**Toogood Transform System**

**Detail Design**

**Written By: Eric Ho**

**Date: June 12, 2019**

Contents

[**Overview** 5](#_Toc11238068)

[**System Overview** 5](#_Toc11238069)

[**Data Format** 5](#_Toc11238070)

[**Source 1:** 5](#_Toc11238071)

[**Source 2:** 5](#_Toc11238072)

[**Output:** 6](#_Toc11238073)

[**Assumption** 6](#_Toc11238074)

[**Class Diagram** 7](#_Toc11238075)

[**Sequence Diagram** 8](#_Toc11238076)

[**Class Definition** 9](#_Toc11238077)

[**1.** **Transform** 9](#_Toc11238078)

[1.1. Details 9](#_Toc11238079)

[1.2. Operations 9](#_Toc11238080)

[**2.** **Builder** 9](#_Toc11238081)

[2.1. Details 9](#_Toc11238082)

[2.2. Operations 9](#_Toc11238083)

[**3.** **Source1OutputBuilderImpl** 10](#_Toc11238084)

[3.1. Details 10](#_Toc11238085)

[3.2. Attributes 10](#_Toc11238086)

[3.3. Operations 11](#_Toc11238087)

[**4.** **Source2OutputBuilderImpl** 12](#_Toc11238088)

[4.1. Details 12](#_Toc11238089)

[4.2. Attributes 12](#_Toc11238090)

[4.3. Operations 12](#_Toc11238091)

[**5.** **Input** 13](#_Toc11238092)

[5.1. Details 13](#_Toc11238093)

[5.2. Operations 14](#_Toc11238094)

[**6.** **Source1DtoInputImpl** 14](#_Toc11238095)

[6.1. Details 14](#_Toc11238096)

[6.2. Attributes 14](#_Toc11238097)

[6.3. Operations 14](#_Toc11238098)

[**7.** **Source2DtoInputImpl** 14](#_Toc11238099)

[7.1. Details 14](#_Toc11238100)

[7.2. Attributes 15](#_Toc11238101)

[7.3. Operations 15](#_Toc11238102)

[**8.** **OutputDto** 15](#_Toc11238103)

[8.1. Details 15](#_Toc11238104)

[8.2. Attributes 15](#_Toc11238105)

[8.3. Operations 15](#_Toc11238106)

[**9.** **OutputTransformer** 15](#_Toc11238107)

[9.1. Details 16](#_Toc11238108)

[9.2. Operations 16](#_Toc11238109)

[**10.** **OutputTransformerFactory** 17](#_Toc11238110)

[10.1. Details 17](#_Toc11238111)

[10.2. Attributes 17](#_Toc11238112)

[10.3. Operations 17](#_Toc11238113)

[**11.** **Constants** 17](#_Toc11238114)

[11.1. Details 17](#_Toc11238115)

[11.2. Attributes 17](#_Toc11238116)

[**12.** **ValidateResults** 18](#_Toc11238117)

[12.1. Details 18](#_Toc11238118)

[12.2. Attributes 18](#_Toc11238119)

[**13.** **CurrencyConstraint** 18](#_Toc11238120)

[13.1. Details 18](#_Toc11238121)

[**14.** **IdentifierConstraint** 18](#_Toc11238122)

[14.1. Details 18](#_Toc11238123)

[**15.** **TypeConstraint** 18](#_Toc11238124)

[15.1. Details 18](#_Toc11238125)

[**16.** **CurrencyValidator** 18](#_Toc11238126)

[16.1. Details 18](#_Toc11238127)

[16.2. Operations 18](#_Toc11238128)

[**17.** **IdentifierValidator** 19](#_Toc11238129)

[17.1. Details 19](#_Toc11238130)

[17.2. Operations 19](#_Toc11238131)

[**18.** **TypeValidator** 19](#_Toc11238132)

[18.1. Details 19](#_Toc11238133)

[18.2. Operations 19](#_Toc11238134)

# **Overview**

## **System Overview**

Toogood Transform System is used for transform input, from multiple different source in different format, to a generic standard output that could be used by other part of the application.

