Uruguay Presidencia | agesic

Manual de Desarrollo SAE 2.x.x

SISINFO

Versión 1.1.2

2021

Histórico de cambios

Versión	Fecha	Modificado por	Descripción
1.0	07/07/2021	Fabricio Gregorio	Creación de Documento
1.1	06/09/2021	Sergio Coronel	
1.1.2	04/04/2022	Juan Moreno	Configuración para uso de TLS 1.2 en Desarrollo

Contenido

1 Introducción

El siguiente documento es una guía para la configuración del entorno de desarrollo del proyecto SAE en sus versiones 2.x.x. La guía describe la lista de elementos necesarios para instalar y como se debe proceder para compilar el código fuente.

2 Audiencia

Desarrolladores de software o administradores de sistemas encargados de instalar y configurar SAE.

3 Instalación

Pre-requisitos

Antes de instalar el entorno de desarrollo es necesario instalar los siguientes componentes:

- JDK 7
- JBoss AS versión 7.1.1
- Apache Ant 1.9 (o superior)

Código fuente

La aplicación SAE está conformada por los siguientes proyectos:

Proyecto	**Descripción**
SAE-Acciones-EJBClient	Contiene las clases base para implementar acciones personalizadas que pueden asociarse a los recursos.
SAE-Autocompletados-EJBClient	Contiene componentes para implementar autocompletados. No se utiliza en SAE v. 2.x.
SAE-Componentes	Contiene clases de utilidades. No se utiliza en SAE v. 2.x.x.
SAE-Config	Contiene archivos básicos de configuración y librerías utilizadas por otros módulos. En particular contiene al archivo build.xml que es utilizado por Ant para compilar y construir los archivos ejecutables.
SAE-EAR	Este proyecto integra los componentes del proyecto y genera el EAR desplegable.
SAE-EJB	Este proyecto contiene los servicios locales y externos correspondientes a la capa del negocio, tanto para el frontend como para el backend. Se corresponde con el archivo generado "sae-1-service.ear".
SAE-EJBClient	Este proyecto contiene las entidades, enumerados y utilidades.
SAE-Profile-Default-EAR	Proviene de la versión original de SAE y no se utiliza en SAE v. 2.x.x

SAE-Profile-Default-EJB	Contiene clases de utilidades para las validaciones y las acciones personalizadas.
SAE-Profile-Default-EJBClient	Contiene clases relacionadas a los servicios web para ser usadas en los clientes que requieran invocarlos. No se utiliza en SAE v. 2.x.x
SAE-Profile-Default-WEB	Contiene recursos estáticos de presentación para construir el perfil por defecto de la presentación de SAE. No se utiliza en SAE v. 2.x.
SAE-Validaciones-EJBClient	Contiene las clases base para implementar validaciones personalizadas que pueden asociarse a los recursos a la hora de crear una reserva.
SAE-WEB	Este proyecto corresponde a la interfaz de usuario de la agenda. Incluye cada una de las páginas (jsf y fragmentos), la lógica de presentación (managed bean) y la integración con la capa de negocio, tanto para el frontend como para el backend. Se corresponde con los archivos "sae-2-frontend.ear", "sae-2-backoffice.ear" y "sae-1-recursos.ear".

Además, los siguientes dos proyectos también son requeridos por la aplicación pero no forman parte de ella sino que se utilizan como librerías adicionales en el servidor de aplicaciones JBoss AS:

- SAE-CDAServiceProvider
- SAE-LoginModule

Entorno de desarrollo – Ant/Eclipse

Los proyectos que componen la aplicación son compilados usando la herramienta Apache Ant, y proveen un script build.xml para dicha tarea. La compilación puede ser realizada directamente en la línea de comandos o puede utilizarse el entorno de desarrollo integrado Eclipse IDE.

Compilación usando Apache Ant en la línea de comandos

Para compilar cualquiera de los proyectos desde la línea de comandos se debe tener instalado y configurado Apache Ant versión 1.9 o superior. Para verificar si está instalado, y en caso afirmativo determinar su versión, ejecutar el siguiente comando:

```
$> ant -version
```

Debería verse un mensaje similar a “Apache Ant(TM) version 1.9.2 compiled on July 8 2013”; aunque la versión y la fecha podrían cambiar, es necesario que la versión sea 1.9 o posterior.

