ePayments Solution

Design description

The ePayments design is a simple design which is built around three entities. These are Account, Person and Transaction.

Account is simple as it contains an account number, an owner (Person), balance and a list of transactions on the account. Person exists of a name and last name. Transaction is used to record the transfer of an amount, transfer type and a date. The transaction is used as part of testing to show how each transfer was actioned internally.

I choose REST as I feel it is lightweight and easy to consume by consumers. The REST layer is mapped around the account resource and its behaviour.

I choose to use a HashMap internally to store the accounts to simplify the solution. This can be swapped out to use a database instead.

Technologies

I choose OpenLiberty as it is a MicroProfile implementation and the requirements was small enough not to warrant a full JEE implementation. MicroProfile can also be configured to generate a minified Uber-jar (an executable jar) which can be deployed and run using a simple java-jar *.jar command. All the technologies that I needed was available in the implementation without introducing any external dependencies.

Limitations

The application does not currently use and external database which means that the data is kept in memory and disappears once the application is restarted. If this was to be deployed in a docker environment or a Kubernetes environment the limitation of each replica having it's own JVM will take a toll on the amount of replicas that could be started up.

Deployment instructions

Currently the application produces a war file which could be deployed into an OpenLiberty environment. If one made minor adjustments to the pom file it could generate a minified jar file which could be run using java -jar ePayments.jar