PeopleFluent Learning

Rustici Engine Multi-database Configuration

For PeopleFluent Partners with a multi-tenant implementation

Contents

About Rustici Engine Multi-Database Configuration

This is a companion document to the Rustici Engine Installation Guide. It describes how to configure Rustici Engine to use a different database for each tenant in a multi-tenant implementation. Rustici Engine is PeopleFluent's solution for launching and tracking SCORM and AICC courses with PeopleFluent Learning.

Audience

This document is intended for PeopleFluent's partner companies, who wish to install Rustici Engine on separate databases in a multi-tenant implementation.

This document assumes that you are familiar with the supported operating systems, database servers, web browsers, and network configuration.

Support

Please direct all support queries regarding Rustici Engine to PeopleFluent Support:

- Case Management System: https://support.peoplefluent.com

- Customer Community: https://customers.peoplefluent.com

Notice to Users

This document is subject to revision based on external hardware and software changes; it may be updated periodically to reflect those changes.

PeopleFluent supports and provides defect fixes for PeopleFluent products under valid Support and Maintenance Agreements on only those operating systems and third-party systems that have been certified and published by PeopleFluent. PeopleFluent will not be responsible for providing any defect fixes for non-certified distributions.

Document Information

This section lists any changes or updates that occur following initial publication.

Table: Document Revision

Revision Information	
Revision Date:	June 17, 2022
Revised Document Version Number:	1.0
Details of Revision:	Initial publication.

Configuration Process Summary

You can configure Rustici Engine to use a different database for each tenant in a multi-tenant implementation. The configuration steps for each tenant are as follows:

1. Create the tenant database and credentials.

2. Build the database connection string.

3. Create the tenant in Rustici Engine.

4. Configure the tenant.

5. Install tables for tenant database.

6. Test the tenant's credentials.

7. Update ekp.properties.

Create the Tenant Database and Credentials

Before installing Rustici Engine, you must create its database and an admin user.

For example, to prepare a SQL Server database:

1. In SQL Server Management Studio, create a new database named **rustici**.

2. Create a new login named **rusticiadmin** with SQL Server authentication and non-expiring password.

3. Set the database role membership for the rusticiadmin user to db_owner:
   a. In the Object Explorer, expand **rustici > Security > Users**.

   b. Right-click **rusticiadmin** and select **Properties** from the context menu.

   c. In the Membership page, select the **db_owner** check box.

Build the Database Connection String

The connection string depends on the database type. Example connection strings for SQL Server and MySQL are shown below.

SQL Server
``jdbc:sqlserver: //
EngineHost;TenantDBName=rustici;user=TenantDBUser;password=TenantDBUserPassword|
com.microsoft.sqlserver.jdbc.SQLServerDriver``` |

MySQL
``jdbc:mysql: //EngineHost/TenantDBName?
user=TenantDBUser&password=TenantDBUserPassword|com.mysql.jdbc.Driver```

You must use the Rustici Engine API to create the tenant in Rustici Engine.

- Base URL: `EngineHost/api/v2/`

- API: `appManagement/tenants/{tenantName}`

Send a PUT request with the body:

```
{"active": true}
```

Rustici Engine uses Basic authentication in the header, with the ApiBasicSystemAccounts property in the RusticiEngineSettings.properties file 'username:password' base-64 encoded:

```
BASIC dXNlcm5hbWU6cGFzc3dvcmQ=
```

A successful request returns a 204 response.

See also the appManagement API documentation (CreateOrUpdateTenant).

Configure the Tenant

Using the API, add the database configuration information to the tenant:

- Base URL: `EngineHost/api/v2/`

- API: `appManagement/configuration`

Send a POST request with the body:

```
{
   "settings": [
      {
      "settingId": "TenantDatabaseConnectionString",
      "value": "{DatabaseConnectionString}"
      }
   ]
}
```

Where the DatabaseConnectionString comes from the earlier step.

See also the [appManagement API documentation](#) (SetApplicationConfiguration).

Rustici Engine uses Basic authentication in the header, with the ApiBasicSystemAccounts property in the RusticiEngineSettings.properties file "username:password" base-64 encoded:

```
BASIC dXNlcm5hbWU6cGFzc3dvcmQ=
```

Send this additional header:

```
engineTenantName: "{tenantName}"
```

A successful request returns a 204 response.