Si Apache Ant está correctamente instalado se puede proceder a compilar la aplicación, para lo cual se debe realizar lo siguiente:

1. Copiar la carpeta que contiene el código fuente del proyecto a una ubicación apropiada; si en lugar de una carpeta se tiene un archivo comprimido (.zip, .rar o .tar.gz), descomprimirlo en la ubicación elegida.
2. Abrir una consola y cambiar a la ubicación elegida en el punto 1 usando comandos “cd”.
3. Definir las siguientes dos variables de entorno:
   - JAVA_HOME: debe apuntar al directorio donde se encuentra la instalación del entorno de desarrollo JDK 7 (para verificar si es el directorio correcto, dentro de la ruta apuntada por esta variable debería verse una carpeta llamada bin y dentro de ésta archivos llamados “javac” o “javac.exe”).
   - ANT_HOME: debe apuntar al directorio donde se encuentra la instalación de Apache Ant (para verificar si es el directorio correcto, dentro de la ruta apuntada por esta variable deberían verse las carpetas “bin” y “etc”).
4. Ejecutar los siguientes comandos:
   - $> ant -f SAE-Config/build.xml clean
   - $> ant -f SAE-Config/build.xml build
5. Aguardar a que termine la compilación. Al finalizar debería verse el siguiente texto: “BUILD SUCCESSFUL”.
6. Verificar que los archivos EAR se generaron correctamente en la carpeta SAE-EAR/build/jar:
   - sae-1-recursos.ear
   - sae-1-service.ear
   - sae-2-backoffice.ear
   - sae-2-frontend.ear

Una vez que la aplicación principal fue compilada correctamente, y los cuatro archivos EAR que la componen generados, se puede proceder a compilar los proyectos auxiliares (SAE-CDAServiceProvider y SAE-LoginModule). Para esto, se debe realizar lo siguiente (para ambos proyectos el procedimiento es el mismo y debe repetirse para cada uno):

1. Entrar a la carpeta que contiene el código fuente del proyecto y copiar el archivo build.xml.template con el nombre build.xml.
2. Abrir el archivo build.xml y configurar la propiedad “jboss.home” para que apunte al directorio de instalación de JBoss AS 7 (buscar el texto <property name="jboss.home" value=""/>). Tener en cuenta que no es necesario que apunte al JBoss donde se instará finalmente la aplicación, sino que esta configuración es para que al momento de compilar se pueda acceder a las librerías necesarias.

3. Abrir una consola y cambiar a la misma carpeta utilizada en el punto 1 usando comandos “cd”.
4. Definir las siguientes dos variables de entorno:
    - JAVA_HOME: debe apuntar al directorio donde se encuentra la instalación del entorno de desarrollo JDK 7 (para verificar si es el directorio correcto, dentro de la ruta apuntada por esta variable debería verse una carpeta llamada bin y dentro de ésta archivos llamados javac o javac.exe).
    - ANT_HOME: debe apuntar al directorio donde se encuentra la instalación de Apache Ant (para verificar si es el directorio correcto, dentro de la ruta apuntada por esta variable deberían verse las carpetas bin y etc).
5. Ejecutar los siguientes comandos:
    - $> ant -f build.xml clean
    - $> ant -f build.xml build
6. Aguardar a que termine la compilación. Al finalizar debería verse el siguiente texto: “BUILD SUCCESSFUL”
7. Verificar que el archivo JAR se generó correctamente dentro de la carpeta “dist”, en la misma carpeta utilizada en el punto 1.

Compilación usando el entorno de desarrollo Eclipse IDE.

Si se desea compilar utilizando el entorno de desarrollo Eclipse IDE se recomienda utilizar la versión 4.4 (Eclipse Luna). Aunque no existen impedimentos para utilizar otra versión, el desarrollo fue realizado utilizando dicha versión y se sabe que el procedimiento descrito en este documento funciona con ella. Puede obtenerse un instalador apropiado seleccionando la opción “Eclipse IDE for Java EE Developers” en el sitio web https://eclipse.org/downloads/packages/release/luna/sr2. Una vez descargado el paquete instalador, se debe descomprimirlo en la ubicación que se considere apropiada (evitando espacios en la ruta). Luego, el IDE se puede ejecutar invocando el programa “eclipse” dentro de la carpeta anterior. Si al iniciar eclipse se observa el mensaje “java was started but returned exit code=13” el problema se debe a que la versión de Eclipse IDE no se corresponde con la versión de Java (32 bits o 64 bits) instalada en el sistema.

Nota: si hay más de una versión de Java instalada en el sistema (por ejemplo, Java 6 o Java 8) se debe instruir a Eclipse IDE de forma explícita la versión de java a utilizar; para ello, en el archivo “eclipse.ini”, que se encuentra en la misma carpeta se debe añadir la siguiente línea: “-vm {path}/bin/java” (en Linux) o “-vm
{path}\bin\javaw.exe” (en Windows), donde {path} debe ser sustituido por la ruta completa al directorio de instalación de Java 7.

