August 2016

Receipt Keeper – Implementing Restful API using Loopback framework

# introduction

Receipt Keeper will allow users to track how much money they spend in certain categories over a specific number of months, and start a group which others can be invited to join. This will allow group admins to see reports for the group. Receipt Keeper is built to work on Android devices and also includes a web app. The mobile application will make it easier for people to manage categories, tags and reports while having the convenience of scanning receipts wherever they are. The web application provides the list of receipts and shows graphs to visualize users’ spending.

## Challenges and objectives

### OCR engine to recognize numbers and characters on user’s receipt

### Providing Rest API for scalability

### Android Application using Rest API to communicate with Server

# methodologies

## Development Methodologies & Team Rules

### Agile Unified Process: every 2weeks

### Coding Rule

## Source Control & Change Management

### Three Github projects are created for managing document, web server & app, android

### Direct commit was sublated and encouraged Pull request for peer review

### Github Wiki: for sharing information & minutes

## Issue Tracking

### Github’s issue feature

## Backend as a Service(BAAS)

### LoopBack

## Deployment

### Heroku.com

#### Setup pipeline: Pull request(Github) -> Review app -> Staging -> Production

#### Buildpacks is used for deploying loopback app

## Data as a Service (DAAS)

### Mongolab

## Sharing Issues and Communication

### Slack channel

# architecture Decisions

## Architectural Driver

### Quality Attribute:

#### Scalability: Extensible to other kind of client

#### Flexibility: Easy to change API

#### Performance: Quick to search

#### Security:

## Architectural Decisions

### Rest API

### No SQL for the speed of search

### Security: CORS middle

# technologY

## Tesseract ; open source engine

## Android-ocr is a mobile version of the Tesseract project which is a well-known open source project for OCR with Open CV which is for computer vision. The web client application will be developed using Angular framework, which is useful for development of single page web apps with the MVC pattern. Additionally, javascript libraries for uploading image files and visualizing spending with graphs will be used.

## Loopback

## Various platforms were considered. Due to the team’s skill level, and to minimize time to implement the server, Loopback was chosen. TABLE1 is a comparison of features between Mobile Baas(Back end As a Service) solutions.

## Loopback is an open source Node application framework based on Express.

## It is supported by IBM and StrongLoop(Acquired by IBM in September, 2015)

## It makes it easy to:

## Create a Rest API platform and manage REST API with provided tools, such as creating a new API or applying ACL(access control list)

## Connect mobile devices and browser to data and services

## Set up models and create REST APIs in minutes using StrongArc

## It has a Built-in API explorer to test API

## Easy authentication and authorization setup: Built-in role based access controls. oAuth user and registration models backed in. Can add custom policies using CLI or JSON

## StrongLoop Arc is a graphical tool for building, deploying, managing and monitoring LoopBack applications and APIs

## Getting started is easy and there’s a Yeaoman-based scaffolding tool for generating new project skeletons. Developers have a great choice of database connectors for all major SQL databases and MongoDB. For development you can also use an in-memory db.

## LoopBack helps with that further by providing a handy model generator for building your classes quickly. These can be validated with the built-in validation methods. Entity relationships are very well supported. The security model is complex with user roles, principals and ACLs

## 

Table 1. Comparison of BAAS (Backend As A Service)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **LoopBack** | **Express** | **Hapi** | **Sails** | **Restify** | **Meteor** |
| **Type** | API framework | HTTP server library | HTTP server framework | Web MVC framework | REST HTTP library | Full-stack JavaScript app platform |
| **Top Features** | Enterprise connectivity, API Explorer, generators, client SDKs, websocket microservices | HTTP routing, middleware | Modularity, security | Rails familiarity, MVC | Simplicity, REST routing | Universal JavaScript, reactive rendering, websocket microservices |
| **Suitable For** | Web apps, APIs | Simple web apps | Web apps, APIs | Web apps, APIs | Simple REST APIs | Web apps |
| **Github Stars** | 5k | 19k | 4k | 10k | 3k | 28k |
| **Support** | StrongLoop | StrongLoop | N/A | N/A | N/A | Meteor Development Group |
| **Pure Node runtime** | Yes | Yes | Yes | Yes | Yes | No |
| **Client SDKs** | Angular, Browser, Node.js, iOS, Android, Xamarin | N/A | None | None | None | JavaScript, Cordova for iOS and Android, React, AngularJS |
| **Export API Definition** | Yes | With strong-remoting | None | None | None | With meteor-rest |
| **Tools** | Visual API composer, Explorer, CLI code generators | CLI app generator | Yeoman generator | Yeoman generator | Yeoman generator | CLI tool |
| **Visual API composition** | Yes | No | No | No | No | No |
| **StrongLoop Arc Build & Deploy, Monitoring, Profiling** | Yes | Yes | Yes | Yes | Yes | Yes |
| **Extensions** | Push, File Storage, Passport, OAuth 2.0, Express Middleware | Express / Connect Middleware | Hapi Plugins |  |  | Proprietary package system and repository, npm |
| **Data sources** | In-memory/file, MongoDB, MySQL, Oracle, PostgreSQL, SQL Server, ATG, Email, REST, SOAP | None | None | In-memory, File, PostgreSQL, MySQL, MongoDB | None | MongoDB, MySQL and PostgreSQL via 3rd-party packages |
| **ACLs** | Yes | No | No | No | No | Basic allow/deny |

