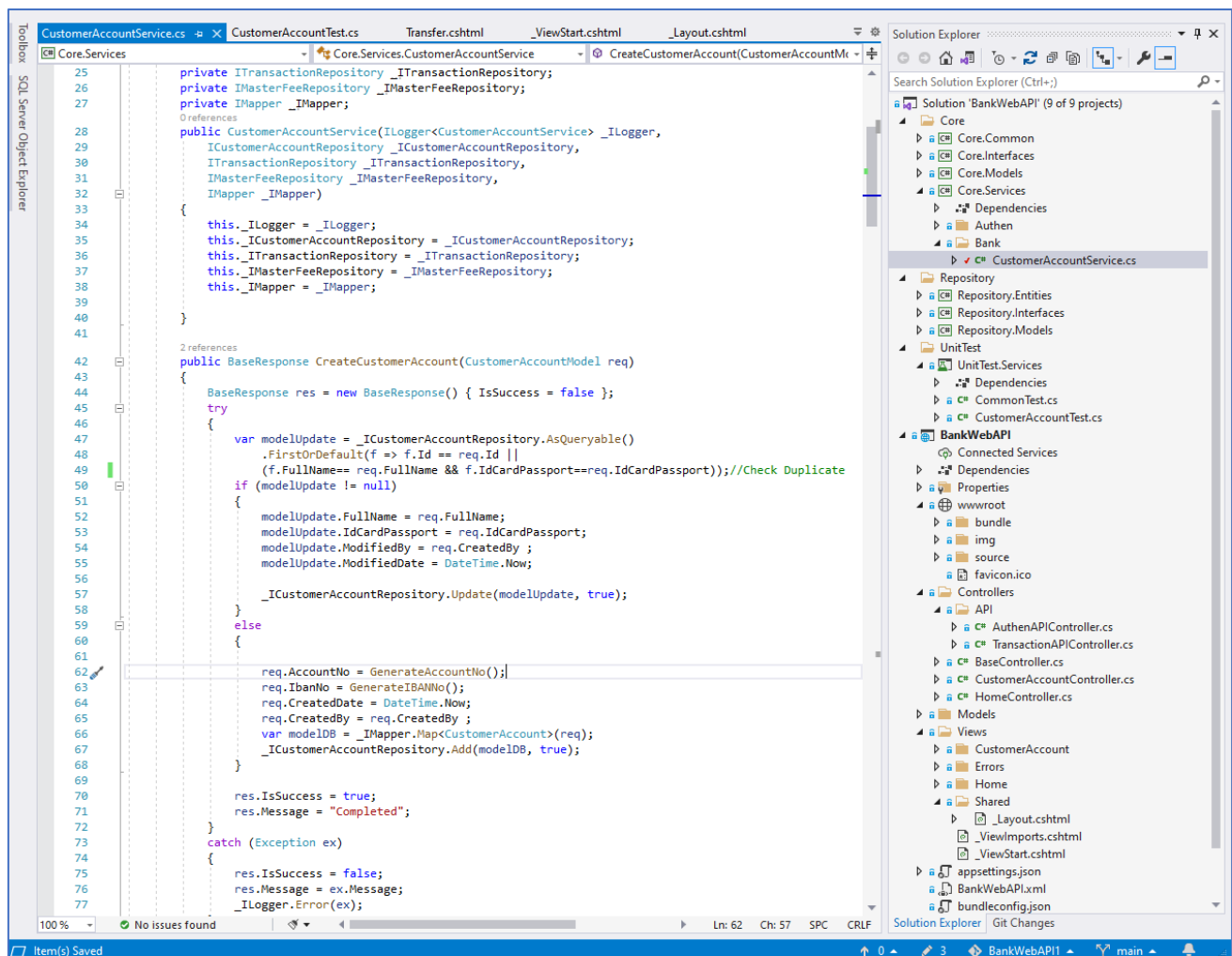
Source Code Example



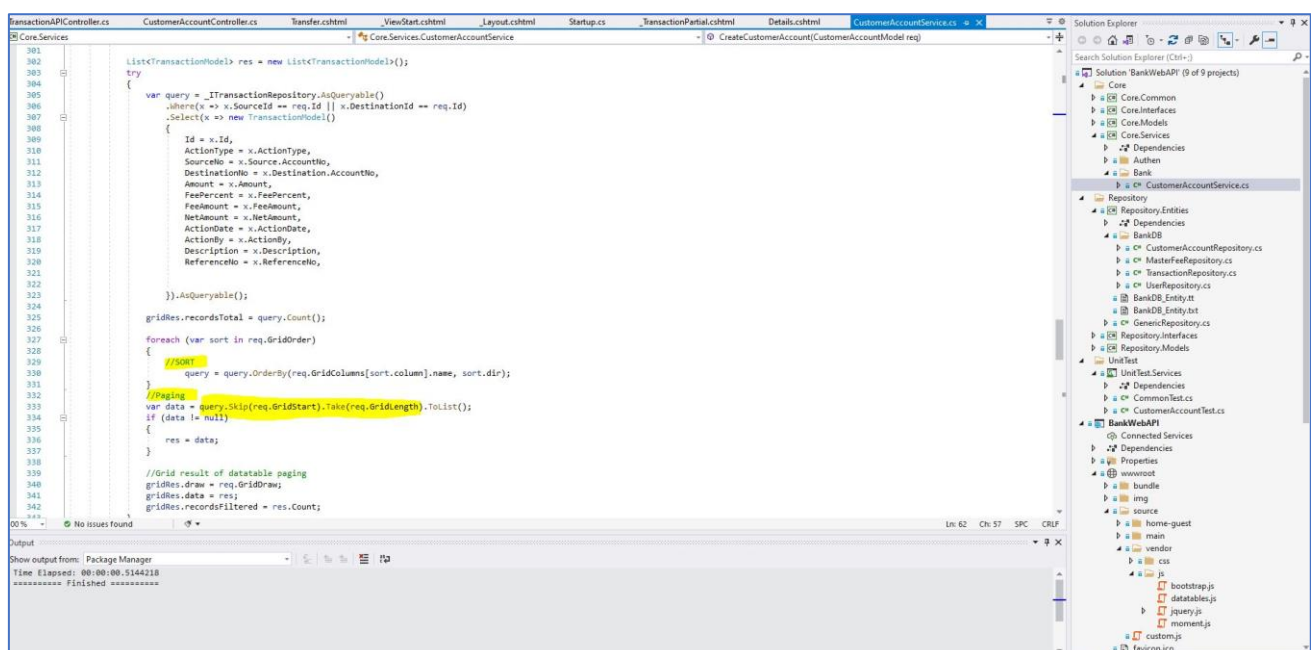
Authentication API



Controller API



Dependency Injection & Services



Server Paging and Sorting

```

--Execute [dbo].[SP_GetCustomerAccountList] '1','Id','Desc','0','2',1
ALTER PROCEDURE [dbo].[SP_GetCustomerAccountList]
(
    @IsActive bit = 1 ,
    @SortBy nvarchar (100) = null ,
    @SortDirection nvarchar (10) = null ,
    @PageStart int = 0 ,
    @PageSize int = 1000 ,
    @TotalRows int output
)
AS
BEGIN
    -- Build WHERE Clause
    DECLARE @WhereClause [nvarchar] (2000)
    SET @WhereClause = ' IsActive = '' + convert(nvarchar, @IsActive) + '' '
