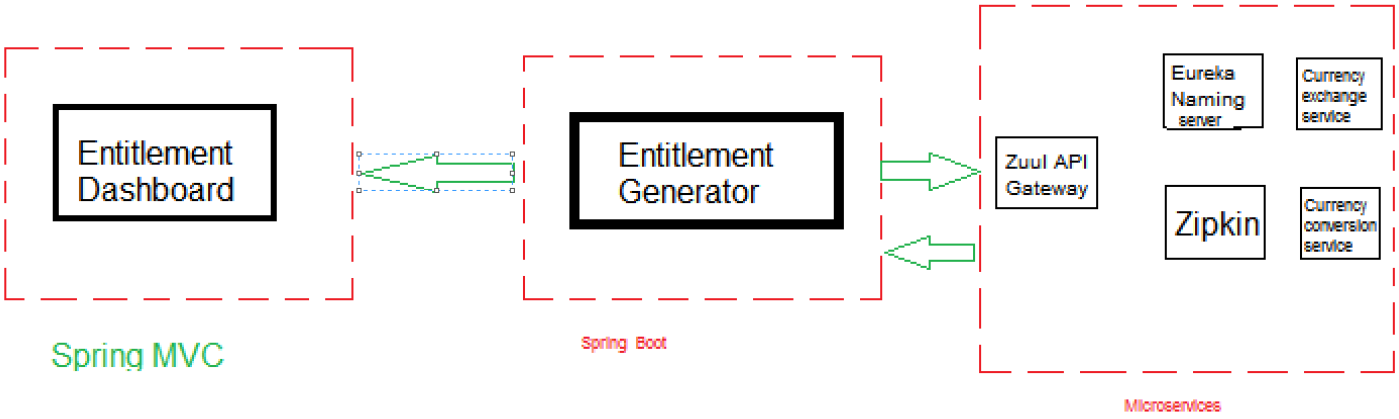
Overall Architecture



Entitlement Generation Service

As part of corporate action, Companies distributes rewards depending on their earnings to a particular class of its shareholder [1]. Cash dividend are paid through the bank and other financial institutes to customer in which they hold their portfolio. Banks are notified of these payments through swift messages. Banks record these payments in the table called as Diary entry. Some of the examples of corporate actions are dividend payment, stock split, merger and spin off and more.

In this application, corporate action pertaining to cash dividend payment is implemented. Entitlement generation service generates the entitlement records for the customer holding shares for which diary event is recorded.

Tables involved:

1. **Diary**: Records the company which has send the dividend, pay per unit of share, pay day and currency in which company holds the account in that particular bank or in which they wish to pay. For example, ONGC decides to pay dividend of $2 per share, it will send a swift message to bank(or financial institute) to pay all shareholders a amount of $2 per share to customer holding ONGS's share. This information is recorded in Diary table.
2. **Portfolio Work File**: It is an associative entity which holds the links between portfolio and share it holds.

3. **Entitlement**: Each record in this table represents a portfolio to which received dividend is to be paid. It also holds the total amount (dividend to be paid per share * number of shares in portfolio). For example, bank has 10 customers who have ONGC shares in their portfolio. Then 10 entitlement records will be created in entitlement tables.

Application utilizes Spring Batch which fetches diary from database(N), portfolios holding shares of that company(f(N)) and generates entitlement (N*f(N)). Within the process of generation, application checks whether the currency in which diary event is recorded is same as that of currency of account to which dividend payment is to be made. If diary payment currency and account currency is not same then application calls currency conversion microservice which returns the converted total amount.

Batch performs following steps:

1. Fetching diary records from database:

```
23 • select * from diary;
```

Result Grid | Filter Rows: | Edit: | Export/Import:

diary_no	event_name	amount	company_name	currency	status	pay_date
2	dividend	2	Infosys	USD	NSNG	2019-02-19 00:00:00
3	dividend	40	TechM	EUR	NSNG	2019-02-19 00:00:00
4	dividend	15	SalesForce	USD	NSNG	2019-02-19 00:00:00
5	dividend	4	Hansa	EUR	NSNG	2019-02-02 00:00:00
6	dividend	29	Cognizant	USD	NSNG	2019-02-19 00:00:00

