Technical Assessment

# Programming Case Study

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. You should build the solution using JavaScript (Node.js), Java or Groovy, and a NoSQL data store such as Cassandra or MongoDB. You are free to use any frameworks and application servers that you choose.

Build an application that performs the following actions:

* Responds to an HTTP request at /products/{id} and delivers product data as JSON (where {id} will be a number.
  + Example product IDs: 15117729, 16483589, 16696652, 16752456, 15643793)
  + Product ID’s actually used for this exercise:

|  |  |
| --- | --- |
| **Product ID** | **Product Description** |
| 14423830 | Apple iPad Air 16GB Wi-Fi - Silver/White (MD788LL/B) |
| 14423840 | Apple iPad Mini 2 with WI-FI 16GB - Space Gray/Black (ME276LL/A) |
| 17293894 | Samsung Galaxy Tab A 8" - Smoky Titanium |
| 17293888 | Samsung Galaxy Tab A 16GB 9.7" Tablet - Smoky Blue (SM-T550NZBAXAR) |
| 14000294 | Apple iPod Nano 16GB (7th Generation)with touch-screen - Purple (MD479LL/A) |
| 14000289 | Apple iPod shuffle 2GB MP3 Player - Silver (MD778LL/A) |
| 14213715 | Apple iPod Touch 64GB MP3 Player (5th Generation)- Pink (MC904LL/A) |

* + Example response: {"id":13860428,"name":"The Big Lebowski (Blu-ray) (Widescreen)","current\_price":{"value": 13.49,"currency\_code":"USD"}}
* Performs an HTTP request to retrieve the product name from an external API. (For this exercise the data will come from api.target.com, but let’s just pretend this is an internal resource hosted by myRetail)
  + Example: <https://api.target.com/products/v3/13860428?fields=descriptions&id_type=TCIN&key=43cJWpLjH8Z8oR18KdrZDBKAgLLQKJjz>
* 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.
* Includes appropriate test cases implemented using a framework such as JUnit, Spock, or Mocha.
* Contains a README.md file with instructions for testing, running and interacting with your application.
* BONUS: Accepts an HTTP request to update the product’s price in the data store. Hint: Ideally, it should respond to a different HTTP method at the same path (/products/{id}).
* BONUS: Contains a build script (pom.xml, build.gradle, package.json, etc) that can be used to build, run or package the application.

Deliverables

Send us a package (.zip or .tar.gz) containing the source code including scripts and configuration files. Compiled and derived artifacts (such as .class files or the resulting .war) are not required if the project contains a build script.

ALTERNATIVE/BONUS: put your project in a public GitHub repository and send us directions for accessing it.