Currently the system only supports 2 different input formats, but developer can easily extend it to support more input source

## **Data Format**

### **Source 1:**

**Identifier**,**Name**,**Type**,**Opened**,**Currency**

123|AbcCode,My Account,2,01-01-2018,CD

**Identifier**

* The identifier column has both a numeric id and an account code, separated by the “|” character
* Need to parse out the account code and output it

**Type**

* 1 = Trading
* 2 = RRSP
* 3 = RESP
* 4 = Fund

**Currency**

* CD = CAD
* US = USD

### **Source 2:**

**Name**,**Type**,**Currency**,**Custodian Code**

My Account,RRSP,C

**Currency**

* C = CAD
* U = USD

**Custodian Code** = Account Code

### **Output:**

**AccountCode**,**Name**,**Type**,**Open Date**,**Currency**

AbcCode,My Account,RRSP,2018-01-01,CAD

**Open Date** is optional field

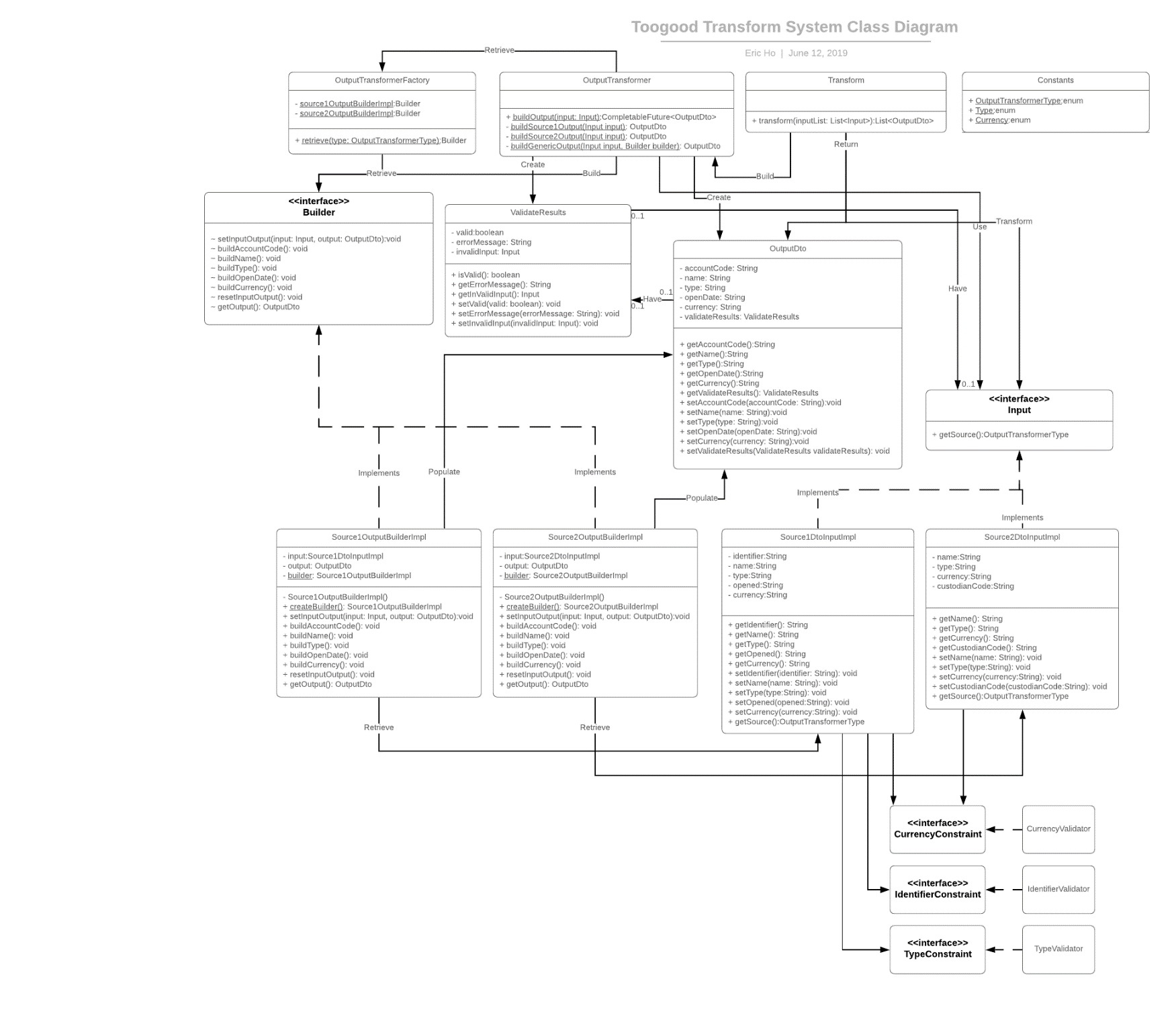
## **Assumption**

1. "Source Account File" means Source 1
2. Source input value as been saved to an object in a separate system
3. Transformed data will only saved to an object and ready to be used by another system, not return in a csv file