Install Tables for Tenant Database

To install the tables, run this command line on the server:

```
java -Dlogback.configurationFile=logback.xml -cp "lib/*"
RusticiSoftware.ScormContentPlayer.Logic.Upgrade.ConsoleApp <persistenceEngine>
<systemDatabaseConnectionString> <tenantDatabaseConnectionString> [schemaPrefix]
<tenantName> -tenant
```

Where:

- `<persistenceEngine>` is the name of your DBMS. The value should be either sqlserver, mysql, oracle, or postgresql.

- `<systemDatabaseConnectionString>` is the connection string for your system database.

- `<tenantDatabaseConnectionString>` is the connection string for the new tenant database.

- `[schemaPrefix]` is the name of the schema that Engine's tables should be created in. This value is optional. If not included, Engine will use the default schema for the user provided in `<systemDatabaseConnectionString>`.

- `<tenantName>` is the name of the tenant for this database. It should be unique, and is required.

Test the Tenant's Credentials

Use the Rustici Engine API to test the credentials:

- Base URL: `EngineHost/api/v2/`

- API: `courses`

- Query String Parameters:
    - `?includeRegistrationCount=true`

    - `&includeCourseMetadata=false`

Send a `GET` request with no body.

See also the appManagement API documentation (GetCourses).

Engine uses Basic authentication in the header, with the ApiBasicSystemAccounts property in the RusticiEngineSettings.properties file "username:password" base-64 encoded:

```
BASIC dXNlcm5hbWU6cGFzc3dvcmQ=
```

Send this additional header:

```
engineTenantName: "{tenantName}"
```

A successful request returns a 200 response.

Update ekp.properties

Follow the guidance to update the ekp.properties file with the following properties:

- rustici.uri=*URL for the Engine*

- rustici.username=*ApiBasicAccounts username*

- rustici.password=*ApiBasicAccounts password*

- rustici.enginetenantname={*tenantName*}

- rustici.importdefault=true

Legal Notice

This document has been created for authorized licensees and subscribers ("Customers") of the software products and associated services of Learning Technologies Group, Inc. by its division PeopleFluent and all of its affiliates (individually and collectively, as applicable, "PeopleFluent"). It contains the confidential and proprietary information of PeopleFluent and may be used solely in accordance with the agreement governing the use of the applicable software products and services. This document or any part thereof may not be reproduced, translated or retransmitted in any form without the written permission of PeopleFluent. The information in this document is subject to change without notice.

PEOPLEFLUENT DISCLAIMS ALL LIABILITY FOR THE USE OF THE INFORMATION CONTAINED IN THIS DOCUMENT AND MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO ITS ACCURACY OR COMPLETENESS. PEOPLEFLUENT DISCLAIMS ALL IMPLIED WARRANTIES INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. PEOPLEFLUENT DOES NOT GUARANTEE THAT ITS PRODUCTS OR SERVICES OR ANY SAMPLE CONTENT CONTAINED IN ITS PRODUCTS AND SERVICES WILL CAUSE OR ENABLE CUSTOMER TO COMPLY WITH LAWS APPLICABLE TO CUSTOMER. USERS ARE RESPONSIBLE FOR COMPLIANCE WITH ALL LAWS, RULES, REGULATIONS, ORDINANCES AND CODES IN CONNECTION WITH THE USE OF THE APPLICABLE SOFTWARE PRODUCTS,INCLUDING, WITHOUT LIMITATION, LABOR AND EMPLOYMENT LAWS IN RELEVANT JURISDICTIONS. THE PEOPLEFLUENT PRODUCTS AND SAMPLE CONTENT SHOULD NOT BE CONSTRUED AS LEGAL ADVICE.

Without limiting the generality of the foregoing, PeopleFluent may from time to time link to third-party websites in its products and/or services. Such third-party links are for demonstration purposes only, and PeopleFluent makes no representations or warranties as to the functioning of such links or the accuracy or appropriateness of the content located on such third-party sites. You are responsible for reviewing all content, including links to third-party web sites and any content that you elect to use, for accuracy and appropriateness, and compliance with applicable law.

Any trademarks included in this documentation may comprise registered trademarks of PeopleFluent in the United States and in other countries.

Microsoft, Windows, and Internet Explorer are trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Oracle and PeopleSoft are registered trademarks of Oracle International Corporation. Adobe and Acrobat are registered trademarks of Adobe Systems Incorporated. All other names are used for identification purposes only and are trademarks or registered trademarks of their respective owners. Portions of PeopleFluent Workforce Communication software may include technology licensed from Autonomy and are the copyright of Autonomy, Inc.Quartz Scheduler is licensed under the Apache License.

Website: peoplefluent.com