    DECLARE @OrderBy [nvarchar] (100)
    SET @OrderBy = @SortBy + ' ' + @SortDirection
    IF (@SortBy is null or LEN(@SortBy) = 0)
    BEGIN
        -- default order by to first column
        SET @OrderBy = 'Id Desc'
    END
    -- SQL Server 2005 Paging
    declare @SqlCount as nvarchar(max)
    declare @SqlTable as nvarchar(max)
    declare @SQL as nvarchar(max)
    -- get row count
    SET @SqlCount = ' SELECT @TotalRows = COUNT(*) '
    -- sql : from table and where ...
    SET @SqlTable = ' FROM [BankDB].[dbo].[CustomerAccount] '
    IF LEN(@WhereClause) > 0
    BEGIN
        SET @SqlTable = @SqlTable + ' WHERE ' + @WhereClause
    END
    SET @SqlCount = @SqlCount + @SqlTable
    EXEC sp_executesql @SqlCount, N'@TotalRows int OUTPUT', @TotalRows OUTPUT ;
    --print @TotalRows;
    -- Get Data
    SET @SQL = ' SELECT '
    SET @SQL = @SQL + ' ROW_NUMBER() OVER (ORDER BY ' + @OrderBy + ') asRowIndex'
    SET @SQL = @SQL + ' , *' --All field
    -- query @SqlTable : from table where ...
    SET @SQL = @SQL + @SqlTable
    SET @SQL = @SQL + ' ORDER BY ' + @OrderBy
    SET @SQL = @SQL + ' OFFSET ' + convert(nvarchar, @PageStart) + ' ROWS FETCH NEXT ' + convert(nvarchar, @PageSize) + ' ROWS ONLY'
    -- Get Data
    EXEC sp_executesql @SQL
    --PRINT @SQL
END

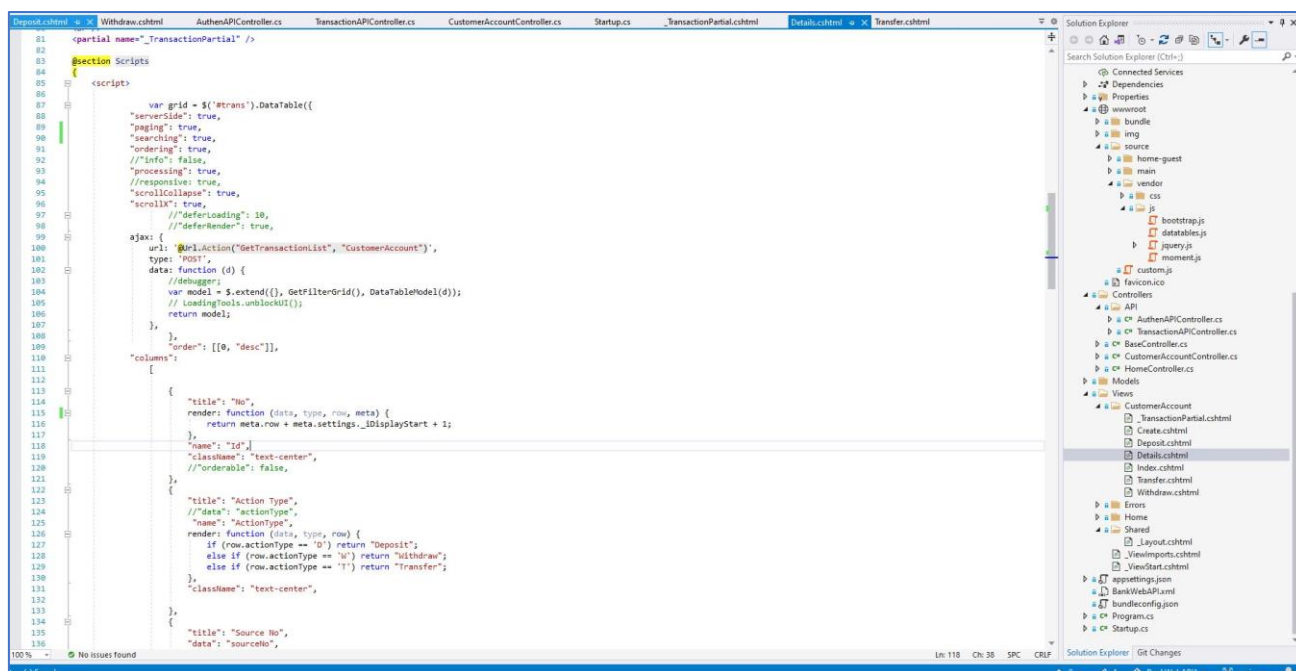
```