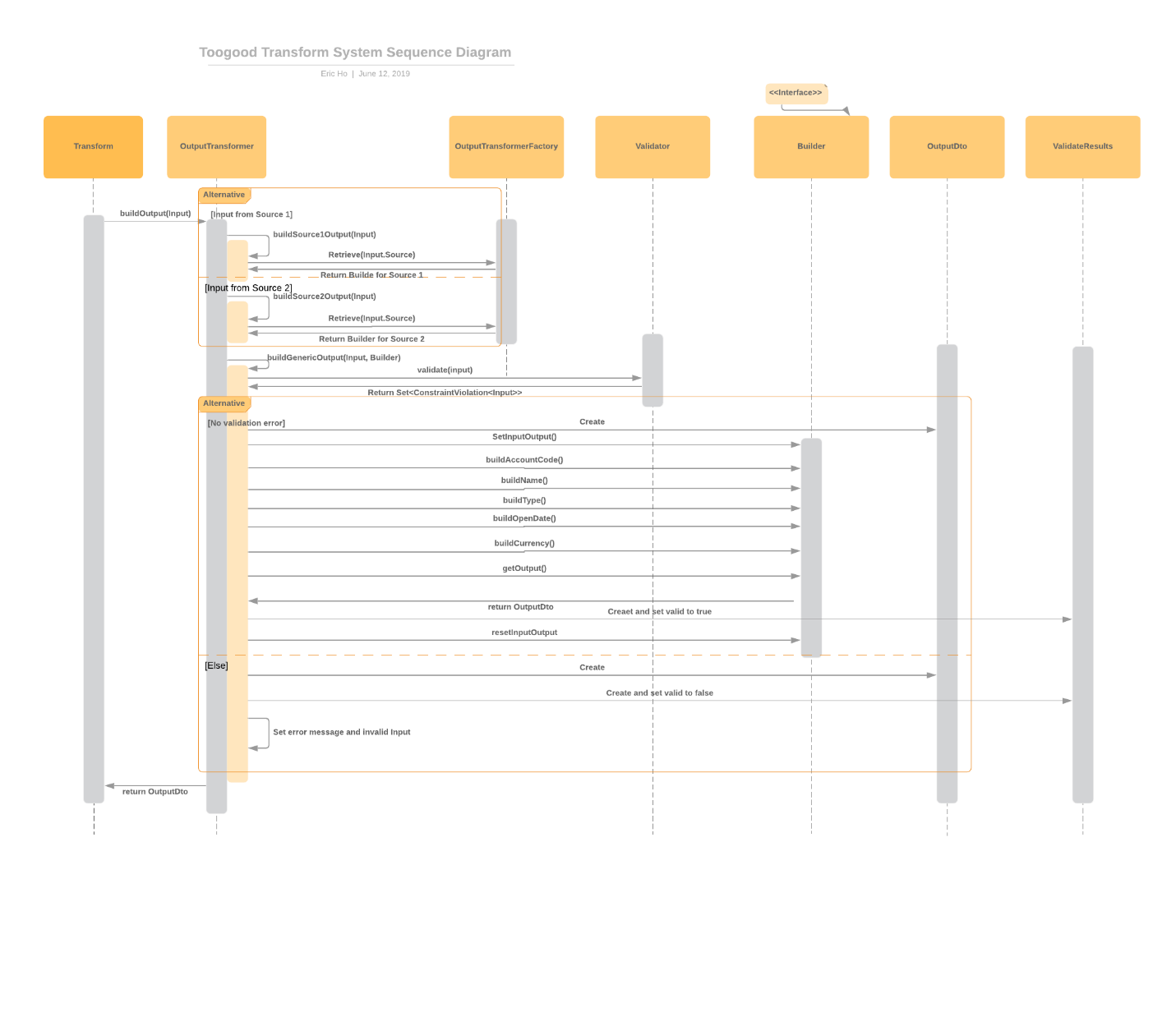
Note: attached the original requirement



# **Class Diagram**



# **Sequence Diagram**



# **Class Definition**

1. **Transform**
   1. Details

|  |  |
| --- | --- |
| Package | com.toogood.data.transform |
| Class Name | Transform |
| Purpose | The main purpose of the class is to transform a list of class implement Input interface and return a list of OutputDto objects for other part of the system to use  Assumption: Data from Source 1 will save to Source1DtoInputImpl, Data from Source 2 will save to Source2DtoInputImpl, etc  To add another Input source, please perform the following   * add a new enum value for Input.OutputTransformerType * add a new Singleton class implementing Input Interface * add a new class implementing Builder Interface * add a new instance value and condition to create new instance of the class in OutputTransformerFactory * add a new condition and private method in OutputTransformer |

* 1. Operations

|  |  |  |
| --- | --- | --- |
| public List<OutputDto> transform(List<Input> inputList) | | |
| Parameters | List<Input> inputList | A list of class implement Input interface needed to be transform |
| Return | List<OutputDto> | A list of OutputDto objects contain the transformed input value |
| Purpose | Transform a list of class implement Input interface and return a list of OutputDto objects for other part of the system to use  By reading CompletableFuture from OutputTransformer, all request is running in asynchronously | |

1. **Builder**
   1. Details

|  |  |
| --- | --- |
| Package | com.toogood.data.transform.builder |
