Notes

* Developed using MAMP on iMac. Uses SQLite3 for database.
* Deployment:
  1. Clone the project: git clone into directory of your choice. Notes below refer to that as *projectdir*.
  2. Open MAMP and Stop MAMP servers
  3. Configure MAMP Apache Port to 80.
  4. Configure virtual host:
     + Edit /private/etc/hosts and add this line: 127.0.0.1 parksidelending
     + Edit /Applications/MAMM/conf/apache/httpd.conf and uncomment the line #Include /Applications/MAMP/conf/apache/extra/httpd-ssl.conf.
     + Edit /Applications/MAMP/conf/apache/extra/httpd-vhosts.conf. Add these lines:  
       <VirtualHost \*:80>

DocumentRoot "*projectdir*"

ServerName parksidelending

</VirtualHost>

* + - Edit /Applications/MAMP/conf/apache/extra/httpd-ssl.conf. Search for the lines with DocumentRoot and ServerName and set them the same above.
  1. Install certificate files: the project contains a self-generated certificate. Two options:
     1. Copy files *projectdir*/server.crt and *projectdir*/server.key to /Applications/MAMP/conf/apache *…or…*
     2. Edit /Applications/MAMP/conf/apache/extra/httpd-ssl.conf
        + Change the SSLCertificateFile line to SSLCertificateFile "*projectdir*/server.crt"
        + Change the SSLCertificateKeyFile line to SSLCertificateKeyFile “*projectdir*/server.key”
  2. Start MAMP servers
  3. Using the web site: browse to <https://parksidelending/www>. You’ll likely get a security notice from your browser since the certificate is self-generated. Allow the browser to proceed to the site (steps to proceed depend on your browser).
* Sample calls using curl. The commands below worked for me on one machine. On another, I had to use the --cacert option to specify the certificate file. So you may need to add --cacert *projectdir*/server.crt option to these commands.
  1. Apply for a loan
     + curl -X POST --data "ssn=123-45-6789&propertyValue=300000&loanAmount=100000" <https://parksidelending/v1/loans>
     + outputs: {"result":"success","resultstring":null,"data":{"loanId":44,"propertyValue":"282500","loanAmount":"113000","loanStatus":1,"createdTime":"2016-10-29 02:38:25","lastUpdatedTime":"2016-10-29 02:38:25"}}
  2. Get loan status for loan 1
     + curl <https://parksidelending/v1/loans/1>
     + outputs: {"result":"success","resultstring":null,"data":{"loanId":1,"ssn":"123-456-7890","loanAmount":100000,"propertyValue":300000,"loanStatus":"2","createdTime":"2016-10-28 18:37:41","lastUpdatedTime":"2016-10-28 18:37:41"}}
* *projectdir* structure
  1. Notes.docx: this file
  2. README.md
  3. server.crt: certificate file
  4. server.key: private key to server.crt
  5. /www:
     + index.html: rudimentary UX for demo / testing. Dialog-based. Can apply for a loan and check status of a loan.
     + /css: main.css has styles
     + /js: Loans.js: wraps calls to the REST APIs using ajax.
  6. /v1: the server
     + .htaccess: Has rewrite rules for redirecting “loans” to LoansController.php
     + LoansController.php: front end for receiving REST requests for loans.
     + autoload.php: PHP class auto loader.
     + project.php: defines for project root directory, loan statuses, debug levels, and debug log location.
     + dbglog.php: methods for generating debug trace.
     + /Classes:
       1. Request.php: contains info about the REST API including the method and parameters. Created from front-end controller (such as LoansController.php) and passed to API class (such as Loans.php) where it gets used.
       2. Loans.php: Implementation of the APIs.
     + /Data:
       1. LoansDB.php: Creates / opens the database and creates the loans table if needed.
       2. LoanRow.php: contains queries for the loans table.
     + /logs: debug log. If you want to see the log, you’ll have to manually create this directory. Note that the debug level is currently set high, but it’s easy to turn off completely (in LoansController.php).