Technical Assessment Case Studies

Please complete of following case studies:

1. myRetail RESTful service

For the case study you choose please meet the following requirements:

- Complete the exercise in the technical stack of your choice.
  - When appropriate use a data store of your choice.
  - Use any external frameworks you desire
  - Be ready to discuss your recommendations to make your solution suitable for use in a production environment

- Provide evidence of the result to the interviewers *(choose one)*
  - Unit test results or other documented output
  - Hosted instance of the implementation
  - Runnable instance of the implementation on your computer

- The end result should be a functional implementation of the problem preferably with associated tests
  - Provide the working code either in a publicly accessible hosted repository like git
  - Provide a README.md file with instructions for testing, running and interacting with your application and any details you feel are relevant to share

- Please bring either a laptop or a hard copy of the code to help facilitate review at the interview.

1. myRetail RESTful service

myRetail is a rapidly growing company with HQ in Richmond, VA and over 200 stores across the east coast. myRetail wants to make its internal data available to any number of client devices, from myRetail.com to native mobile apps.

The goal for this exercise is to create an end-to-end Proof-of-Concept for a products API, which will aggregate product data from multiple sources and return it as JSON to the caller.

Your goal is to create a RESTful service that can retrieve product and price details by ID. The URL structure is up to you to define, but try to follow some sort of logical convention.

Build an application that performs the following actions:

- Responds to an HTTP GET request at /products/{id} and delivers product data as JSON (where {id} will be a number.
- Example product IDs: 15117729, 16483589, 16696652, 16752456, 15643793)
- Example response: {"id":13860428,"name":"The Big Lebowski (Blu-ray) (Widescreen)","current_price":{"value": 13.49,"currency_code":"USD"}}
- Performs an HTTP GET to retrieve the product name from an external API. (For this exercise the data will come from redsky.target.com, but let's just pretend this is an internal resource hosted by myRetail)
- Example: http://redsky.target.com/v2/pdp/tcin/13860428?excludes=taxonomy,price,promotion,bulk_ship,rating_and_review_reviews,rating_and_review_statistics,question_answer_statistics
- Reads pricing information from a NoSQL data store and combines it with the product id and name from the HTTP request into a single response.
- BONUS: Accepts an HTTP PUT request at the same path (/products/{id}), containing a JSON request body similar to the GET response, and updates the product's price in the data store.