| Interface Name | Builder |
| Purpose | In order to use the Builder design pattern, an interface containing all the steps that the implemented class need to implement |

* 1. Operations

|  |  |  |
| --- | --- | --- |
| void setInputOutput(Input input, OutputDto output) | | |
| Parameters | Input input | A class implemented the Input interface to contain the input value |
| Parameters | OutputDto output | A class contain an object for the builder class to update and return with transformed data |
| Purpose | The process to set up the input class and the output class | |

|  |  |
| --- | --- |
| void buildAccountCode() | |
| Purpose | Build the value of Account Code for the OutputDto object |

|  |  |
| --- | --- |
| void buildName() | |
| Purpose | Build the value of Name for the OutputDto object |

|  |  |
| --- | --- |
| void buildType() | |
| Purpose | Build the value of Type for the OutputDto object |

|  |  |
| --- | --- |
| void buildOpenDate() | |
| Purpose | Build the value of Open Date for the OutputDto object |

|  |  |
| --- | --- |
| void buildCurrency() | |
| Purpose | Build the value of Currency for the OutputDto object |

|  |  |
| --- | --- |
| void resetInputOutput() | |
| Purpose | Reset the value of Input object and Output object in the implemented class  As the implemented class is a Singleton class, the object need to be remove after the process so it would not affect the result for the next process |

|  |  |  |
| --- | --- | --- |
| OutputDto getOutput() | | |
| Return | OutputDto | a class contain an object with the transformed data from input value |
| Purpose | Return the transformed class | |

