

# Rick Donalson's Code Samples

## **Financial Planner Forecast Ledger (Angular & .Net Core)**

- The Financial Planner is an Angular 15 SPA with a .NET Core API are Web Apps hosted on Azure Cloud. The UI is a mix of PrimeNg, Material, Bootstrap 5 and NgRx-Redux. The API is .NET Core using Entity Framework Core with a Code First approach. The UI Login and the API security are handled by Microsoft Intra ID (Active Directory). The Database is SQL Server also hosted on Azure.
- Financial Planner Abstract: This Application enables You to keep your checking or saving account solvent out to any future point in time. You enter expenses and incomes with a starting amount, then by entering a time frame, you can easily see a total by end of that time frame. These calculated items can be displayed in either a graphic or ledger format.
  1. Once logged in go to the Item Detail section and add your credits (Income sources: paychecks, tax refunds, etc.) and the Period of Occurrence. Be sure to enter the amounts you get after taxes are taken out (take-home pay).
  2. Then do the same with your debits (Bills, mortgage, car payment, etc.). Enter debit amounts in positive values, the program will take of the subtracting.
  3. Then enter your starting Balance.
  4. Then go to the Display Section. With either the Timeline or the Ledger Display & Export enter a time frame and click enter. With Ledger Display & Export you can export your data to Excel.

### **\*\* Financial Planner**

- Demo: [Financial Planner Forecast Ledger](#)
  - Demo -> Upon request
- GitHub: [Source Code](#)
- **Documentation**
  - [Financial Planner Guide](#) or [Using the Excel Export Helper](#)

### **\*\* Front End (Angular)**

- **Components, Services & Models**
  - [ItemListComponent](#)
  - [ItemEditComponent](#)
  - [ItemService](#)
  - [Item](#)
  - [GenericValidator](#)
  - [MessageUtilService](#)
- **NgRx & Redux**
  - [Item State](#)
  - [ItemType State](#)
  - [Period State](#)
- **Markup**
  - [item-list.component.html](#)
  - [item-edit.component.html](#)
- **Routing & Lazy Loading**
  - [AppRoutingModule](#)
  - [ItemDetailRoutingModule](#)
- **Styles**
  - [app.component.scss](#)
  - [display-main.component.scss](#)

- *Misc.*
  - [form-utility-items.json](#)

## ***\*\* Back End (.Net Core)***

- *API*
  - [ItemsController](#)
  - [RepoItems](#)
  - [RepoPeriods](#)
- *EF Core: Code First*
  - [FPFLContext](#)
  - [Item](#)
  - [Migrations](#)
- *T-SQL Stored Procedure*
  - [spCreateLedgerReadout.sql](#)
    - [Entities](#)