Max Plomer

maxplomer@gmail.com ● codingprojects.co ● 203-945-8606 ● New York, NY

**Education**

**App Academy** Sept 2014 - Nov 2014

*Coding Bootcamp in New York, NY*

**University of Connecticut** Completed May 2013

*Master of Science in Mechanical Engineering*, GPA: 3.5/4.0

**University of Connecticut** Completed Dec 2010

*Bachelor of Science in Mechanical Engineering*

**Experience**

**Robert Half Consulting** February 2016 – Present

*New York, NY*

Project: Truveris / OneRx.com

● Collaborated with UX/UI team at a pharmaceutical software company correcting React.js app to mock-ups and troubleshooting responsive design elements in a very complex mobile-friendly frontend

● Investigated design interaction and implemented integration between map and list components by creating a custom event on the window object, when user hovered over item in list the associated Google Maps marker bounces twice and comes to front of markers

● Developed a React.js text input component to fix the placeholder text disappearing on focus glitch for Internet Explorer versions 10 and 11, as this was an important design interaction

**Gramercy Consultants** July 2015 – December 2015

*New York, NY*

● Crafted a great user experience for commercial real estate brokers with an HTML / JS front-end and straightforward JSON API routes customized for the front-end to consume

● Created secure digital downloads feature using Stripe API to process payments; each download link contains a cryptographically strong token generated by SecureRandom's urlsafe\_base64 method

● Setup authentication stack using Devise / CanCanCan / Rolify for secure API calls and user permissions

● Analyzed new project proposal for multiplayer browser-based game; evaluated tools (Ruby, JavaScript, Mongo, PostgreSQL, Ruby on Rails, Volt, Ember.js, Meteor, Unity3D) to solve the problem statement and chose the best solution to bootstrap a playable prototype

● Optimized real-time Volt app with a complicated data/model structure; finished features such as reset password that are built into the Volt User System, but were not yet finished by the Volt team

**The Psalm of Howard Thurman Documentary Film** Apr 2015 - May 2015

*Boston, MA*

● Used Adobe Muse to generate the most cutting-edge single-page flat user interface

● Deployed Adobe Muse app in hosting environment that supports PHP server-side scripting

● Explored many template and design technologies, such as Twitter Bootstrap templates and scratch coding CSS/JS

● Helped refine content and customize Muse template to best tell the story of Howard Thurman

● Staging URL => http://www.jsbox.co/howardthurman

**Free Fundraise.com** Mar 2015 - May 2015

*Boston, MA*

● Engineered API-driven Backbone.js app enabling you to shop and give the advertising affiliate fees generated to the charity of your choice

● Designed and programmed user interface in Backbone.js with carousels built on jQuery fading methods

● Allowed users to either select a featured charity, or to explore all, which adds a selected charity to the featured collection as well as activating it for donations

● Integrated charity tracking-ids into each unique set of query string parameters required by Amazon, eBay and FlexOffers affiliate programs

**Outlearn.com** Jan 2015 - Feb 2015

*Boston, MA*

● Researched technical training topics and formatted content into prototype format

● Integrated with GitHub through importation of repositories and allowing login using GitHub account

● Experimented with JavaScript Fancytree library, a dynamic tree view plugin for jQuery, to display imported contents

● Engineered advanced coding challenges using RSpec tests for Ruby and Jasmine tests for JavaScript

● Created interactive terminal sessions for Ruby, JavaScript and PHP using jq-console, a jQuery plugin

**Westport, CT and World Maker Faires** Apr 2014 (Westport) and Sept 2014 (World)

*Westport, CT and New York, NY*

● Presented the science behind my software package Clean Fuel Chemical Kinetics and how it can be used to model clean fuels in the engines of tomorrow

● Articulated reaction rate and Lagrange multiplier problems to teach science and math to faire attendees

**Saigeworks LLC** May 2014

*Trumbull, CT*

● Utilized my chemical kinetics software to transform inventor’s hands-on experience into concrete engineering terms

● Determined that 2.4 kJ of work and 3.3 kJ of heat for methane, and 3.0 kJ of work and 5.1 kJ of heat for hydrogen, was produced in one cycle per gram of fuel and oxygen at stoichiometric ratio

**University of Connecticut** Jan 2011 - Dec 2012

*Storrs, CT*

● Teaching Assistant for Combustion, Compressible Flow, Renewable Energy, and Senior Design Project classes

**Projects**

**Volt Framework Prototypes** Dec 2015 - Present

The Volt Ruby framework (voltframework.com) is great for rapid prototyping of web apps. It integrates MongoDB, WebSocket, and Opal Ruby-to-JavaScript converter right out of the box.