1. **Source1OutputBuilderImpl**
   1. Details

|  |  |
| --- | --- |
| Package | com.toogood.data.transform.builder |
| Class Name | Source1OutputBuilderImpl |
| Implements | Builder |
| Purpose | Implementing Builder interface for Builder design pattern  Using Singleton design pattern as only a single object is needed  All method in Builder interface must be implemented for transform input from Source 1 to Output |

* 1. Attributes

|  |  |
| --- | --- |
| private Source1DtoInputImpl input | Input class containing input from Source 1 |
| private OutputDto output | OutputDto class will contain all transformed input value from Input class |
| private static Source1OutputBuilderImpl builder | Private static value for Singleton design pattern |

* 1. Operations

|  |  |
| --- | --- |
| private Source1OutputBuilderImpl() | |
| Purpose | Private constructor for Singleton design pattern |

|  |  |
| --- | --- |
| public static Source1OutputBuilderImpl createBuilder() | |
| Purpose | Create a single copy of the class for Singleton design pattern |

|  |  |  |
| --- | --- | --- |
| public void setInputOutput(Input input, OutputDto output) | | |
| Parameters | Input input | A class implemented the Input interface to contain the input value |
| Parameters | OutputDto output | A class contain an object for the builder class to update and return with transformed data |
| Purpose | Set the value for Input and OutputDto class  Check the instance of the Input class to make sure it is the Input class from Source 1 can be handle by this class | |

|  |  |
| --- | --- |
| public void buildAccountCode() | |
| Purpose | Transform Identifier, input value from Source 1, to OutputDto AccountCode  Split the value of Identifier by '|', and use the right side of the value as AccountCode |

|  |  |
| --- | --- |
| public void buildName() | |
| Purpose | Transform Name, input value from Source 1, to OutputDto Name |

|  |  |
| --- | --- |
| public void buildType() | |
| Purpose | Transform Type, input value from Source 1, to OutputDto Type  Use the input id to find out the value from enum Type in Constants class |

|  |  |
| --- | --- |
| public void buildOpenDate() | |
| Purpose | Transform Opened, input value from Source 1, to OutputDto OpenDate |

|  |  |
| --- | --- |
| public void buildCurrency() | |
| Purpose | Transform Currency, input value from Source 1, to OutputDto Currency  Use the input currency to find out the value from enum Currency in Constants class |

|  |  |
| --- | --- |
| public void resetInputOutput() | |
| Purpose | Reset the value of Inputlist and Output in the implemented class  As the implemented class is a Singleton class, the object need to be remove after the process so it would not affect the result for the next process |

|  |  |  |
| --- | --- | --- |
| public OutputDto getOutput() | | |
| Return | OutputDto | a class contain an object with the transformed data from input value |
| Purpose | Return the transformed OutputDto class | |

1. **Source2OutputBuilderImpl**
   1. Details

|  |  |
| --- | --- |
| Package | com.toogood.data.transform.builder |
| Class Name | Source2OutputBuilderImpl |
| Implements | Builder |
| Purpose | Implementing Builder interface for Builder design pattern  Using Singleton design pattern as only a single object is needed  All method in Builder interface must be implemented for transform input from Source 2 to Output |

* 1. Attributes

|  |  |
| --- | --- |
| private Source2DtoInputImpl input | Input class containing input from Source 2 |
| private OutputDto output | OutputDto class will contain all transformed input value from Input class |
| private static Source2OutputBuilderImpl builder | Private static value for Singleton design pattern |