2. Processing the data involves loading all portfolios holding companies share for which diary is recevied. After fetching the portfolios, next task currency conversion is performed if currency in which dividend is to be paid is different from customer's account currency. For this purpose, currency conversion microservice running on port 8765 is consumed.

```
63 • select * from portfolioworkfile;
64 • DROP TABLE portfolioworkfile;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap

portfolio_wf_id	customer_id	account_id	account_currency	diary_currency	units	company
1001-100-1	1001	100	INR	NULL	20.5	Infosys
1001-102-2	1001	102	EUR	NULL	1.5	Cognizant
1002-100-1	1002	100	INR	NULL	205	TechM
1002-100-2	1002	100	EUR	NULL	10.5	Cognizant
1003-103-1	1003	103	CHN	NULL	20.5	SalesForce
1004-100-1	1004	100	INR	NULL	203	TechM
1005-103-1	1005	103	INR	NULL	2.5	Infosys
1006-100-1	1006	100	INR	NULL	40.5	TechM
1007-103-1	1007	100	INR	NULL	20.5	Infosys
1008-102-1	1008	102	EUR	NULL	60.5	Infosys
1009-101-1	1009	101	YEN	NULL	7.5	Hansa
1010-101-1	1010	100	YEN	NULL	20.4	Hansa
1010-103-1	1010	100	CHN	NULL	5.5	SalesForce
NULL	NULL	NULL	NULL	NULL	NULL	NULL

portfolioworkfile 36 ×

Note:

Microservice results:

localhost:8100/currency-converter/from/USD/to/INR/quantity/1000

Apps | Gmail | YT | Others | TAMU | MF | Bank | linux | Spring MVC hello w... | Intern | User Stories and Us...

{"id":10017,"cfrom":"USD","cto":"INR","conversionMultiple":70.00,"quantity":1000,"totalCalculatedAmount":70000.00,"port":0}

Yellow being input and Blue circle being output.

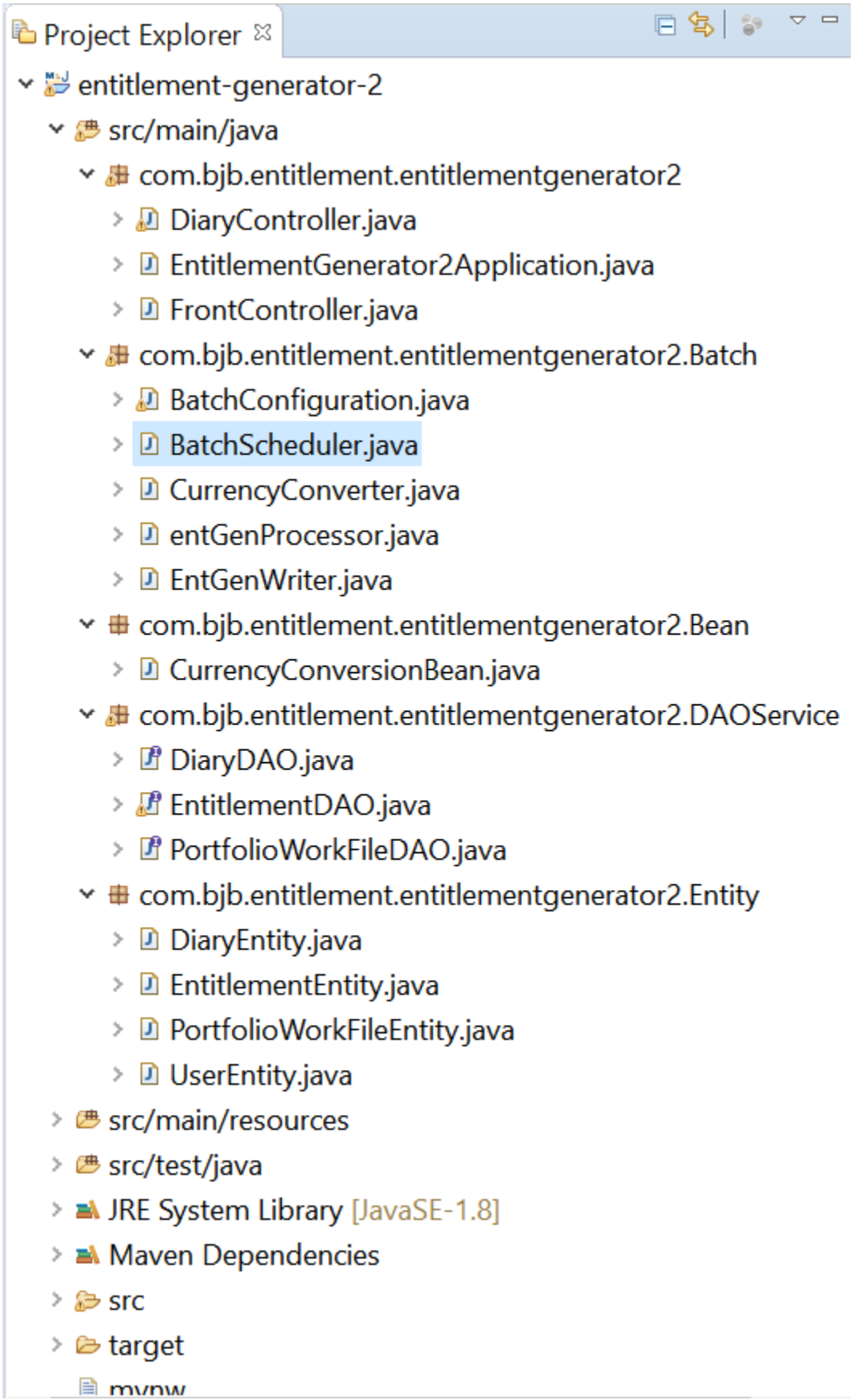
3. Item writer is a customized class which writes data received from batch processor.

```
82 •  select * from entitlement;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

entitlement_id	diary_id	customer_id	account_id	portfolio_id	pay_currency	account_currency	units	per_unit_amount	total_div
470	2	1001	100	1001-100-1	USD	INR	20.5	2	2870
471	2	1005	103	1005-103-1	USD	INR	2.5	2	350
472	2	1007	100	1007-103-1	USD	INR	20.5	2	2870
473	2	1008	102	1008-102-1	USD	EUR	60.5	2	1815
474	3	1002	100	1002-100-1	EUR	INR	205	40	713400
475	3	1004	100	1004-100-1	EUR	INR	203	40	706440
476	3	1006	100	1006-100-1	EUR	INR	40.5	40	140940
477	4	1003	103	1003-103-1	USD	CHN	20.5	15	1845
478	4	1010	100	1010-103-1	USD	CHN	5.5	15	495
479	5	1009	101	1009-101-1	EUR	YEN	7.5	4	3750
480	5	1010	100	1010-101-1	EUR	YEN	20.4	4	10200
481	6	1001	102	1001-102-2	USD	EUR	1.5	29	652.5
482	6	1002	100	1002-100-2	USD	EUR	10.5	29	4567.5

Application structure:



References:

1. https://www.investopedia.com/terms/d/dividend.asp