Store Procedure Paging and Sorting

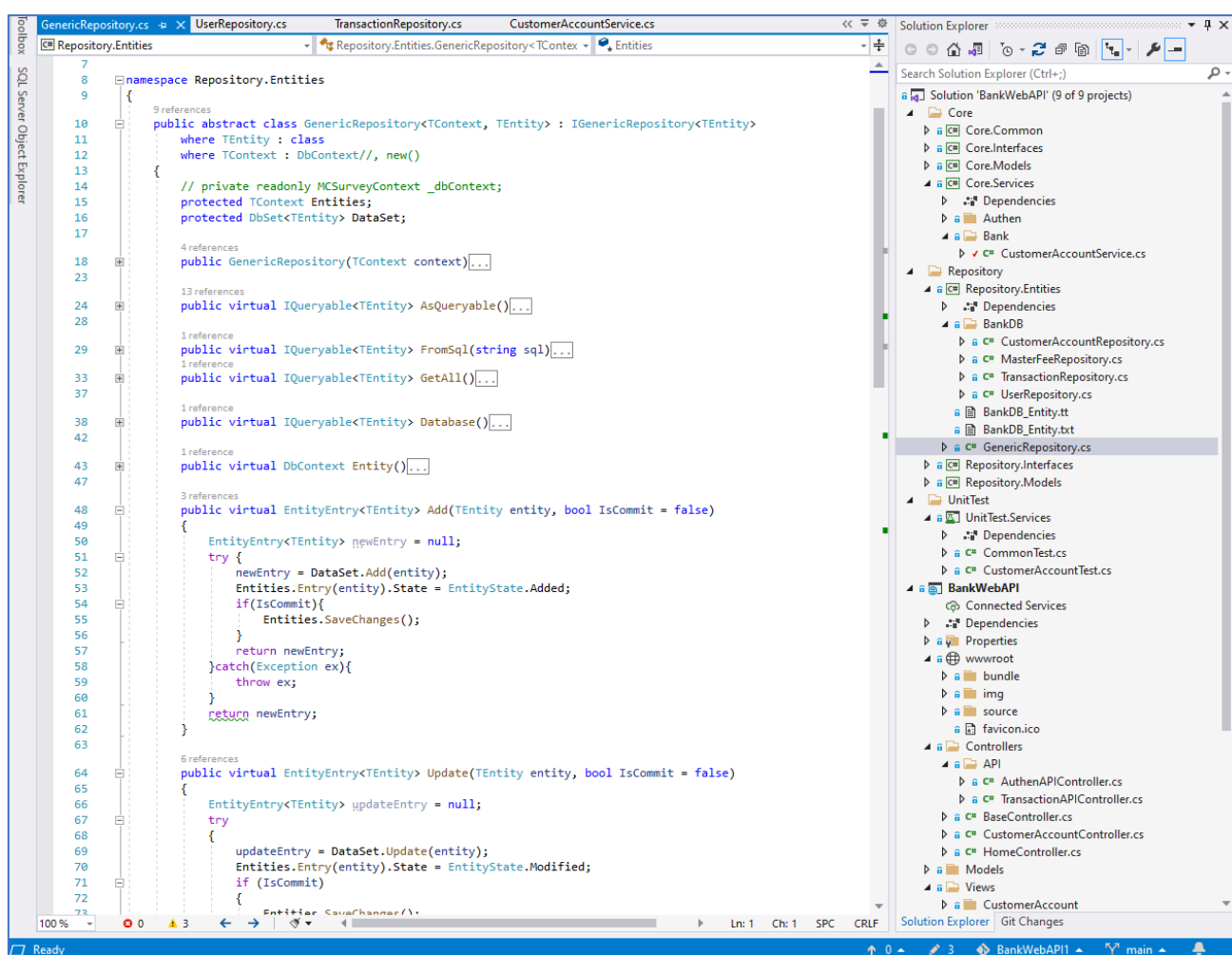
The screenshot displays the Visual Studio IDE with the following components:

