# **Exercise for Java Developer Candidates**

## **Requirements**

Please implement the couple of RESTful web APIs specified in the appendix for creating and retrieving details of a BrightTALK entity.

## Technical **Constraints**

The application should be implemented as a Java web-app using Spring MVC and MySQL 5.x, and be deployable to Tomcat. You are otherwise free to design and build the code in whatever way you see fit.

## **Guidelines**

Our aims in setting candidates this exercise are:

* To check the quality of Java code produced by candidates.
* To gain some level of assurance that candidates have experience of the relevant APIs and frameworks we currently use.

There are no intended tricks or traps in this exercise.

Please submit at minimum your source code, a DB schema, a *deployable* Java app (and any other artifacts you wish to supply to support your application) to [**dev-jobs@brighttalk.com**](mailto:dev-jobs@brighttalk.com) with subject line **Java Evaluation Exercise.**

Please also feel free to provide feedback if you wish about the exercise.

Best of luck and thanks in advance for the time spent.

The BrightTALK Tech Team.

PS - If you have examples of other, relevant software you have produced which is publicly available e.g. contributions to an open source project, which you would like us to take into account then please feel free to references these also in your email. This is optional, and is not a substitute for completing the coding exercise.

# 

# 

# **Appendix - API Specification**

The APIs specified below support the creation and retrieval of representations of a User Realm.

A User Realm is a context for the registration and authentication of a user, which comprises a unique integer identifier, a unique name, an (encryption) key and a description.

For the purposes of this exercise all that is required is to implement APIs to create (store) and retrieve the entity:

* JSON representations must be supported. Support for XML representations (as shown in the API spec below is optional).
* No security (authentication or authorisation) checks are required on the APIs. The code for generating the encryption key can be stubbed, returning any string.

## **Create Realm**

### **Purpose**

Creates a new realm including the generation of an encryption key. (Note, for the purposes of this test the key generation / algorithm can be stubbed).

### **Request**

POST /service/user/realm HTTP/1.1

Content-Type: application/xml; charset=utf-8

<realm name=”{name}”>

<description>{description}</description>

</realm>

### **Response**

**Success**

HTTP/1.1 201 Created

Content-Type: application/xml; charset=utf-8

Date: Tue, 15 Nov 2010 08:12:31 GMT

<realm id=”{id}” name=”{name}”>

<description>{description}</description>

<key>{key}</key>

</realm>

**Errors**

If the mandatory realm name is not supplied:

HTTP/1.1 400 Bad Request

Content-type: application/xml; charset=utf-8

Date: Tue, 15 Nov 2010 08:12:31 GMT

<error>

<code>InvalidRealmName</code>

</error>

If the requested realm name matches the name of an existing realm.

HTTP/1.1 400 Bad Request

Content-type: application/xml; charset=utf-8

Date: Tue, 15 Nov 2010 08:12:31 GMT

<error>

<code>DuplicateRealmName</code>

</error>

## **Get Realm**

### **Purpose**

Returns the details of an individual realm, identified by its unique id.

### **Request**

GET /service/user/realm/{id} HTTP/1.1

Accept: application/xml

### **Response**

**Success**

HTTP/1.1 200 OK

Content-Type: application/xml; charset=utf-8

Date: Tue, 15 Nov 2010 08:12:31 GMT

<realm id=”{id}” name=”{name}”>

<description>{description}</description>

<key>{key}</key>

</realm>

**Errors**

If the requested realm id is not an integer value -

HTTP/1.1 400 Bad Request

Content-type: application/xml; charset=utf-8

Date: Tue, 15 Nov 2010 08:12:31 GMT

<error>

<code>InvalidArgument</code>

</error>

If the requested realm id does not identify an existing realm.

HTTP/1.1 404 Not Found

Content-type: application/xml; charset=utf-8

Date: Tue, 15 Nov 2010 08:12:31 GMT

<error>

<code>RealmNotFound</code>

</error>

## Realm Entity

|  |  |
| --- | --- |
| **Field** | **Description and Constraints** |
| id | Unique ID. Primary key. System-generated. |
| name | Realm name (alias for ID). Must be unique. |
| description | Realm description. Up to 255 chars. |
| key | Realm encryption key. Fixed length 32 chars. |