Installing and configuring the CAST-RestAPI.war

On this page:

Target audience:

CAST AI Administrators

ⓘ **Summary:** This section describes how to install and configure the **CAST-RestAPI.war** - see the CAST APIs Guide for more information.

Requirements

Installing the **CAST-RestAPI.war** requires the following:

✅	**Java JRE**	Must be installed on the machine that will host the Application Server. Please see Supported Platforms about JRE support.
✅	**Application Server**	Must be installed on a compatible machine. Please see Supported Platforms about Application Server support.
✅	**Dashboard Service or Measurement Service**	A **single** CAST Dashboard Service or Measurement Service or **multiple** CAST Dashboard Services / Measurement Services. To ensure compatibility, except when working with the CAST Report Generator, you must only use a CAST Dashboard Service or Measurement Service installed with a version of CAST AIP that exactly matches the major, minor or Service Pack release number of the CAST RestAPI WAR that you are installing. The RestAPI can interact with CAST Dashboard Services or Measurement Services installed on all supported host CSS2 /RDBMS. ⓘ When using the CAST-RestAPI.war with the CAST Report Generator, CAST highly recommends configuring connections to the **CAST Dashboard Service** ONLY.
✅	**WAR file**	The **CAST-RestAPI.war** can be retrieved from the **WARS** folder located in the CAST installation folder: 📁 deprecated 📄 CAST-AAD.war 📄 CAST-AAD-AED.war 📄 CAST-AED.war 📄 CAST-AICPortal.war 📄 CAST-CED.ear 📄 CAST-CED.war 📄 CAST-RESTAPI.war
✅	**License Key**	Access to data in the CAST Dashboard Services is governed by a **license key** - see CAST-AED-RESTAPI - Dashboard Service license key configuration. You must have a valid license key in your possession before you begin if you wish to configure access to a CAST Dashboard Service - for example when using the CAST Report Generator.

Pre-installation check list

Before beginning the installation process, please ensure that you have carried out all of the following tasks and that the following requirements have been met:

✅	Ensure you have read all Release Notes accompanying CAST products for any last-minute information.
✅	Decide where the CAST-RESTAPI.war will be installed - CAST Administration workstation, separate dedicated machine or existing machine used for the CAST Application Analytics or Engineering Dashboards/CAST AIC Portal.
✅	Ensure that your user login on the target machine has sufficient user privileges to install applications (see Appendix - CAST Administration workstation installation and hardware requirements)
✅	Make sure you have the required .WAR file ready for deployment

Installation procedure

> ℹ️ This document assumes you are using **Apache Tomcat** in a **Windows environment** to host the CAST-RESTAPI. If you are using **IBM Websphere Application Server** in a **Linux** environment, please see Appendix - AAD - AED - RestAPI - Deployment on IBM WebSphere Application Server.

The installation process is divided into various steps:

Step 1	Deploy and configure the CAST-RESTAPI.war file on your application server
Step 2	Configure access security
Step 3	Upload snapshot data for display
Step 4	Configure data authorization

Step 1 - deploy and configure the CAST-RestAPI.war file on your application server

Deploy the WAR file

Locate the **CAST-RestAPI.war** file and move it to the web application installation location. By default on Apache Tomcat this is set to:

```
%CATALINA_HOME%\webapps
```

- If the Apache Tomcat server is running, the .WAR file will then be unpacked and deployed in a folder called "**CAST-RESTAPI**".

Tomcat 7 - configure connections to a single Dashboard / Measurement Service on CSS

If you are using **Tomcat 7** to host the CAST RestAPI and **CSS** to host your CAST Dashboard/Measurement Services, please follow these instructions:

- Once deployed you now need to configure the **context.xml** file to tell the web application where the Dashboard Service / Measurement Service are stored. This file is located here:

```
%CATALINA_HOME%\webapps\CAST-RESTAPI\META-INF\context.xml
```

