Contents

- Configuration
  - Item Viewer Configuration
    - General
    - Logging
  - Tomcat
- Item Viewer Service Manual Setup
  - Dependencies
    - Compile Time Dependencies
    - Run Time Dependencies
  - Configuration
    - Dictionary API
    - Aws Container Cluster Information (Optional)
    - Logging
    - Local System
  - Building and Running

* Building
* Running

Configuration

Item Viewer Configuration

General

The Item Viewer Service config file is located in `app/src/main/resources/settings-mysql.xml`
Most of the vlaues are carried over from Iris. The following options must be
configured in the settings-mysql.xml config file for the Item Viewer to function
correctly. - `iris.ContentPath` must be set to the location of the content
package on the local filesystem. - `iris.DictionaryUrl` must be set to the url
of the dictionary API. - `AwsRegion` Set this to the AWS region the Ite Viewer is
running in if it is running on AWS. - `AwsClusterName` Set this to the AWS ECS
cluster the Item Viewer Service is running in if it is running on AWS ECS.

Logging

The Item viewer service uses Logback Classic bound to SLF4J for logging. The
Logging configuration file is `app/src/main/resources/logback.xml`. The con-
fig file included will log to stdout and `/home/tomcat7/itemviewerservice.log`.
Details for configuring the log output can be found in the loback classic docu-
mentation.

Tomcat

The Item Viewer must be run in Apache Tomcat 7 or newer.

In order to run correctly the following Tomcat configuration needs to be set.

Set a 25 character alphanumeric numeric encryption key for Iris in $TOM-
CAT_HOME/conf/context.xml under the context element. The entry
follows the form ``.

The dictionary API call is made as a cross origin request. A CORS filter must
be added to `$TOMCAT_HOME/conf/web.xml`.

```
<filter>
  <filter-name>CorsFilter</filter-name>
  <filter-class>org.apache.catalina.filters.CorsFilter</filter-class>
</filter>
```

```xml
<filter-mapping>
  <filter-name>CorsFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

Item Viewer Service Manual Setup

To build and run the item viewer service on your local machine you will need the following.

Dependencies

Compile Time Dependencies

Compile time dependencies are built into the Maven POM file. Java 7 is required to build and run the Item Viewer Service.

The Item Viewer Service depends on the Iris package of TDS_Student.

Run Time Dependencies

- Apache Tomcat 7 or newer
- Smarter Balanced Dictionary API access
- Read access to the local file system

Configuration

The item viewer service configuration file is located at `app/src/main/resources/settings-mysql.xml`.

The `iris.ContentPath` variable in the settings-mysql.xml file needs to be set to the local directory where the content packages are going to be stored. If this directory does not exist, or the application can not access it, it will fail to launch.

Iris requires a 25 character alphanumeric numeric encryption key set as a parameter in $TOMCAT_HOME/conf/context.xml under the context element. The entry follows the form `<Parameter value="YOUR KEY ENCRYPTION KEY HERE" override="false" name="tds.iris.EncryptionKey"/>`.

Dictionary API

In order to use the dictionary you need to set the `iris.DictionaryUrl` value in the settings-mysql.xml config file for the Iris. The dictionary should be a running instance of the TDS_Dictionary application.

The dictionary API call is made as a cross origin request. A CORS filter needs to be added to the Tomcat $TOMCAT_HOME/conf/web.xml file.

```xml
<filter>
  <filter-name>CorsFilter</filter-name>
  <filter-class>org.apache.catalina.filters.CorsFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>CorsFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

Tomcat has detailed documentation on setting up CORS filtering. Please refer to it if you want to set up a more detailed filter.

Aws Container Cluster Information (Optional)

The "AwsRegion" and "AwsClusterName" keys are used if the application is being run outside of an AWS ERS cluster. If you are running the Item Viewer locally you can ignore these. Please note that the /status url will not work if the Item Viewer is being run outside of an AWS ECS cluster. Use /statusLocal for local diagnostics.

Logging

The item viewer service uses SLF4J bound to Logback Classic for logging. The log settings are found in logback.xml. For basic logging to a file you will need to set the file location for the file appender. For a full reference on configuring the log output levels and locations please refer to the Logback Classic documentation.

Local System

The application requires read permissions to the Iris content directory specified in settings-mysql.xml.

Building and Running

Building

To build the item viewer run `mvn install` in the top level project directory. The compiled WAR file will be generated in `app/target/itemviwerservice.war`.

