

Lexicon Digital Coding Challenge



Prince's Theatre

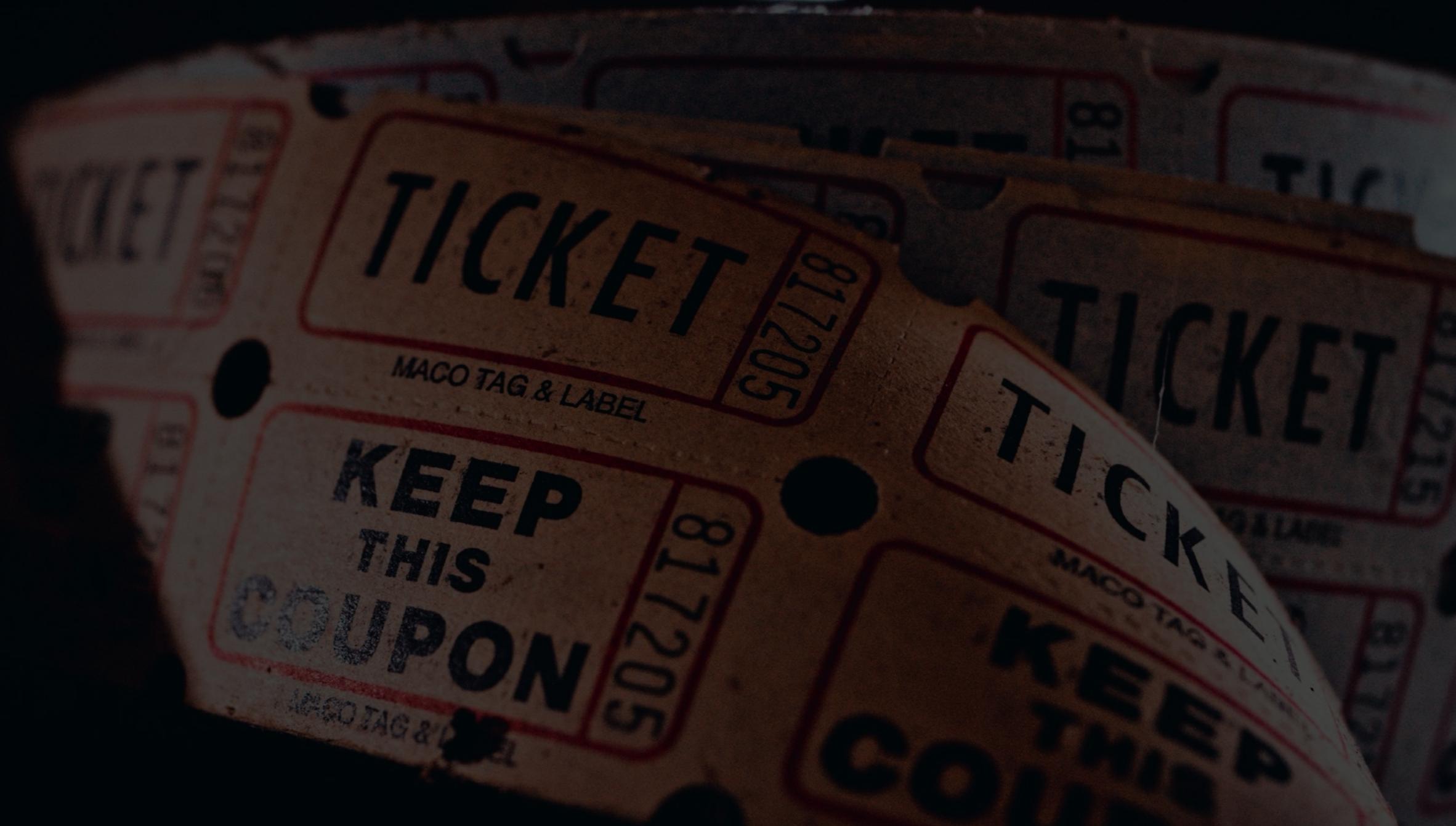
Introduction

This is a coding challenge that has been developed to give you an opportunity to show us your ability. It involves solving a small problem such as you may encounter working at a client. In this scenario you are a Developer responsible for deciding what to build and how.

The client requirements are communicated in the form of a problem description. Be sure to read them carefully and understand all the information. They will give you pointers that will help you develop an ideal architecture that fits this problem.

The brief has intentionally been made flexible to give you space to show off your ability. Where specifics have not been provided, you are encouraged to make and document any assumptions you think are sensible.

Movie Price Comparison



Problem Description

Prince's theatre is a small independent theatre in the inner eastern suburbs of Melbourne. The Prince family has run it for over 50 years. It has a small faithful clientele who love watching niche independent films.

Now well past retirement age, the theatre has been passed onto grandchildren Jane and Travis Prince. Jane is a graduate in digital marketing and Travis is a successful tech blogger. They both have big plans for the cinema and want to bring this old world institution into the digital age.

The first thing they want to achieve is building up their digital presence. Jane has a hypothesis that there is a lack of online aggregators that go through different streaming websites to show which has the lowest available price. Cinemaworld and Filmworld are two websites that Travis has identified as being popular with their target audience.

They have decided to engage Lexicon Digital to develop a solution that can test their hypothesis. We must build functionality that will allow their users to see the movies that are available and then see which of the movie providers is streaming it at a cheaper price.

Jane has heard rumours that the API sharing the movie catalogue might not be reliable. As this solution will be user facing, they want it to be able to handle any issues in a way that will create good user experience. The site should remain functional and be able to handle any instability.



API Details

There is one API that supports two operations where film information for Filmworld and Cinemaworld can be accessed:

- <https://challenge.lexicondigital.com.au/api/{cinemaworld or filmworld}/movies> - that returns the movies available;
- <https://challenge.lexicondigital.com.au/api/{cinemaworld or filmworld}/movie/{ID}> - that returns details about a single movie.

Access to the web service is authenticated by adding an additional header field: `x-api-key`. The value for `x-api-key` has already been provided in the invitation email.

This endpoint is unreliable and you should anticipate problems when you use it.



Deliverables

You must develop a solution that can solve their core problem: allow users to see which of the two streaming providers are streaming their chosen movie at a cheaper price. The supplied wireframes are a suggested user experience for web, however, if you choose to provide a mobile solution, please use the web designs and adapt them to mobile format. You may deviate from them to show off your coding flair so long as your solution still solves Jane and Travis's problem.

Your task is to complete the challenge by creating a solution that solves the problem with functionality developed in your speciality (Front-End/Mobile or Full-Stack). You can code in the language and tools you are most familiar and comfortable with (C#, Java, Kotlin, Go, Javascript, etc).

A link to a private Github/Gitlab/Bitbucket repository should be emailed to Lexicon Digital once the challenge is complete. Access should be granted to the Lexicon Digital reviewer account as mentioned in the email. The Github/Gitlab/Bitbucket repository with a working solution should also contain:

- Documentation
- Source files
- Test code
- A full commit history for the solution



What We're Looking For

We will be evaluating the following:

- The modelling of the domain and how it's translated into code.
- The demonstrated knowledge of the language and tools used.
- Code that is readable and maintainable by being simple, concise and idiomatic with good test coverage.
- The README is important, we need instructions, context, architectural elements and decisions, trade-offs and assumptions made.
- We want you to be creative and show off your ability but stay within the core customer problem: allow users to see which of the two streaming providers are streaming their chosen movie at a cheaper price.



Copyright Lexicon Digital 2020. All Rights Reserved.