* 1. Operations

|  |  |
| --- | --- |
| private Source2OutputBuilderImpl() | |
| Purpose | Private constructor for Singleton design pattern |

|  |  |
| --- | --- |
| public static Source2OutputBuilderImpl createBuilder() | |
| Purpose | Create a single copy of the class for Singleton design pattern |

|  |  |  |
| --- | --- | --- |
| public void setInputOutput(Input input, OutputDto output) | | |
| Parameters | Input input | A class implemented the Input interface to contain the input value |
| Parameters | OutputDto output | A class contain an object for the builder class to update and return with transformed data |
| Purpose | Set the value for Input and OutputDto class  Check the instance of the Input class to make sure it is the Input class from Source 2 can be handle by this class | |

|  |  |
| --- | --- |
| public void buildAccountCode() | |
| Purpose | Transform CustodianCode, input value from Source 2, to OutputDto AccountCode |

|  |  |
| --- | --- |
| public void buildName() | |
| Purpose | Transform Name, input value from Source 2, to OutputDto Name |

|  |  |
| --- | --- |
| public void buildType() | |
| Purpose | Transform Type, input value from Source 2, to OutputDto Type |

|  |  |
| --- | --- |
| public void buildOpenDate() | |
| Purpose | Not need to implemented because Source 2 input didn't contain open date |

|  |  |
| --- | --- |
| void buildCurrency() | |
| Purpose | Transform Currency, input value from Source 2, to OutputDto Currency  Use the input currency to find out the value from enum Currency in Constants class |

|  |  |
| --- | --- |
| void resetInputOutput() | |
| Purpose | Reset the value of Input object and Output object in the implemented class  As the implemented class is a Singleton class, the object need to be remove after the process so it would not affect the result for the next process |

|  |  |  |
| --- | --- | --- |
| OutputDto getOutput() | | |
| Return | OutputDto | A class contain an object with the transformed data from input value |
| Purpose | Return the transformed OutputDto class | |

1. **Input**
   1. Details

|  |  |
| --- | --- |
| Package | com.toogood.data.transform.Input |
| Interface Name | Input |
| Purpose | Interface all input class from different source needed to implement  All implemented class needed to return a type of source in the getSource method |

* 1. Operations

|  |  |  |
| --- | --- | --- |
| public OutputTransformerType getSource() | | |
| Return | OutputTransformerType | A enum Type of the source |
| Purpose | Method to return the source of the implemented class | |

1. **Source1DtoInputImpl**
   1. Details

|  |  |
| --- | --- |
| Package | com.toogood.data.transform.input |
| Class Name | Source1DtoInputImpl |
| Implements | Input |
| Purpose | Class implemented Input interface representing the input from Source 1 |

* 1. Attributes

|  |  |
| --- | --- |
| **Attribute** | **Validation** |
| private String identifier | Cannot be empty  Must contain “|” |
| private String name | Cannot be empty |
| private String type | Cannot be empty  Must be a valid type |
| private String opened |  |
| private String currency | Cannot be empty  Must be a valid currency |

* 1. Operations
* Getter of the attributes
* Setter of the attributes

|  |  |  |
| --- | --- | --- |
| public OutputTransformerType getSource() | | |
| Return | OutputTransformerType | A enum Type of the source |
| Purpose | Return the enum value to identify where Source 1 is the source of input for this class | |

1. **Source2DtoInputImpl**
   1. Details

|  |  |
| --- | --- |
| Package | com.toogood.data.transform.input |
| Class Name | Source2DtoInputImpl |
| Implements | Input |
| Purpose | Class implemented Input interface representing the input from Source 2 |

* 1. Attributes

|  |  |
| --- | --- |
| **Attribute** | **Validation** |
| private String name | Cannot be empty |
| private String type | Cannot be empty |
| private String currency | Cannot be empty  Must be a valid currency |
| private String custodianCode | Cannot be empty |

