Design and design decisions

Design diagram

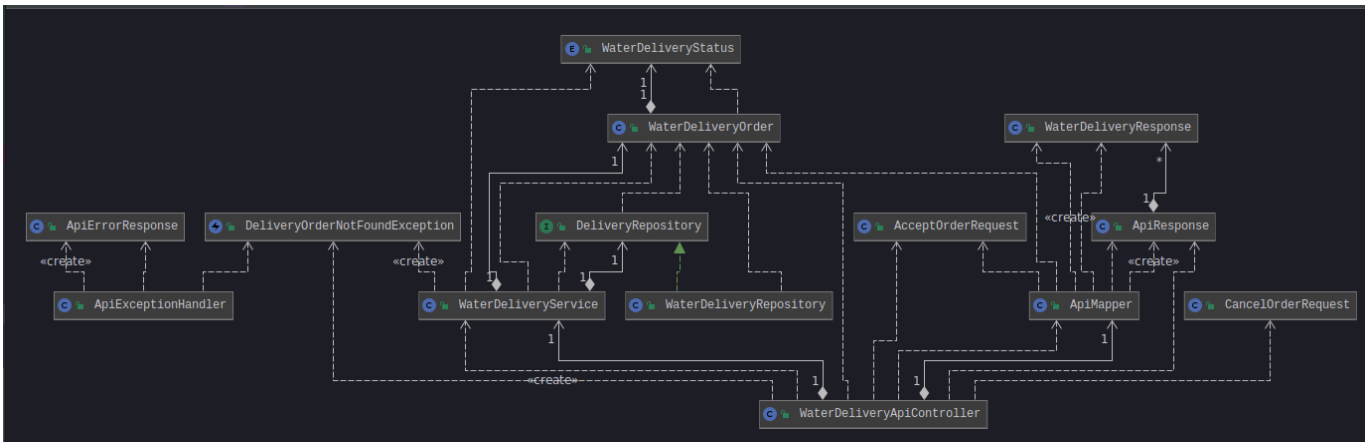

Figure 1: Class diagram

Some design decisions

Representation of dates

> Dates across the entire system is Java's `LocalDateTime` and when printed or serialized is represented in ISO-8601 format.

Water delivery status

> WaterDeliveryStatus is represented as an enum with an `allowCancel` attribute. This attribute indicates which statues can be cancelled, taking away the complexities of if/else conditional statements. It also makes it easy to add new delivery statues without having to change business logic.

Error handling

> Maintained a centralised copy of the list of possible errors known to the application aimed at reducing error code duplication. This can be found in Errors.java Each error object contains a code and the description of the error can be easily retrieved across the entire system without having to duplicate error descriptions. This can also be used as a customer reference code for back-office investigation in the event that an error occurred while trying to interact with the APIs.

Bulk placing of orders

> In the event that one or more entries is invalid while performing bulk placing/cancelling of orders, those valid entries before the rejected entry would be accepted and processed.

Orders in REQUESTED state but past delivery end date

> When the application starts up, those orders which are in a `WaterDeliveryStatus.REQUESTED` state but are past `DeliveryTimeWindow.endDate` are automatically marked as `WaterDeliveryStatus.CANCELLED`

Orders in REQUESTED state but are within DeliveryTimeWindow

> When the application starts up and during execution, those orders which are in a `WaterDeliveryStatus.REQUESTED` state but are within the `DeliveryTimeWindow` are automatically marked as `WaterDeliveryStatus.IN_PROGRESS`. For example, if the system is actively running from 27-Dec-2019 @ 10:45am to 30-Dec-2019 @ 09:10am, those orders which are active within that time frame is marked as `WaterDeliveryStatus.IN_PROGRESS`

Batch Date

> When developing state machines or systems that are clock/time dependent. There's the notion of a configurable BatchDate. The reasoning behind this is to make delivery order batch processing to be independent of the application clock. In the event that the system/application fails you still want to process those delivery backlogs that couldn't be processed because of system downtime.