- Open this file with a Text editor. By default, the file will contain an **uncommented** template for **CSS2**.
- If your Dashboard Service or Measurement Service is located on CSS2, you can use the uncommented templates for CSS2 - this contains two templates:

```
<Resource name="jdbc/domains/AAD" url="jdbc:postgresql://localhost:2280/postgres"
    initConnectionSqls="SET search_path TO [Measure Schema];"
    username="operator" password="CastAIP"

    auth="Container" type="javax.sql.DataSource" driverClassName="org.postgresql.Driver"
    validationQuery="select 1"
    initialSize="5" maxActive="20" maxIdle="10" maxWait="-1"/>

<Resource name="jdbc/domains/AED" url="jdbc:postgresql://localhost:2280/postgres"
    initConnectionSqls="SET search_path TO [Central Schema];"
    username="operator" password="CastAIP"

    auth="Container" type="javax.sql.DataSource" driverClassName="org.postgresql.Driver"
    validationQuery="select 1"
    initialSize="5" maxActive="20" maxIdle="10" maxWait="-1"/>
```

- First comment one of the templates by adding **<!--** and **-->** around the section.
- Change the **name** parameter to something other than "**jdbc/domains/AAD**" or "**jdbc/domains/AED**" - CAST recommends using the name of your target Dashboard Service or Measurement Service - for example "**jdbc/domains/V80_CENTRAL_CSS2**"
- Set the **URL** parameter to the server on which the **Dashboard Service** or **Measurement Service** is located. Usually you just need to alter the "localhost" to the name of the server on which the host RDBMS is located.
- Change **[Measure Schema]** or **[Dashboard Schema]** to the name of the **Dashboard Service** or **Measurement Service**.
- Ensure the **password** parameter is correct
- You should end up with a section like this - this is for a **CSS2 connection**:

```
<Resource name="jdbc/domains/V80_CENTRAL_CSS2" url="jdbc:postgresql://NEFYN:2280/postgres"
        initConnectionSqls="SET search_path TO V80_CENTRAL_CSS2;"
        username="operator" password="CastAIP"

        auth="Container" type="javax.sql.DataSource" driverClassName="org.postgresql.Driver"
        validationQuery="select 1"
        initialSize="5" maxActive="20" maxIdle="10" maxWait="-1"/>
```

- Following any changes you make, **save the context.xml file** and then **restart** your application server so that the changes are taken into account.

ⓘ Note that if your Dashboard / Measurement Service is located on CSS1, you need to alter the port number from 2280 (CSS2) to 2278 (CSS1).

Tomcat 8 / 8.5 - configuration for a single Dashboard / Measurement Service on CSS

If you are using **Tomcat 8** or **8.5** to host the CAST RestAPI and **CSS** to host your CAST Dashboard/Measurement Services, please follow these instructions:

- Open the **context.xml file** (as detailed above) with a Text editor.
- By default, the file will contain an **uncommented** template as follows - this is tailored for connections to **CAST schemas** located on **CAST Storage Service 2** using **Tomcat 7**:

```
<Resource name="jdbc/domains/AAD" url="jdbc:postgresql://localhost:2280/postgres"
    initConnectionSqls="SET search_path TO [Measure Schema];"
    username="operator" password="CastAIP"

    auth="Container" type="javax.sql.DataSource" driverClassName="org.postgresql.Driver"
    validationQuery="select 1"
    initialSize="5" maxActive="20" maxIdle="10" maxWait="-1"/>

<Resource name="jdbc/domains/AED" url="jdbc:postgresql://localhost:2280/postgres"
    initConnectionSqls="SET search_path TO [Central Schema];"
    username="operator" password="CastAIP"

    auth="Container" type="javax.sql.DataSource" driverClassName="org.postgresql.Driver"
    validationQuery="select 1"
    initialSize="5" maxActive="20" maxIdle="10" maxWait="-1"/>
```

- You need to comment this template by adding **<!--** and **-->** around the section:

```
<!--
<Resource name="jdbc/domains/AAD" url="jdbc:postgresql://localhost:2280/postgres"
    initConnectionSqls="SET search_path TO [Measure Schema];"
    username="operator" password="CastAIP"

    auth="Container" type="javax.sql.DataSource" driverClassName="org.postgresql.Driver"
    validationQuery="select 1"
    initialSize="5" maxActive="20" maxIdle="10" maxWait="-1"/>

<Resource name="jdbc/domains/AED" url="jdbc:postgresql://localhost:2280/postgres"
    initConnectionSqls="SET search_path TO [Central Schema];"
    username="operator" password="CastAIP"

    auth="Container" type="javax.sql.DataSource" driverClassName="org.postgresql.Driver"
    validationQuery="select 1"
    initialSize="5" maxActive="20" maxIdle="10" maxWait="-1"/>
-->
```

