

Coding Test (Dev)

Registration with Exclusion Services

Build a web application that offers a remote service interface (endpoint) to register a user. A user can only register if he is not part of a blacklist. The goal is to demonstrate that the web application runs, the tests are passing and real HTTP requests can be made. In summary:

1. Set up a way to run a web application server
2. Implement an HTTP remote endpoint for the 'register' operation based on the given interface specifications
3. Implement a *simple* way for the web application to remember already registered users (no real persistence needed, in-memory is sufficient)
4. Implement a way to use the given 'ExclusionService' interface to simulate the business logic of validating the user against a black-list. The ExclusionService *does not need to be implemented* but can be stubbed out/ faked.
5. Check the implementation with unit tests and integration tests

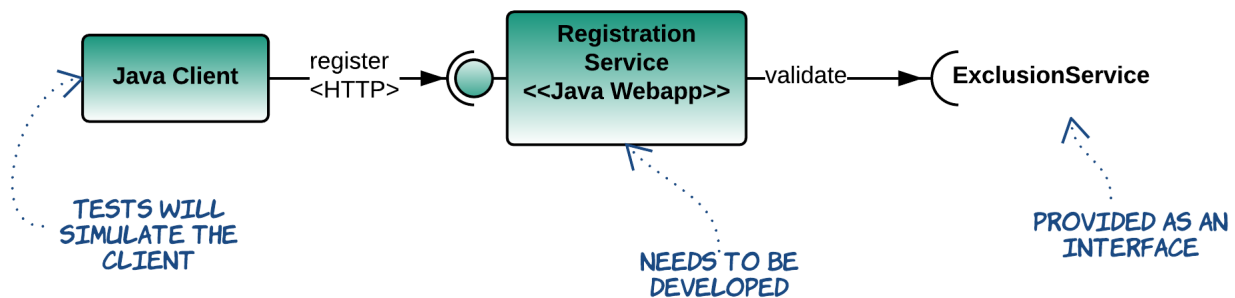
It is not expected that all edge cases and error scenarios are covered nor that anything more is done than to implement the **minimal requirements in a clean and reasonable** way. Any detail that is missing in the requirements can be implemented at discretion. If the candidate feels the wish to use more technology than needed (e.g. build systems, persistence layers, testing tools), he is happily invited to do so as long as he can successfully run the system on our computers (see section Execution Environment). The project will have to be demonstrated in the interview and will be open for discussion. This will also be a good chance to talk about anything that was unclear, couldn't be implemented or leaves room for improvement.

Technology

The choice of technologies and frameworks is up to the candidate as long as they are Java-based and freely available:

- **Programming Language**
Java (up to version 8)
- **Webserver**
Any e.g.: Jetty plugin in Maven, Dropwizard, Tomcat..
- **Remote Service Protocol**
Any HTTP based one, e.g. REST (preferred), SOAP..
- **Testing/ Stubbing**
Any (e.g. Junit, Mockito, self-made stubs, rest-assured, SoapUI ..)

Architecture



Registration Service (Web Application)

This will be the web application the candidate needs to set up and implement. It offers an HTTP endpoint (REST, SOAP or other) to register a user. The definition of the endpoint is:

- method name: "register"
- parameters:
 - username (alphanumeric, no spaces)
 - password (at least four characters, at least one upper case character, at least one number)
 - date of birth
 - Social Security Number (SSN) (does *not* have to be validated, for better understanding see: http://en.wikipedia.org/wiki/Social_Security_number)
- checks:
 - parameter's are valid according to specification
 - user hasn't been already registered
 - user is not blacklisted (see 'Registration' service)
- returns ok/ empty/ void response if user could be registered, for any failure throws an appropriate remote exception/ error

Exclusion Service

The exclusion service is given as an interface and does not need to be implemented. For use in the Registration Service it can be stubbed, mocked or faked.

Exclusion Service

```
/**
 * Service to offer validation of a user against a 'blacklist'. Blacklisted users
 * fail the validation.
 * @author gamesys
 */
public interface ExclusionService {
    /**
     * Validates a user against a black list using his date of birth and social
     * security number as identifier.
     *
     * @param dob the user's date of birth in ISO 8601 format
     * @param ssn the user's social security number (United States)
     * @return true if the user could be validated and is not blacklisted, false if
     * the user is blacklisted and therefore failed validation
     */
    boolean validate(String dob, String ssn);
}
```

Execution environment

- Mac OS X
- Java 8 (Java as language)
- Git
- Maven (optional)

How to submit the project

1. Sign up for a free personal account on <http://bitbucket.org>.
2. Fork https://bitbucket.org/gamesys_careers/coding-exercise.git (you MUST create a private fork). Please note: The provided pom.xml (Maven) is optional for this exercise to use and can be deleted.
3. Commit and push your submission to your fork.
4. If you want to comment on your work in advance or made some essential assumptions, please provide a README
5. Share the repository with the user gamesys_careers. Do NOT create a pull request, as we will not merge them!
6. We'll acknowledge receipt.