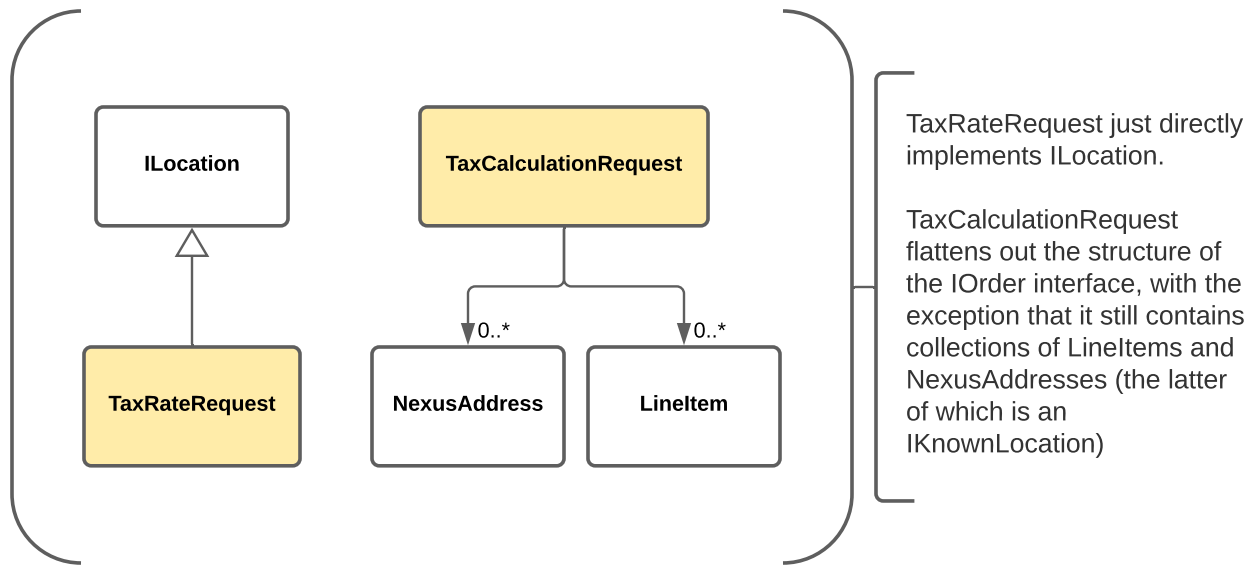
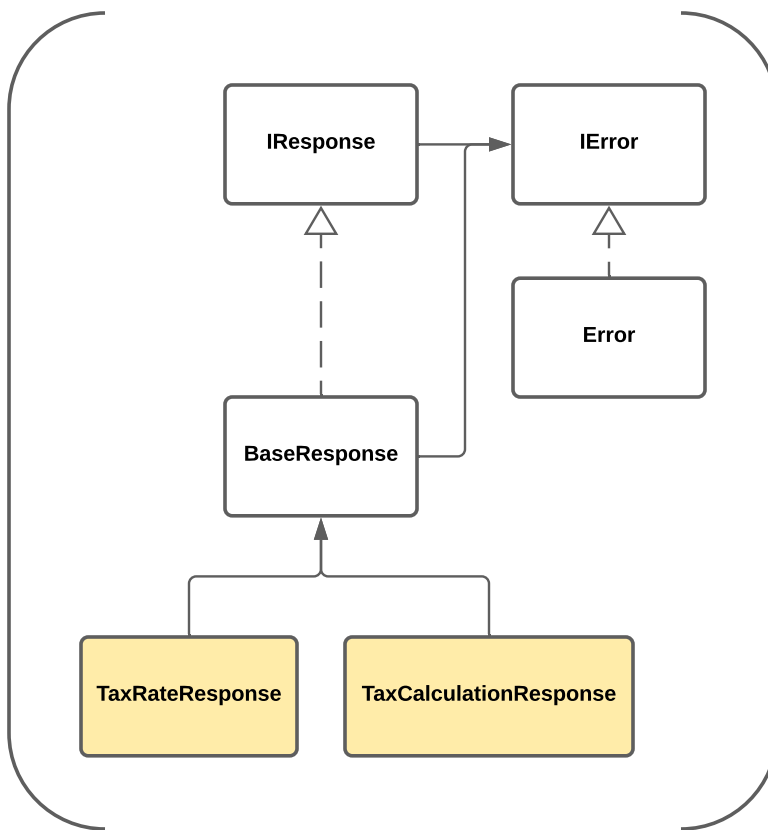


