Assignment 1 - SENG365 2017

Introduction

The objective of this assignment is to implement the server-side component of a RESTful API for a crowd-funding site (think Kickstarter or Pledgeme). A short introduction to crowdfunding follows in a later section.

Assignments 1 and 2 combine to give you the full experience of building a complete web application following an API-first methodology. The lectures and labs this term will give you the tools you'll need to implement a web service to deliver the crowdfunding API in Assignment 1. You'll also design and develop a web client in Assignment 2 that uses the API to match projects with potential funders.

We'll discuss the domain and the API design for both assignments in the first two weeks of class, and at the end of the second week we'll have developed a formal API specification for Assignment 1.

Marks and due date

Assignment 1 contributes towards up to 15% of your final grade.

20 July	Introduce assignment, and discuss business side of APIs in context of assignment
27 July	API design refined during classroom discussion
30 July	API specification finalised
11:55pm 21 August	**Assignment 1 due**
11:55pm 28 August	Drop dead date, with 15% penalty

There is an assignment forum on Learn. You may ask questions about the assignment there, and any updates or clarifications will be posted to that forum.

Assessment

This assignment is all about correctness and meeting expectations for the API; your creativity should be directed towards a clean and accurate implementation.

Among other things, we'll be looking for:

1. The proportion of the API that you've implemented correctly.
2. Appropriate handing of error conditions.
3. Effectiveness of your implementation of the authentication and authorisation requirements (given in "Authentication and Authorisation" below.)

You may use whatever technology you wish for the implementation, but we'll be covering Node.js, Express, and MySQL in the labs, so we strongly recommend

you use those. Regardless of technology though there are some things your API implementation must do ("It's not just about the features." ™):

1. It must be persistent - we'll run it, stop it, and expect any changes to still be there. You're responsible for any backend storage to enable this. If you chose a database (and we expect that you will), you will need to also design an appropriate database schema.
2. You must also make sure that your application will automatically initialise any storage that you use, and add any sample data you might have created. We will expect your application will be able to recreate any necessary storage if we reset it, as we might during testing to be sure that we're working from a clean installation.
3. Any third-party client (maybe even another student's app) may connect to your API and they will expect it to work precisely as documented in the API specification. For each element of the API that you implement, we'll expect it to work just as defined.

Submission

Your code for this assignment must be committed to your personal git repository for SENG365 on eng-git.canterbury.ac.nz. We'll assess the last commit made before the cut-off time for the assignment, and we'll expect to find your service running on your assigned port on the SENG365 docker host.

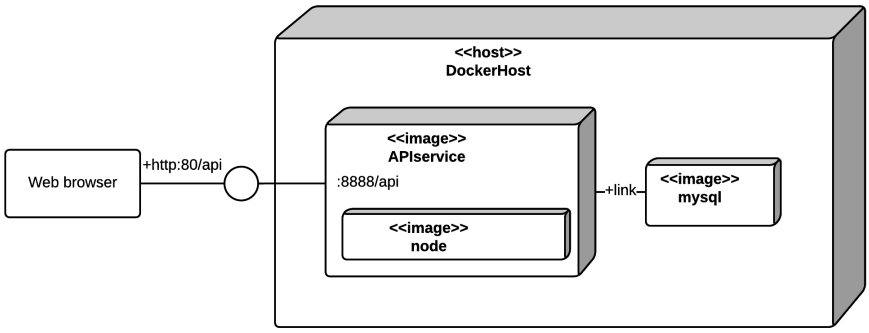

Figure 1: Components for Assignment 1

We've configured a Continuous Deployment pipeline for SENG365.

Any commit made to your repository will automatically trigger a series of steps that should end with your application running on the SENG365 docker host at the following URL (if all is well): `http://csse-s365.canterbury.ac.nz:<port>`where `<port>` is a unique port number that we will assign to you for the duration of the class.

This all depends on these four files in your repository (we've already put them there):

1. package.json (you will have to maintain this file as you'll learn in the labs.)
2. docker-compose.yml (you might have to edit this if you're using something other than MySQL.)
3. Dockerfile (you shouldn't have to touch this one.)
4. .gitlab-ci.yml (or this one.)

API implementation

You are to implement the back-end (server-side) API for our crowdfunding site. The design of the API was introduced in Workshop 2, and is driven by the needs of the front-end to be developed later this semester in Assignment 2. User stories for Assignment 2 are available now for reference only in the Workshop 2 handout, available on Learn.

The API you are to implement is described by the downloadable Swagger definition file from the assignments section of Learn. You can load the definition into the free online viewer at http://editor.swagger.io to see documentation for the API.

Authentication and Authorisation

There are some general rules your implementation must follow:

- A Person must be a Creator of a Project to modify any of the elements of that Project.
- Only the Backer of a Pledge to a Project, or the Creator of the Project with that Pledge, can modify or remove that Pledge.
- Only the Backer of a Pledge to a Project can receive the Reward associated with the Pledge.
- Only those Pledges that are not 'anonymous' are visible to anyone other than the Backer of the Pledge.
- No credit card numbers can be held in persistent storage.

Introduction to crowd-funding

If you've ever given money to a project on Kickstarter, or donated to a good cause on GiveALittle, you've used a crowdfunding app. Crowdfunding is the means by which people, organisations or companies attempt to gather financial support for their projects from a large and diverse gathering of individuals.

Instead of hoping to raise a large amount of money from a few sources, they turn it around and look for a little from a lot of people (in the ideal case; more often it's a little from a little–Kickstarter has a 35% success rate–and the project goes under :)

Crowdfunding is grass-roots fundraising, and has taken off with the availability of a low-cost, low-friction channel: web applications. GoFundMe advertises that they have raised over $3 billion USD for projects on their site since 2010. More than 10 million people have backed projects on Kickstarter since it launched in 2009.

We can identify three different types of crowdfunding application:

1. Funneling money to causes (generally altruistic, no rewards other than perhaps recognition)
   - GoFundMe
   - Crowdrise
   - Givealittle
   - Boosted ("Bringing artists and audiences closer together than ever")
2. Project/reward-based (to fund specific projects, delayed gratification)
   - Kickstarter (now a Benefit Corporation with explicit social goals.)
   - PledgeMe ("Helping Kiwis fund the things they care about".)
   - Pozible (for creative industries in Australia. Claims 57% success, with fees of 3% > $500K, 4% > $100K.)
3. Equity-based (take a stake in a company, marketplace investing)
   - Seedrs
   - Circleup
   - Crowdcube
   - PledgeMe equity

Crowdfunding is quite similar to some other well-known domains, so your work on this assignment might come in handy if you end up working in any of these areas in the future:

- Marketplaces - auctions, listings
- Skills matching
- Microfinancing - peer-to-peer or crowd (e.g., crowdrise)
- Volunteer/project/resources matching
- Ongoing financial support for individuals working in the arts or sciences (e.g., Patreon)