

## Technologies / Libraries to leverage:

- Java 8
- Spring (<http://start.spring.io/>)
  - Web
  - MyBatis
  - H2 DB
  - Actuator
- Swagger Springfox
- JUnit
- Jacoco
- Gradle
- Git

## Functional Requirements:

- Advertiser
  - Fields
    - Advertiser Name
    - Advertiser Contact Name
    - Advertiser Credit Limit
  - Manage in H2 DB
- Restful API for an Advertiser (base path = /api/advertiser)
  - POST new advertiser
  - PUT update advertiser
  - DELETE advertiser
  - GET advertiser
  - GET endpoint to validate if the advertiser has enough credit to perform a transaction
  - Properly handle errors and don't expose a stack trace through the API
- Make sure the advertiser API endpoints are accessible via a swagger UI embedded in the application at <http://localhost:8080/swagger-ui.html>
  - Do not expose the actuator endpoints via Swagger
- Make sure you have at least 90% Test coverage via gradle test for all controllers and domain models
- Make sure the end build artifact is an executable jar
- Make sure the actuator endpoints are working
  - Add some sort of relevant information to /info
- Add a README.md with some description on how to develop, test, run
- Must be a git repository with several commits
  - Share via GitHub or BitBucket, or .zip file via email, your choice

**Extra Credit options:**

- POST endpoint to deduct credit from the advertiser's account
- Manage H2 Schema via Flyway
  - Make sure the flyway actuator endpoint is working
- Leverage Lombok Annotations in the Advertiser Model
- Build the resulting jar into a Dockerfile
- 100% Test Coverage
- Run the management / actuator endpoints on 8090 instead of 8080
- Granular git commits and commit history (no points off for code issues / mistakes throughout the git history, only judged on the HEAD git repo state)
- Any additional functionality you want to tack on :)