**Java Practical Test**

You may use any preferred technologies or libraries, but you MUST use:

1. Maven or Gradle. (Gradle preferable)
2. Java for the API.
3. Make use of an in-memory or embedded db (ie. H2) for persistence.

Making use of the Spring/SpringBoot framework is not required but is preferred.

**Test Submission:**

If possible, submit your source to a public SCM host (Preferably Git based) and provide Global Kinetic with access details and documentation on how to run your project(s).

Assessment will be based on:

* Proficiency using the required technologies
* SCM usage
* Project portability
* Documentation (if any, can be in simple markdown format)
* Simplicity
* Improvisation

**NOTE:**

**Senior** candidates are *strongly advised* to complete the **optional** items.

**[Section 1: API - Create a RESTful service]**

1. Refer to the ***API Contract*** below.
2. The context root is: /api
3. You may modify/create end-points or payload properties if required.
   * This must be clearly indicated in your documentation.

**[Section 2: Client - Invoking the API endpoints]**

1. Create a client for *Section 1*.
   * Client can be based on any of these: Web/SPA/JMeter/Postman.
2. Add functionality to allow users to login/logout.
3. Allow users to submit their details containing a username, phone number and password.

**[Section 3: Security and Session management]**

1. Ability to list of all the unique users that are registered.
2. Security token must expire after 3 minutes of inactivity.
3. Ability to list or count users that have a valid session.
4. Restrict the functionality in 3. to authenticated users only.
5. When a user token expires or logs out, this must reflect when invoking 3.

**Optional:**

1. Send the session token as a Base64 Header.
2. Make use of encryption.

***API Contract :***

**[PUT : /users]**

request body :

{

    "username" : "user\_name",

    "phone" : "phone",

    "password" : "user\_password"

}

**[GET : /users]**

response body :

{

    "users" : [

        {

            "id": "user\_id",

            "phone": "phone"

        }

    ]

}

**[POST : /login]**

request body :

{

    "username" : "username",

    "password" : "user\_password"

}

response body :

{

    "id" : "id",

    "token" : "session\_token"

}

**[POST : /logout/{id}]**

request body :

{

    "token" : "session\_token"

}