Asynchronous Logging API

The Asynchronous Logging API provides actions to perform the following asynchronously:

- insert records into the database
- register request events of your applications

The record or request event gets added to a message queue. Then the OutSystems log service processes it, and adds it to the database.

The message queue is non-persistent. This means that in case of a system failure, pending records and request events get lost.

To use this API, simply reference the AsynchronousLogging module in your application.

Summary

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogError</td>
<td>Asynchronously inserts a error into the database. Errors are kept in a message queue and inserted into the database in bulk after a short period.</td>
</tr>
<tr>
<td>LogRecord</td>
<td>Asynchronously inserts a record into the database. Records are kept in a message queue and inserted into the database in bulk after a short period.</td>
</tr>
<tr>
<td>LogRequestEvent</td>
<td>Asynchronously logs a request event. The events are kept in a message queue and inserted in bulk after a short period.</td>
</tr>
</tbody>
</table>

Actions

Updated: Thu, 02 Jan 2020 08:31:17 GMT
LogError

Asynchronously inserts an error into the database. Errors are kept in a message queue and inserted into the database in bulk after a short period. Note that the message queue is non-persistent.

**Inputs**

**Instant**
- Type: DateTime. Mandatory.
- Date and time when the error occurred.

**RequestKey**
- Type: Text.
- The GUID that identifies the request. Use the Request_GetKey action of the RuntimePlatform API to get this value.

**ModuleName**
- Type: Text. Mandatory.
- Name of the module where the event occurred.

**Message**
- Type: Text. Mandatory.
- A text with the error main message.

**Detail**
- Type: Text.
- A text with the error details.

LogRecord

Asynchronously inserts a record into the database. Records are kept in a message queue and inserted into the database in bulk after a short period. Note that the message queue is non-persistent.

**Inputs**

**Record**
- Type: Object. Mandatory.
- The record you want to save to the database, converted to Object type. Use the ToObject built-in function for the conversion.
LogRequestEvent

Asynchronously logs a request event. The events are kept in a message queue and inserted in bulk after a short period.

**Inputs**

**Instant**
- Type: DateTime. Mandatory.
  Date and time when the event occurred.

**RequestKey**
- Type: Text. Mandatory.
  The GUID that identifies the request. Use the Request_GetKey action of the RuntimePlatform API to get this value.

**RequestEventName**
- Type: Text. Mandatory.
  Name of the event.

**ModuleKey**
- Type: Text. Mandatory.
  Unique identifier of the module where the event occurred. Use the GetEspace action with the GetOwnerEspaceIdentifier() built-in function to get this information.

**ModuleName**
- Type: Text. Mandatory.
  Name of the module where the event occurred. Use the GetEspace action with the GetOwnerEspaceIdentifier() built-in function to get this information.

**ApplicationKey**
- Type: Text. Mandatory.
  Unique identifier of the application where the event occurred. Query the ‘Application’ system entity to get this information.

**ApplicationName**
- Type: Text. Mandatory.
  Name of the application where the event occurred. Query the ‘Application’ system entity to get this information.

**RequestEventDetails**
- Type: Text.
  A text with event details in JSON format. It is a regular JSON object with the fields 'Key' and 'Value'.

Updated: Thu, 02 Jan 2020 08:31:17 GMT