# 

# Open Sources

## Server

|  |  |
| --- | --- |
| bunyan: ^1.8.1, | Apache 2.0 |
| compression: ^1.0.3, |  |
| cors: ^2.5.2, |  |
| errorhandler: ^1.1.1, |  |
| express-bunyan-logger: ^1.3.0, |  |
| helmet: ^1.3.0, |  |
| loopback: ^2.22.0, |  |
| loopback-boot: ^2.6.5, |  |
| loopback-component-explorer: ^2.4.0, |  |
| loopback-component-storage: ^1.5.0, |  |
| loopback-connector-mongodb: ^1.15.2, |  |
| loopback-datasource-juggler: ^2.39.0, |  |
| loopback-ds-timestamp-mixin: ^3.2.4, |  |
| morgan: ^1.7.0, |  |
| serve-favicon: ^2.0.1 |  |

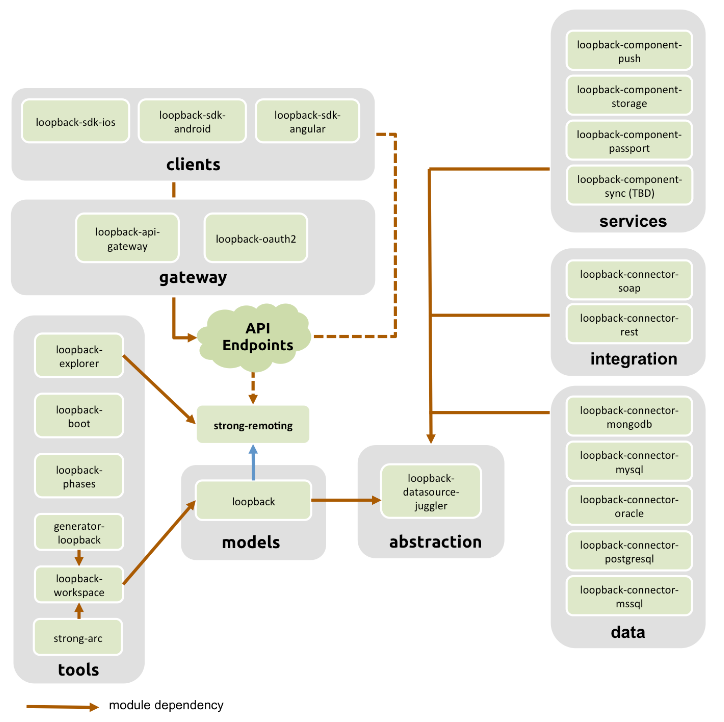
## Web Client

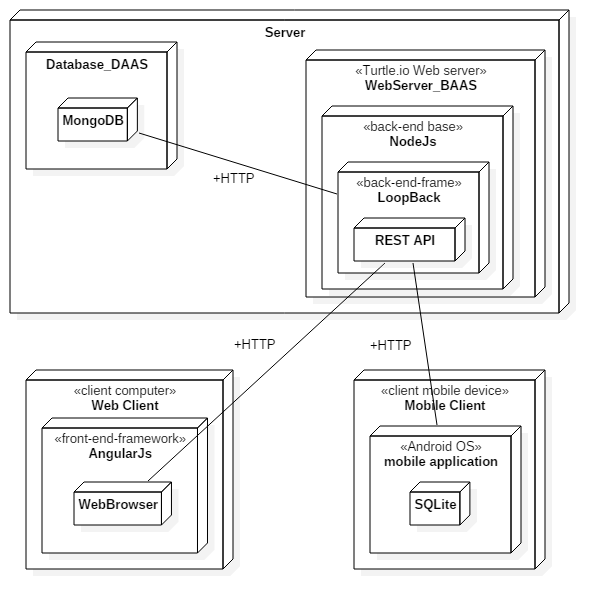
|  |  |
| --- | --- |
| json3: ~3.3.1, |  |
| font-awesome: 4.3.0, |  |
| oclazyload: ~0.5.2, |  |
| angular-loading-bar: ~0.7.0, |  |
| angular-chart.js: ~0.5.2, |  |
| angular-ui-router: ^0.3.0, |  |
| angular-resource: ^1.5.6, |  |
| bootstrap: ^3.3.6, |  |
| eonasdan-bootstrap-datetimepicker: ^4.17.37 |  |

## Android Mobile App

|  |  |
| --- | --- |
| **com.rmtheis:tess-two:5.4.0 com.strongloop:loopback-sdk-android:1.5.+ com.github.bumptech.glide:glide:3.6.0** |  |
| **com.rmtheis:tess-two:5.4.0** Tesserat: https://github.com/tesseract-ocr/tesseract |  |

# APIs





## Application Layer

### Routes

### Middleware

### Errors

### Formatters

## Domain Layer

## 

## Service Layer

### Data Access

### JWT (JSON Web Token)

### Authentication

### ACL(Access Control List)

## Foundation Layer

### Database

### 

# conclusion

By using the Loopback framework, it is easy and quick to implement Rest API for scalability. It is also fast because is based on NodeJs and so it is easy to implement single page applications with the help of AngularJS.

# References

<http://loopback.io/>

android-ocr : <https://github.com/rmtheis/android-ocr>

LoopBack: <https://loopback.io/>

<http://www.erudika.com/blog/2015/10/21/backend-frameworks-usergrid-loopback-para-baasbox-deployd-telepat/>

<http://www.restapitutorial.com/lessons/whatisrest.html>

# Glossary

# A

ACL: Access control list, a list associated with an object that identifies all the subjects that can access the object and their access rights.  See [Authentication, authorization, and permissions](https://docs.strongloop.com/display/LB/Authentication%2C+authorization%2C+and+permissions).

# B

**Buildpacks**

Buildpacks are scripts that are run when the app is deployed. They are used to install dependencies for the app and configure the environment.

.

# C

# D

**data source**

A data source connects with specific database or other back-end system using a connector.

# E

# F

# G

# H

# I

# J

# K

# L

# M

# N

# O

# P

# Q

# R

# S

# T

# U

# V

# W

# X

# Y

# Z