- Now locate the **Tomcat 8** template for **CAST Storage Service 2** - this is located at the top of the following section:

```
<!--  For Tomcat 8 : http://tomcat.apache.org/tomcat-8.0-doc/jndi-datasource-examples-howto.html &
http://commons.apache.org/proper/commons-dbcp/configuration.html

template for resources on CSS :
<Resource name="jdbc/domains/${domainName}"
        url="jdbc:postgresql://${host}:${port}/postgres"
        connectionInitSqls="SET search_path TO ${schema};"
        username="${user}" password="${password}"

        auth="Container" type="javax.sql.DataSource" driverClassName="org.postgresql.Driver"
        validationQuery="select 1" initialSize="5" maxTotal="20" maxIdle="10" maxWaitMillis="-1"/>

template for resources on Oracle :
<Resource name="jdbc/domains/{domainName}"
        url="jdbc:oracle:thin:@{server}:{port}:{instance}"
        connectionInitSqls="ALTER SESSION SET CURRENT_SCHEMA={schema}"
        username="{user}"    password="{password}"

        auth="Container" type="javax.sql.DataSource" driverClassName="oracle.jdbc.OracleDriver"
        validationQuery="select 1 from DUAL" initialSize="5" maxTotal="20" maxIdle="10" maxWaitMillis="-1" />

template for resources on MS SQL Server :
<Resource name="jdbc/domains/{domainName}"
        url="jdbc:jtds:sqlserver://{host};instance={instance}"
        connectionInitSqls="use {database}"
        username="{user}" password="{password}"

        auth="Container" type="javax.sql.DataSource" driverClassName="net.sourceforge.jtds.jdbc.Driver"
        validationQuery="select 1" initialSize="5" maxTotal="20" maxIdle="10" maxWaitMillis="-1"/>
-->
```

- Copy the **Tomcat 8 for CSS2** template into **uncommented white space**:

```
<Resource name="jdbc/domains/${domainName}"
        url="jdbc:postgresql://${host}:${port}/postgres"
        connectionInitSqls="SET search_path TO ${schema};"
        username="${user}" password="${password}"

        auth="Container" type="javax.sql.DataSource" driverClassName="org.postgresql.Driver"
        validationQuery="select 1" initialSize="5" maxTotal="20" maxIdle="10" maxWaitMillis="-1"/>
```

- Change the **${domainName}** parameter to the name of your target Dashboard Service or Measurement Service - for example "**jdbc/domains /V80_CENTRAL_CSS2**"
- Set the **URL** parameter to the server on which the **Dashboard Service** or **Measurement Service** is located (usually you just need to alter the **${h ost}** and **${port}** values to the name/port of the server on which the host CSS is located. Remember that CSS1 uses port 2278 and CSS2 uses port 2280.

- Change **${schema}** to the name of the **Dashboard Service** or **Measurement Service** (see Install CAST schemas).
- Ensure the **${user}** and **${password}** parameters are correct - use the **operator** and **CastAIP** username/password combination (note that if you need to **encrypt** the username and password to avoid entering names in clear text, please see: CAST-AAD-AED - Encrypt login and password for datasource and Active Directory)
- You should end up with a section like this:

```
<Resource name="jdbc/domains/V80_CENTRAL_CSS2"
       url="jdbc:postgresql://NEFYN:2280/postgres"
       connectionInitSqls="SET search_path TO V80_CENTRAL_CSS2;"
       username="operator" password="CastAIP"

       auth="Container" type="javax.sql.DataSource" driverClassName="org.postgresql.Driver"
       validationQuery="select 1" initialSize="5" maxTotal="20" maxIdle="10" maxWaitMillis="-1"/>
```

- Following any changes you make, **save the context.xml file** and then **restart** your application server so that the changes are taken into account.

