Billingware

# Introduction

This project aims to provide a distributed, highly scalable (open source) service that provides a developer/user intuitive tool to perform credit/debit actions on an entity.

## Features

1. An REST-like HTTP API interface that performs basic banking functions such as credit and debit an account
2. A read-only web user interface to view accounts, balances and transaction history.
3. An SDK with which to interact with the API service (beginning with C#)

## Features Breakdown

1. REST-like HTTP Interface: All requests and responses are in JSON. Are requests will have Authorization with Basic scheme. The keys will be provided by the system upon successful setup. Each request will have to contain a “reference” parameter supplied by the caller. Each response will contain the said “reference” and a “ticket” (an encrypted token that validates the transaction as received and processed). Below are the functions in the HTTP interface
   1. Create Account
      1. Supports GET, POST, PUT and DELETE
      2. The following will be used to create the account:
         1. Account Number – from caller’s legacy system, must be unique and may be used in all subsequent API calls
         2. Account Code – system generated and may be used in all subsequent API calls
         3. Extra – Flat JSON field
            1. Will be used for generic manipulations, e.g. allow overdraft, name, address, etc.
   2. Credit
      1. Supports POST
      2. Asynchronous – callback may be provided
      3. Payload includes:
         1. Amount
         2. Reference
         3. Narration
   3. Debit
      1. Supports POST
      2. Asynchronous – callback may be provided
      3. Payload includes:
         1. Amount
         2. Reference
         3. Narration
   4. Get Balance
      1. Supports GET
      2. Payload includes:
         1. Account Code or account number
2. Web UI: This will be a basic, simple read-only interface to view all accounts and transaction history. Access to this portal will use a flat file.
3. SDK: The following languages
   1. C#