Running

Deploy the WAR file to Apache Tomcat by placing the itemviewerservice.war file in your tomcat webapps directory, then starting Tomcat.

Configuration

Item Viewer Configuration

General

The Item Viewer Service config file is located in `app/src/main/resources/settings-mysql.xml` Most of the vlaues are carried over from Iris. The following options must be configured in the settings-mysql.xml config file for the Item Viewer to function correctly. - `iris.ContentPath` must be set to the location of the content package on the local filesystem. - `iris.DictionaryUrl` must be set to the url of the dictionary API. - `AwsRegion` Set this to the AWS region the Ite Viewer is running in if it is running on AWS. - `AwsClusterName` Set this to the AWS ECS cluster the Item Viewer Service is running in if it is running on AWS ECS.

Logging

The Item viewer service uses Logback Classic bound to SLF4J for logging. The Logging configuration file is `app/src/main/resources/logback.xml`. The config file included will log to stdout and `/home/tomcat7/itemviewerservice.log`. Details for configuring the log output can be found in the loback classic documentation.

Tomcat

The Item Viewer must be run in Apache Tomcat 7 or newer.

In order to run correctly the following Tomcat configuration needs to be set.

Set a 25 character alphanumeric numeric encryption key for Iris in $TOM-CAT_HOME/conf/context.xml under the context element. The entry follows the form `<Parameter value="YOUR KEY ENCRYPTION KEY HERE" override="false" name="tds.iris.EncryptionKey"/>`.

The dictionary API call is made as a cross origin request. A CORS filter must be added to `$TOMCAT_HOME/conf/web.xml`.

```
<filter>
  <filter-name>CorsFilter</filter-name>
```

```
    <filter-class>org.apache.catalina.filters.CorsFilter</filter-class>
</filter>
<filter-mapping>
    <filter-name>CorsFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

Item Viewer Service Manual Setup

To build and run the item viewer service on your local machine you will need the following.

Dependencies

Compile Time Dependencies

Compile time dependencies are built into the Maven POM file. Java 7 is required to build and run the Item Viewer Service.

The Item Viewer Service depends on the Iris package of TDS_Student.

Run Time Dependencies

- Apache Tomcat 7 or newer
- Smarter Balanced Dictionary API access
- Read access to the local file system

Configuration

The item viewer service configuration file is located at `app/src/main/resources/settings-mysql.xml`.

The `iris.ContentPath` variable in the settings-mysql.xml file needs to be set to the local directory where the content packages are going to be stored. If this directory does not exist, or the application can not access it, it will fail to launch.

Iris requires a 25 character alphanumeric numeric encryption key set as a parameter in $TOMCAT_HOME/conf/context.xml under the context element. The entry follows the form `<Parameter value="YOUR KEY ENCRYPTION KEY HERE" override="false" name="tds.iris.EncryptionKey"/>`.

Dictionary API

In order to use the dictionary you need to set the `iris.DictionaryUrl` value in the settings-mysql.xml config file for the Iris. The dictionary should be a running instance of the TDS_Dictionary application.

The dictionary API call is made as a cross origin request. A CORS filter needs to be added to the Tomcat $TOMCAT_HOME/conf/web.xml file.

```xml
<filter>
  <filter-name>CorsFilter</filter-name>
  <filter-class>org.apache.catalina.filters.CorsFilter</filter-class>
</filter>
<filter-mapping>
  <filter-name>CorsFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

Tomcat has detailed documentation on setting up CORS filtering. Please refer to it if you want to set up a more detailed filter.

Aws Container Cluster Information (Optional)

The "AwsRegion" and "AwsClusterName" keys are used if the application is being run outside of an AWS ERS cluster. If you are running the Item Viewer locally you can ignore these. Please note that the /status url will not work if the Item Viewer is being run outside of an AWS ECS cluster. Use /statusLocal for local diagnostics.

Logging

The item viewer service uses SLF4J bound to Logback Classic for logging. The log settings are found in logback.xml. For basic logging to a file you will need to set the file location for the file appender. For a full reference on configuring the log output levels and locations please refer to the Logback Classic documentation.

Local System

The application requires read permissions to the Iris content directory specified in settings-mysql.xml.

Building and Running

Building

To build the item viewer run `mvn install` in the top level project directory. The compiled WAR file will be generated in `app/target/itemviwerservice.war`.

Running

Deploy the WAR file to Apache Tomcat by placing the itemviewerservice.war file in your tomcat webapps directory, then starting Tomcat.