Architecture Summary

Ministocks, the android widget, follows a loose "Model View Controller" architecture. The structure is not clearly defined but the widget clearly uses MVC methodologies.

The model is represented by the "StockQuoteRepository" class. This class in itself does not access data directly but consolidates data retrieved by multiple stock APIs. The REST API calls are all contained within classes in the "dataaccess" package. Contrary to the naming convention, this is class is not representative of a repository pattern but a collection of stock quote data. The application does not use any persistent memory but instead uses interfaces to mock storage. The Storage interface is simply a wrapper for a hashmap. Other sources of data separate from the stock quote data use this Storage interface to hold portfolio and preference data.

The controller layer is represented by two main classes: WidgetProviderBase and WidgetView. The "WidgetProviderBase" class refers to the models to create the base configuration of the widget and contains event listeners to refresh stock quotes displayed in the widget view . The "WidgetView" instantiates the view with data provided by all data sources using the row template provided by the "WidgetRow" class and contains event listeners that change the font end format on touch.

The View is initially set via the "ConfigureActivityBase" class determines the size and functionality of the widget and populates it with model data mapped through the "WidgetProviderBase". The end result is that the appropriate XML view is displayed.