Investing App

*backend: github.com/maxplomer/investing\_rest\_api, frontend: github.com/maxplomer/investing\_angular2*

● Uses Volt as backend API and Angular2 for frontend to show 10 recent stock trades

● Deployed Ruby backend using Dokku an open source Platform as a Service; GitHub Pages is used to host Angular2 frontend

● Modified Volt HTTP Controller to add Access-Control-Allow-Origin header to response

Quick Calculator

*github.com/maxplomer/quick\_calc*

● Simple online calculator that show recent calculations, evaluates equations safely in the browser

● Uses Volt built-in flash alerts to display "We were unable to evaluate your equation!" error message

Simple Public Timeline

*github.com/maxplomer/simple\_public\_timeline*

● Effectively uses Volt backend as a proxy server between client browser and Twitter API via the Twitter Ruby gem

● An asynchronous test was created that calls the get\_tweets task and ensures it returns an array of proper length

Volt Todo App

*github.com/maxplomer/volt-tutorial*

● Low fidelity todo app that explores project structure using database associations (Project has many Tasks, Task has many Items)

● Inputs and textareas are saved/updated in real-time

● Evaluate JavaScript, such as calls to confirm() method, directly in back-ticks (``) when no Opal/Ruby method is given

Exploration of NYC Event API

*github.com/maxplomer/api\_explorer*

● Built app for App Academy hackathon that explores different routes of the NYC Event API

**Ember.js Todo App** Sept 2015

*github.com/maxplomer/ember-todo*

● Instead of using Ember Data to manage model data, AJAX calls containing an array of todos are used to keep data synced between the Ember.js front-end and Rails back-end

● List of todos stored as serialized data on a text attribute of a single ActiveRecord model instance

● Deployed Ember.js app on Heroku using ember-cli-rails gem; increased build timeout to fix Heroku glitch

**Carbon Footprint.co** May 2015

*github.com/maxplomer/carbon-calculator*

● Architected AngularJS app that performs carbon footprint calculation and persists data to Ruby on Rails back-end

● Added user authentication using the angular-devise service, and charted user progress with the angular-charts component

● Sent result of footprint calculation, coded in Ruby, to the AngularJS front-end via API as an attribute in the ActiveModel Serializer

**Andrew Talty.tv** Apr 2015

*github.com/maxplomer/andrewtaltytv*

● Used jquery-tubular plugin to display a responsive YouTube video in the background of a TV producer’s Ruby on Rails portfolio page

● Detected client’s browser type using JavaScript’s navigator object, then displayed a still image background for mobile devices

● Created a subtle effect by blurring social media and video control buttons using the text-shadow CSS attribute

**JavaScript Sandbox.com** Dec 2014

*github.com/maxplomer/javascript\_sandbox*

● Constructed online code testing environment using Ruby on Rails and Backbone.js, great for evaluating HTML, CSS and JavaScript code in the browser

● Architected user authentication using a current\_user Rails controller and corresponding current\_user Backbone.js Model

● Selected the Bootstrap navbar component to create an intuitive site navigation

**Real Estate Tracker.co** Dec 2014

*github.com/maxplomer/real\_estate\_tracker*

● Presented complex financial analyses for investment properties using instantly updating forms and compelling diagrams

● Communicated data to API using jQuery Ajax requests from within the Ruby on Rails rendered view

**Coding Challenges.co** Nov 2014

*github.com/maxplomer/coding\_challenges*

● Enlisted Ruby on Rails to create a community where users can compete and boost their programming skills

● Harnessed Ace high performance code-editor to deliver code syntax highlighting for Ruby, JavaScript and SQL

● Jailed user-submitted code as to only permit safe method calls

**Bit Chart.co** Nov 2014

*github.com/maxplomer/finance-clone*

● Built stock investing competition application using Ruby on Rails and Backbone.js; a pure Rails version was also created

● Applied Heroku scheduler and market\_beat ruby gem to save stock market quotes to a PostgreSQL database

● Graphed user portfolio performance using Highcharts pure JavaScript library; calculated chart data using algorithm optimized for least SQL queries

**Clean Fuel Chemical Kinetics** June 2013 - Dec 2013

*combustionhelp.com*

● Programmed simplified chemical kinetics software package in MATLAB; a version using hybrid of GNU Octave and Fortran was also created

● Investigated CHEMKIN-II mechanism format, Arrhenius/Lindeman/Troe form reactions, calculating thermodynamic data from NASA polynomials, programming language speeds and license costs

● Applied MATLAB profiler to improve program inefficiencies; time to converge constant volume reactor with methane fuel was reduced from 619 sec to 0.8 sec

**Technology Skills**

**Languages** => Ruby, JavaScript, HTML, CSS, SQL, MATLAB, C, Fortran

**Tools** => Ruby on Rails, jQuery, AngularJS, Backbone.js, Ember.js, Git, Linux, AWS, Heroku

**Hobbies** => Raspberry Pi, Unity3D Game Engine, Surfing, Biking

**Favorite Ruby Gems** => rest-client, whenever