

## AMP Bank

AMP's new Banking division, AMP Bank, is a digital bank offer built on the "Engine By Starling" (Engine) SaaS platform.

## The Challenge

We'd like you to develop a "round-up" feature for AMP Bank customers using Engine's public developer API.

For a customer, take all the transactions in a given week and round them up to the nearest dollar. For example with spending of \$4.35, \$5.20 and \$0.87, the round-up would be \$1.58. This amount should then be transferred into a savings goal, helping the customer save for future adventures.

Note: You should use the default currency of the account for creation of the savings goal.

### Time:

We like you to submit your code, complete or otherwise within 2 days.

### Submitting:

Provide a link to a public github repo containing the project.

If you have any questions, please ask your AMP recruitment partner

## API Calls

To make this work, the key parts from our public [API](#) you will need are:

1. Accounts - To retrieve accounts for the customer
2. Transaction feed - To retrieve transactions for the customer
3. Savings Goals - Create a savings goals and transfer money to savings goals

We do expect to see your working here: please do not use any of the libraries out there which provide an sdk for interacting with our api.

## What to build

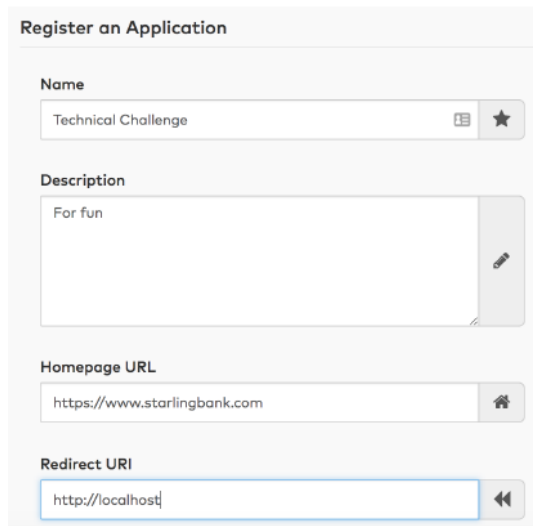
Pick your favourite platform to develop for (Android / iOS). If you have experience with multiple languages that's fantastic - you only need to submit for one and your choice won't impact potential roles with us.

- **Android:** Simple application in Java or Kotlin which displays the amount to round up for a week with a button to perform the transfer.
- **iOS:** Simple iOS application written in Swift and using UIKit. It should display the amount to round up for a week of transactions with a way to perform the transfer into a saving goal.

## Getting Started

Don't worry about reading the top half of the API documentation - *we aren't asking you to build out an OAuth implementation!* To jump start to the point of being able to make API calls:

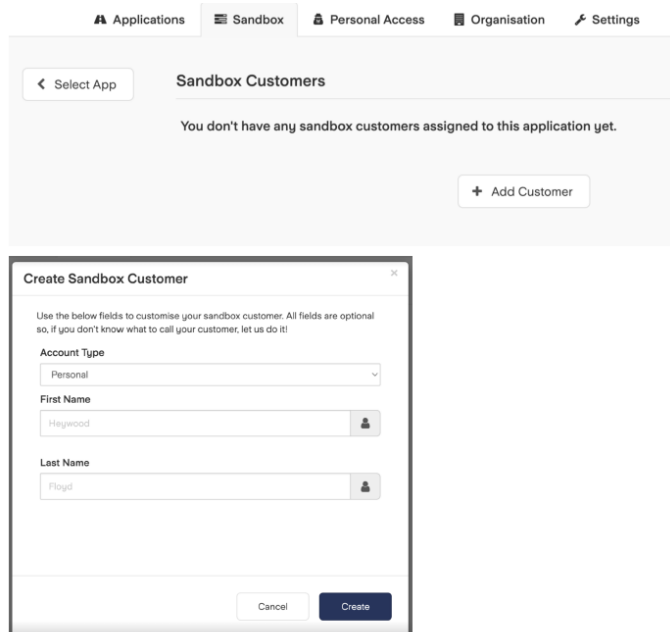
1. [Sign-up](#) for a Starling developer account and verify / secure your account.
2. [Create an API application](#) (you can use any URLs as they won't be used for local development).  
*Applications are how we track API keys and usage.*



The 'Register an Application' form contains the following fields:

- Name:** Technical Challenge
- Description:** For fun
- Homepage URL:** https://www.starlingbank.com
- Redirect URI:** http://localhost

3. [Create a sandbox customer.](#)



The 'Sandbox Customers' interface shows a message: "You don't have any sandbox customers assigned to this application yet." with an "Add Customer" button.

The 'Create Sandbox Customer' modal includes the following fields:

- Account Type:** Personal
- First Name:** Heywood
- Last Name:** Floyd

Buttons: Cancel, Create

4. Copy the **access token** for the customer

**Sandbox Customer**

Use the simulator to experiment with the API by simulating transactions and seeing the results.

**Customer Details**

Name	River Tam
Access	Tier 5
Customer ID	1f3d0beb-b12d-46e5-a780-7e5ec7d2ae38 <input type="button" value="Copy"/>
Access Token	y5q3uDBWWhEXznpKbC <input type="button" value="Copy"/>
Refresh Token	Copied! dw4qrjdhkubhPTc <input type="button" value="Copy"/>

Token expires in 1440 minutes at 23:26 21/02/18

- Click the auto-simulate button for the customer. *This makes about 30 transactions on the customer account to give a reasonable history to play with.*

**Auto-simulator**

Auto-magically populate your customer's history with a bunch of different transactions.

That's it! You can now make API calls for the customer with your favourite HTTP client.

**Base URL for resources:** `https://api-sandbox.starlingbank.com`

**Required headers:** Accept: application/json

Authorization: {yourAccessTokenFromAbove}

User-Agent: Your Name

*Example with curl:*

```
curl --get https://api-sandbox.starlingbank.com/api/v2/accounts -H "Accept: application/json" -H "Authorization: Bearer {yourAccessToken}"
```