Java Skills Test - BrightTALK

The test is designed to provide an indicator of your technical level. We are looking for the right approach and appropriate considerations, not a business-ready solution. There are no intended tricks or traps in this 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.

Technical Constraints

The application should be implemented as a Java web-app using Spring Boot RESTful MVC, store data in a relational database, and be deployable to Tomcat. You are otherwise free to design and build the code in whatever way you see fit.

Submission

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). This can be either a link to an online repo, or attached directly in an email.

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 and XML representations must be supported.
- 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

XML Representation

```
POST /service/user/realm HTTP/1.1
Content-Type: application/xml; charset=utf-8

<realm name="realmName">
  <description>realmDescription</description>
</realm>
```

JSON Representation

```
POST /service/user/realm HTTP/1.1
Content-Type: application/json; charset=utf-8

{
  "name": "realmName",
  "description": "realmDescription"
}
```

Response

Success

XML Representation

```
HTTP/1.1 201 Created
Content-Type: application/xml; charset=utf-8
Date: Tue, 15 Nov 2010 08:12:31 GMT

<realm id="1234" name="realmName">
  <description>realmDescription</description>
  <key>realmKey</key>
</realm>
```

JSON Representation

```
HTTP/1.1 201 Created
Content-Type: application/json; charset=utf-8
Date: Tue, 15 Nov 2010 08:12:31 GMT

{
  "id": 1234,
  "name": "realmName",
  "description": "realmDescription",
  "key": "realmKey"
}
```

Errors

If the mandatory realm name is not supplied:

XML Representation

```
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>
```

JSON Representation

```
HTTP/1.1 400 Bad Request
Content-type: application/json; charset=utf-8
Date: Tue, 15 Nov 2010 08:12:31 GMT

{
 "code": "InvalidRealmName"
}
```

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

XML Representation

```
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>
```

JSON Representation

```
HTTP/1.1 400 Bad Request
Content-type: application/json; charset=utf-8
Date: Tue, 15 Nov 2010 08:12:31 GMT
{
  "code": "DuplicateRealmName"
}
```

Get Realm

Purpose

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

Request

XML Representation

```
GET /service/user/realm/{id} HTTP/1.1
Accept: application/xml
```

JSON Representation

```
GET /service/user/realm/{id} HTTP/1.1
Accept: application/json
```

Response

Success

XML Representation

```
HTTP/1.1 200 OK
Content-Type: application/xml; charset=utf-8
Date: Tue, 15 Nov 2010 08:12:31 GMT

<realm id="1234" name="realmName">
  <description>realmDescription</description>
  <key>realmKey</key>
</realm>
```

JSON Representation

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Date: Tue, 15 Nov 2010 08:12:31 GMT

{
  "id": 1234,
  "name": "realmName",
  "description": "realmDescription",
  "key": "realmKey"
}
```

Errors

If the requested realm id is not an integer value -

XML Representation

```
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>
```

JSON Representation

```
HTTP/1.1 400 Bad Request
Content-type: application/json; charset=utf-8
Date: Tue, 15 Nov 2010 08:12:31 GMT
{
  "code": "InvalidArgument"
}
```

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

XML Representation

```
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>
```

JSON Representation

```
HTTP/1.1 404 Not Found
Content-type: application/json; charset=utf-8
Date: Tue, 15 Nov 2010 08:12:31 GMT
{
  "code": "RealmNotFound"
}
```

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.