* 1. Operations
* Getter of the attributes
* Setter of the attributes

|  |  |  |
| --- | --- | --- |
| public OutputTransformerType getSource() | | |
| Return | OutputTransformerType | A enum Type of the source |
| Purpose | Return the enum value to identify where Source 2 is the source of input for this class | |

1. **OutputDto**
   1. Details

|  |  |
| --- | --- |
| Package | com.toogood.data.transform.output |
| Class Name | OutputDto |
| Purpose | OutputDto contain all generic value that would be used by other part of the system  All input classes needed to transform their data to the data in this class |

* 1. Attributes

|  |  |
| --- | --- |
| private String accountCode | Value for output |
| private String name | Value for output |
| private String type | Value for output |
| private String openDate | Value for output |
| private String currency | Value for output |
| private ValidateResults | Contain results after validation |

* 1. Operations
* Getter of the attributes
* Setter of the attributes

|  |  |  |
| --- | --- | --- |
| public String toString() | | |
| Return | String | A string contain the output of toString() |
| Purpose | Override the toString() method to display output | |

1. **OutputTransformer**
   1. Details

|  |  |
| --- | --- |
| Package | com.toogood.data.transform.transformer |
| Class Name | OutputTransformer |
| Purpose | This is the class to specify how to use the Builder to transform the input data to output |

* 1. Operations

|  |  |  |
| --- | --- | --- |
| public static CompletableFuture<Optional<OutputDto>> buildOutput(Input input) | | |
| Parameters | Input input | A class implemented the Input interface to contain the input value |
| Return | CompletableFuture<Optional<OutputDto>> | An Optional class contain an OutputDto object for the builder class to update and return with transformed data |
| Purpose | Main method to transform the class implemented the Input interface to OutputDto based on the builder returned from OutputTransformerFactory | |

|  |  |  |
| --- | --- | --- |
| private static Optional<OutputDto> buildSource1Output(Input input) | | |
| Parameters | Input input | A class implemented the Input interface to contain the input value |
| Return | Optional<OutputDto> | An Optional class contain an OutputDto object for the builder class to update and return with transformed data |
| Purpose | Using the Singleton builder for Source 1 input to build the output  Use synchronized to make sure it is a thread-safe method as static object will be used | |

|  |  |  |
| --- | --- | --- |
| private static Optional<OutputDto> buildSource2Output(Input input) | | |
| Parameters | Input input | A class implemented the Input interface to contain the input value |
| Return | Optional<OutputDto> | An Optional class contain an OutputDto object for the builder class to update and return with transformed data |
| Purpose | Using the Singleton builder for Source 2 input to build the output  Use synchronized to make sure it is a thread-safe method as static object will be used | |

|  |  |  |
| --- | --- | --- |
| private static Optional<OutputDto> buildGenericOutput(Input input, Builder builder) | | |
| Parameters | Input input | A class implemented the Input interface to contain the input value |
| Parameters | Builder builder | A builder class for specific source |
| Return | Optional<OutputDto> | An Optional class contain an OutputDto object for the builder class to update and return with transformed data |
| Purpose | Call Validator to validate the Input object.  If the input object is valid, perform a generic way to transform input to ouput regardless which builder object is in used  If the input object is invalid, add the error message and the invalid input object to the output  Return the output object | |

1. **OutputTransformerFactory**
   1. Details

|  |  |
| --- | --- |
| Package | com.toogood.data.transform.transformer |
| Class Name | OutputTransformerFactory |
| Purpose | Use Factory design pattern to return a builder class based on the enum Type in the Input Interface |

* 1. Attributes

|  |  |
| --- | --- |
| private static Builder source1OutputBuilderImpl | Builder implemented class for Source 1 |
| private static Builder source2OutputBuilderImpl | Builder implemented class for Source 2 |