An IOrder (implemented by Order) contains zero or more Lineltems, and zero or more IKnownLocations, which extends ILocation by adding an Id, and is implemented by NexusLocation.

It contains two ILocations: the "to" and the "from", and also contains some raw order data such as amount, shipping, etc.

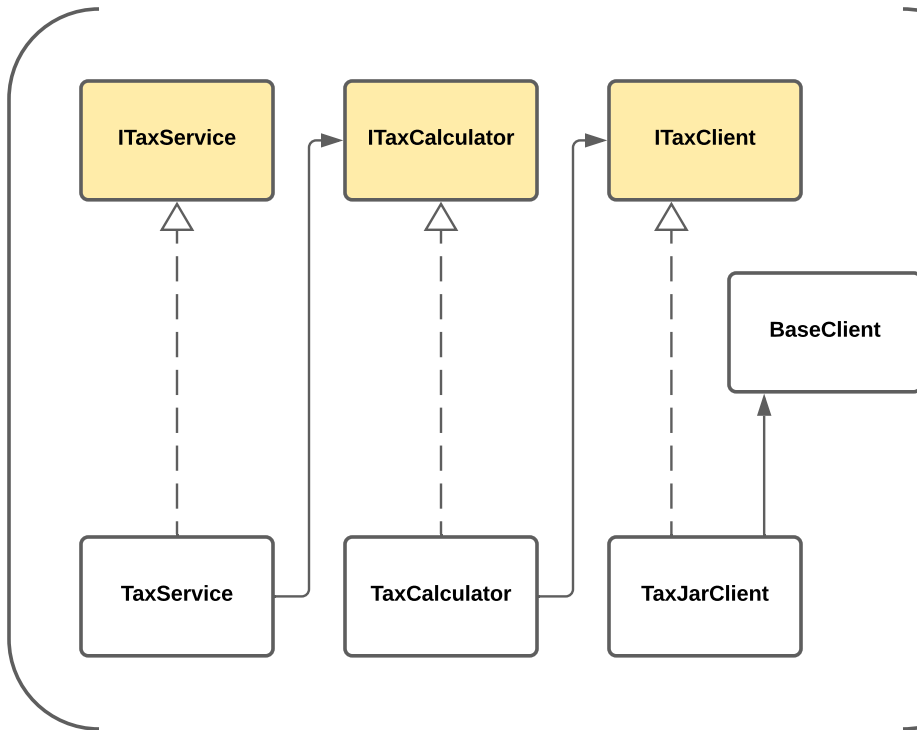




IResponse contains an **IError** (which is the minimum amount of info we need for any given request to tell if it's success/failure and try to get some idea of why - so capture exceptions, error messages, etc).

BaseResponse implements **IResponse**, and all responses derive from it, so all responses can track error information.

Otherwise, the individual responses need to correspond with the API responses.



Note: Use DI everywhere.

TaxService implements ITaxService and relies on an injected ITaxCalculator. Otherwise, it is a passthrough. Additional logic may be utilized later to provide for more than one tax calculator, and to decide which tax calculator to use.

ITaxCalculator provides operations to lookup a taxrate for a given ILocation, or to provide the taxes to collect from a given IOrder.

In both cases, convert those interfaces into the corresponding requests and hand that down to the ITaxClient to fetch (in this case, setting up communication to taxjar's API).

ITaxClient handles any web-specific necessities above and beyond what is provided by the BaseClient (where the actual requests are made, responses and errors handled, etc).

Successful requests from the tax client return the needed value. Unsuccessful requests log errors and return **NaN**.