- Toolbox:** SQL Server Object Explorer.
- Editor:** The file `Transfer.cshtml` is open, showing HTML code for a modal dialog. The code includes:
 - A modal container with `id="editModal"` and `class="modal fade"`.
 - A header section with a title "Edit account details" and a close button.
 - A body section with a form containing:
 - A hidden input for `accountId` bound to `Model.CustomerAccountModel.Id`.
 - A text input for `name` with a required attribute.
 - A text input for `idCardPassport` with a required attribute.
 - Buttons for "Cancel", "OK", and "Close".
 - A partial view `_TransactionPartial` is included at the bottom.
- Solution Explorer:** Shows the project structure for `BankWebAPI` (9 of 9 projects). The `BankWebAPI` folder is expanded, showing:
 - `Connected Services`
 - `Dependencies`
 - `Properties`
 - `wwwroot` (containing `bundle`, `img`, `source`, and `favicon.ico`)
 - `Controllers` (containing `API`, `AuthenAPIController.cs`, `TransactionAPIController.cs`, `BaseController.cs`, `CustomerAccountController.cs`, and `HomeController.cs`)
 - `Models`
 - `Views` (containing `CustomerAccount` and `Errors`)

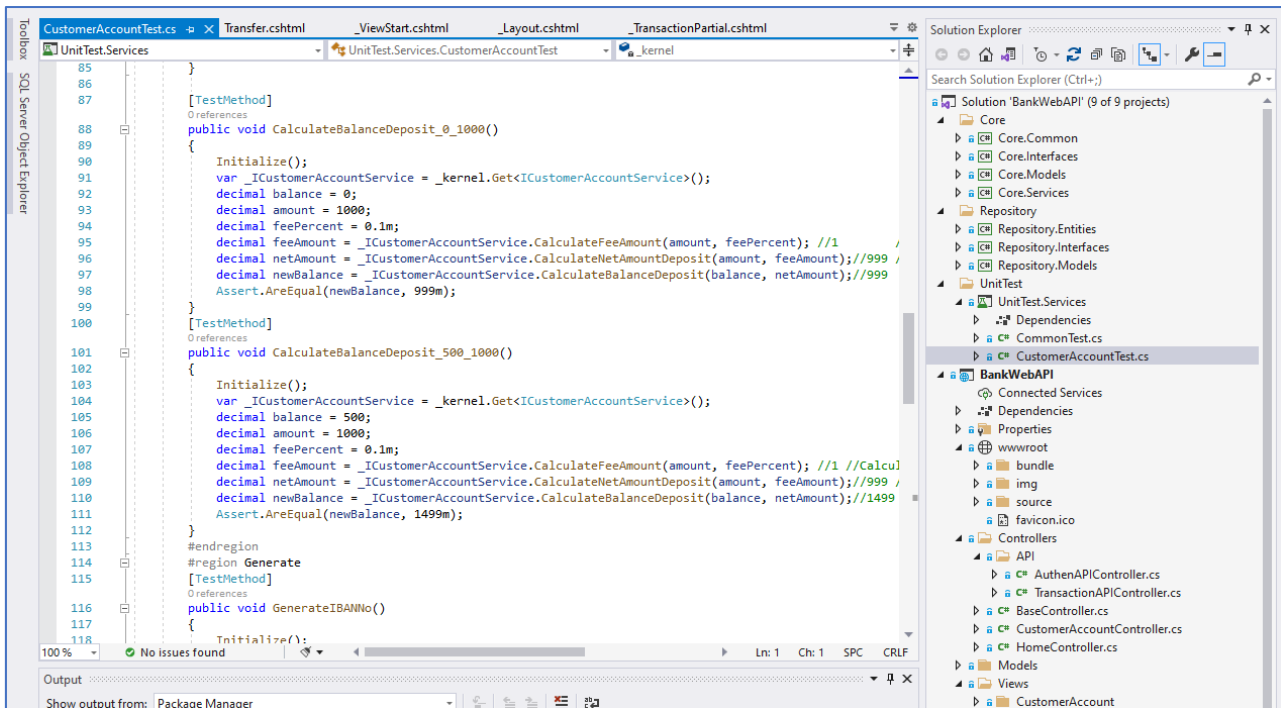
HTML & Bootstrap



Jquery



Repository (Connect Database)



Unit Test

Other Technique

Memory controlling

Dispose object (clears memory)

Garbage collection (clears memory)

String builder

Date conversion

Null validation

SQL Transaction (Begin, Commit, Rollback)

Reduce looping

Reduce connect database

Security coding (OWASP10), JWT

OOP concept

Clean code

Load balancing

Scaling