* 1. Operations

|  |  |  |
| --- | --- | --- |
| public static Builder retrieve(OutputTransformerType type) | | |
| Parameters | OutputTransformerType type | A type of source input from enum OutputTransformerType in constants |
| Return | Builder | A builder object based on the source input |
| Purpose | Return a class implemented Builder Interface based on the enum Type in the Input Interface | |

1. **Constants**
   1. Details

|  |  |
| --- | --- |
| Package | com.toogood.data.transform.util |
| Class Name | Constants |
| Purpose | To store constant value |

* 1. Attributes

|  |  |
| --- | --- |
| public static enum OutputTransformerType | Enum containing all different kind of input source |
| public static enum Type | Enum type for identify the Output Type value from the input value |
| public static enum Currency | Enum type for identify the Output Currency value from the input value |

1. **ValidateResults**
   1. Details

|  |  |
| --- | --- |
| Package | com.toogood.data.transform.validation |
| Class Name | ValidateResults |
| Purpose | To store validation results |

* 1. Attributes

|  |  |
| --- | --- |
| private boolean valid | Contains true if validation pass, otherwise contain false |
| private String errorMessage | Contain all error message after validation if any |
| private Input inValidInput | Contain the invalid Input object |

1. **CurrencyConstraint**
   1. Details

|  |  |
| --- | --- |
| Package | com.toogood.data.transform.validation.constraint |
| Interface Name | CurrencyConstraint |
| Purpose | Interface for CurrencyValidator |

1. **IdentifierConstraint**
   1. Details

|  |  |
| --- | --- |
| Package | com.toogood.data.transform.validation.constraint |
| Interface Name | CurrencyConstraint |
| Purpose | Interface for IdentifierValidator |

1. **TypeConstraint**
   1. Details

|  |  |
| --- | --- |
| Package | com.toogood.data.transform.validation.constraint |
| Interface Name | CurrencyConstraint |
| Purpose | Interface for TypeValidator |

1. **CurrencyValidator**
   1. Details

|  |  |
| --- | --- |
| Package | com.toogood.data.transform.validation.validator |
| Class Name | CurrencyValidator |
| Implements | ConstraintValidator<CurrencyConstraint, String> |
| Purpose | A Validator for any input filed having @CurrencyConstraint |

* 1. Operations

|  |  |  |
| --- | --- | --- |
| public boolean isValid(String currency, ConstraintValidatorContext cxt) | | |
| Parameters | String currency | Field value |
| Parameters | ConstraintValidatorContext cxt | Field context |
| Return | boolean | Indicator for valid or invalid field |
| Purpose | Validate is the inputted currency are the valid currency in Constants.Currency | |

1. **IdentifierValidator**
   1. Details

|  |  |
| --- | --- |
| Package | com.toogood.data.transform.validation.validator |
| Class Name | IdentifierValidator |
| Implements | ConstraintValidator<IdentifierConstraint, String> |
| Purpose | A Validator for any input filed having @IdentifierConstraint |

* 1. Operations

|  |  |  |
| --- | --- | --- |
| public boolean isValid(String identifier, ConstraintValidatorContext cxt) | | |
| Parameters | String identifier | Field value |
| Parameters | ConstraintValidatorContext cxt | Field context |
| Return | boolean | Indicator for valid or invalid field |
| Purpose | Validate is the inputted identifier having the “|” symbol | |

1. **TypeValidator**
   1. Details

|  |  |
| --- | --- |
| Package | com.toogood.data.transform.validation.validator |
| Class Name | TypeValidator |
| Implements | ConstraintValidator<TypeConstraint, String> |
| Purpose | A Validator for any input filed having @TypeConstraint |

* 1. Operations

|  |  |  |
| --- | --- | --- |
| public boolean isValid(String type, ConstraintValidatorContext cxt) | | |
| Parameters | String type | Field value |
| Parameters | ConstraintValidatorContext cxt | Field context |
| Return | boolean | Indicator for valid or invalid field |
| Purpose | Validate is the inputted type are the valid currency in Constants.Type | |