From:	Clive Vermeulen
To:	Consumer Data Rights Data61
Cc:	David Grahame
Subject:	ConsumerDataStandardsAustralia feedback to decision proposal 39 - Draft Standards
Date:	Monday, 26 November 2018 3:29:00 PM

Hi there, Clive here from Xero Ltd. Apologies for missing the feedback window for Decision Proposal 39, had I responded in time I would have used my @clivevermeulen-xero Github profile. Could you please include the content of this email as a response to that discussion thread, thanks!

As others have pointed out, it's really great to see this process being conducted in such a public and transparent fashion. Our feedback to the proposed standards is generally reflective of comments made by several commenters on this thread as well as those specific to the Accounts and Transactions payloads:

- TransactionId property
    - The matter of mandatory unique and immutable transactionId's received a fair amount of discussion in the Decision Proposal 28 thread - it's unclear from the current standard whether this field is mandatory or conditional, i.e. when looking at the online documentation for the TransactionBasic payload (https://consumerdatastandardsaustralia.github.io/standards/#schematransactionbasic) the field is currently marked as not required while the Description field suggests the field is conditional (i.e. *mandatory unless…*). However, when referring to the PDF documenting outcomes of Decision Proposal 28 transactionId is listed as a Mandatory field.
    - This ambiguity is concerning because of the important part that transactionId's play in the de-duplication of transactions. (jh-a and others have also made a strong case for the importance of transactionId's in the Decision Proposal 28 discussion)
    - The matter of de-duplication of transactions is really relevant because API consumers don't necessarily know when transactions transition from hold to posted, whether the hold transaction is updated "in-place" to posted, and how far back in time posted transactions can be added/updated in the transaction history. Would it only be necessary to query as far back as the beginning of the previous day, or perhaps we may need to query a longer span of time like the 3 to 5 days considered by BrianParkerAU in his previous comment?
    - Another point of possible confusion, the specification refers to ordering of transaction data by effectiveDate, suggesting that the ordering of transactions per customer bank account may change as they switch from a hold to a posted status, further complicating matters.
    - Also, the Description suggests the option of creating a unique identifier by hashing a combination of fields, but if this hash is limited to those fields already available through the API then a unique value may not be achieved.

Relationship between Account balances and Transactions

  - In Xero we package collections of transactions into statements with a closing balance - this is important for our customers' confidence that their bank balance is consistent with their accounting balance and we imagine other API consumers would want to achieve similar outcomes. The separation of Transactions and Account balances from an API perspective does make sense, but since we’re retrieving this data from multiple endpoints there is a risk of race conditions or that the underlying data or caches could be out of sync.

  - As ANZ AU noted in their recent feedback, the inclusion of a timestamp or hash would be useful for API consumers to know that the latest retrieved account balances and transactions are in deed in sync.

  - Beyond this point though, we’ve heard mention that account balances will not to be updated intra-day initially. I haven’t found confirmation of this point in these discussion threads, so possibly I’m mistaken. If that is the case though, this would raise a number of questions around when the balance is calculated, e.g. would it be at midnight at the end of the previous day, or the time when daily batch processing for that bank runs, etc?

  - As an example of challenges API consumers could face: if the balance values are provided as at midnight at the end of the previous day, that would mean that if you wanted a list of transactions including an accurate end balance you would need to limit yourself to using the transactions up until and including the end of the previous day.

- Additional calls to access TransactionDetails

  - This matter has been the subject of a lot of really good discussion and proposed solutions. We also feel that requiring API consumers to perform additional calls per transaction line is bad for API consumers and for the banks providing those API’s. We’re supportive of solutions like the “includeDetail” parameter.

- Account number masking

  - We’d also be in favour of adjusting the format of MaskedAccountNumber from Last 3 digits unmasked to Last 4 digits unmasked.

  - This would aid customers in identifying their accounts correctly, particularly given that a nickName is not always supplied by customers, in which case the account displayName is presumably based on the product name and so may not be unique.

- Transaction Codes
    - We agree that the inclusion of Transaction Codes is preferable and that adoption of ISO 20022 would be ideal, but we'd be satisfied with the UK Open Banking compromise of allowing PriorietyBankTransactionCode in lieu of banks being able to complete the ISO 20022 mapping for initial go-live.

cheers,
Clive

--

Clive Vermeulen
Product Architect
19-23 Taranaki St, Wellington, 6011
www.xero.com