Dashboard / Measurement Services located on a Microsoft SQL Server or Oracle Server for Tomcat 7/8 /8.5

If your target Dashboard Service is located on a Microsoft SQL Server or Oracle Server, you need to:

- Remove or re-comment the **uncommented** template for **CSS2/Tomcat 7** from the **context.xml** file
- Copy the **commented** template for Microsoft SQL Server or Oracle into uncommented space - ensure you chose the correct template for **Tomcat 7** or **Tomcat 8 / 8.5**.
- Make the same configuration changes as suggested above:
    - First change the **{domainName}** parameter from **name="jdbc/domains/{domainName}"** - CAST recommends using the name of your target Dashboard Service or Measurement Service - for example **name="jdbc/domains/V80_CENTRAL_CSS2"**
    - Set the **URL** parameter to the server on which the **Dashboard Service** or **Measurement Service** is located. Please see http://jtds. sourceforge.net/faq.html#urlFormat for more information about the Microsoft SQL Server jTDS JDBC driver URL format.
    - Set the **initConnectionSqls** parameter to the name of the **Dashboard Service** or **Measurement Service** {database} or {schema}
    - Ensure the **username** and **password** parameters are correct - these must provided access the to **Dashboard Service** or **Measurement Service** database or schema.
- Following any changes you make, **save the context.xml file** and then **restart** your application server so that the changes are taken into account.

Configuring connections to multiple Dashboard / Measurement Services for Tomcat 7/8/8.5

The above information describes how to configure a connection to **one single** Dashboard Service or Measurement Service. It is also possible to configure connections to **multiple** Dashboard or Measurement Services, simply by adding in an uncommented template for each target Dashboard or Measurement Service.

You can mix Dashboard Services with Measurement Services and/or you can mix Services stored on legacy releases of CAST AIP starting 7.0.x and with the CAST RestAPI stored on Tomcat 7 or 8 / 8.5. Following any changes you make, **save the context.xml file** and then **restart** your application server so that the changes are taken into account.

> ⓘ Please ensure that each template is configured with a unique **{domainName}** parameter from **name="jdbc/domains/{domainName}"**.

Connection pooling

It is possible to modify the connection configuration for the CAST Application Engineering Dashboard to pool multiple CAST Dashboard Service schemas /databases in one resource "domain". In other words, you no longer have to configure one resource (in the context.xml) per CAST Dashboard Service "schema/database", instead, one resource (in the context.xml) can function for multiple CAST Dashboard Service schemas/databases. Please see Appendix - CAST Dashboard Service schema connection configuration pooling for more information.

Install the license key

As explained in CAST-AED-RESTAPI - Dashboard Service license key configuration, when you want to access a **CAST Dashboard Service** using the **CAST RESTAPI** (i.e. via the the **CAST Report Generator**), a special **license key** is required. This license key grants specific access to one or multiple CAST Dashboard Services for the web application in which it is installed (i.e. the CAST RestAPI).

You must therefore install the license key and, if you are using a restricted license key, define data access authorization. These two steps are explained in CAST-AED-RESTAPI - Dashboard Service license key configuration in the sections **How do I install a license key?** and **How to authorize users when using a RESTRICTED license key**.

Step 2 - configure access security

Please see either:

- CAST-AAD - Configuring user authentication
- CAST-AED - Configuring user authentication

The configuration is identical for the CAST RestAPI.

Step 3 - upload snapshot data for display

Before your users can exploit data via the CAST RestAPI, you need to generate snapshot data and (if accessing a Measurement Service) ensure the snapshot data is uploaded to the Measurement Service. This is achieved using the **CAST Management Studio**. See 2.2.3. Snapshot generation and validation for more information.

Step 4 - Configure data authorization

An **Authorization** defines permission to access and "consume the data" in a specific Application via the CAST RestAPI. If permission is not granted then any information related to this Application will be not accessible: application properties such as name, technologies or grades and measures, etc. Therefore, an Authorization must be defined before a user/group of users can access a specific application.

 Please see either:

- CAST-AAD - Configuring data authorization
- CAST-AED - Configuring data authorization

The configuration is identical for the CAST RestAPI.