Una vez que instalado e iniciado Eclipse IDE se debe abrir todos los proyectos que componen la aplicación SAE. Para esto se debe utilizar la opción de importar proyectos existentes, a la cual

se accede seleccionando “File→ Import→ General→ Existing Projects into Workspace”. Luego, en el campo “Select root directory” se debe seleccionar la carpeta que contiene a los proyectos de SAE y luego en el área “Projects” seleccionar todos los proyectos mostrados. Finalmente se debe hacer clic en el botón “Finish” para proceder a la importación de los proyectos.

Para compilar todos los proyectos de la aplicación principal se debe seleccionar el proyecto SAE-Config, desplegando la rama correspondiente, y realizar clic derecho sobre el archivo “build.xml”. Luego, en el cuadro de diálogo que se muestra seleccionar la opción “Run As” y finalmente “Ant build”. Al finalizar el proceso de compilación, en la parte inferior derecha, en la ventana titulada “Console” debería verse el mensaje “BUILD SUCCESSFUL”. Los archivos generados son colocados en la carpeta SAE-EAR/build/jar:

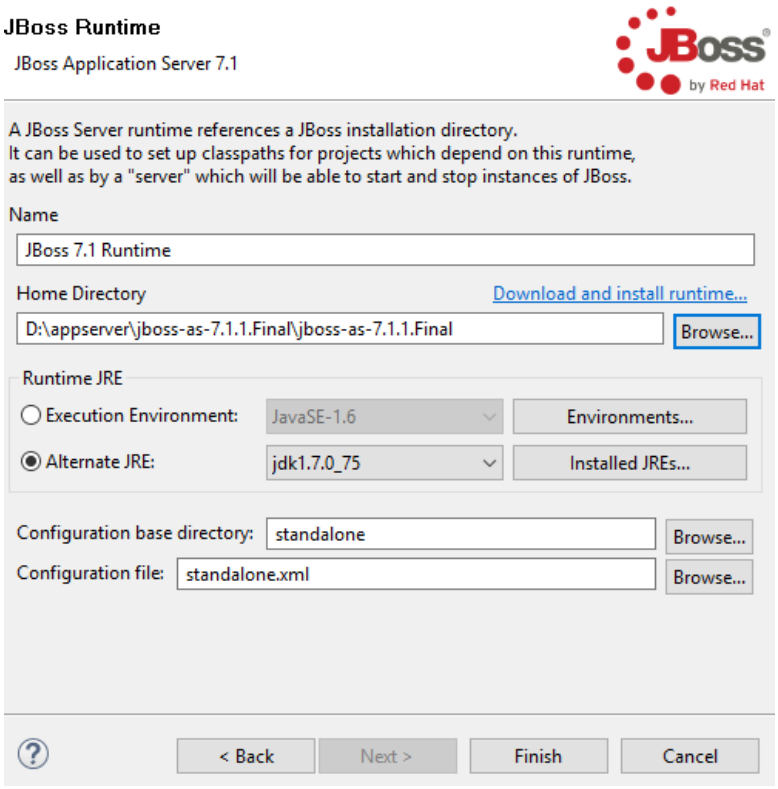
- sae-1-recursos.ear
- sae-1-service.ear
- sae-2-backoffice.ear
- sae-2-frontend.ear

Una vez que la aplicación principal fue compilada correctamente, y los cuatro archivos EAR que la componen generados, se puede proceder a compilar los proyectos auxiliares (SAE-CDAServiceProvider y SAE-LoginModule). Para esto, se debe realizar lo siguiente (para ambos proyectos el procedimiento es el mismo y debe repetirse para cada uno):

1. Copiar el archivo build.xml.template con el nombre build.xml.
2. Abrir el archivo build.xml y configurar la propiedad “jboss.home” para que apunte al directorio de instalación de JBoss AS 7 (buscar el texto <property name="jboss.home" value=""/>). Tener en cuenta que no es necesario que apunte al JBoss donde se instará finalmente la aplicación, sino que esta configuración es para que al momento de compilar se pueda acceder a las librerías necesarias.
3. Hacer clic derecho sobre el archivo “build.xml”, seleccionar la opción “Run As” y finalmente “Ant build”. Al finalizar el proceso de compilación, en la parte inferior derecha, en la ventana titulada “Console” debería verse el mensaje “BUILD SUCCESSFUL”. Los archivos generados son colocados en la carpeta “dist” dentro de la carpeta del proyecto.

Modificación del código fuente con Eclipse IDE

Para la compilación y generación de los binarios se utiliza ANT si lo que se desea es generar los binarios a partir del código fuente solo con ANT es suficiente y los mismos se generan en \SAE-EAR\build\jar. Sin embargo, si se quiere modificar el código fuente es necesario que eclipse reconozca los proyectos de forma correcta. Para esto debe indicar a Eclipse IDE dónde se encuentra la instalación de JBoss 7. Para esto, desde la opción Windows-→ Preferences → Server-→ Runtime Environments se debe seleccionar la opción “Add”; luego seleccionar “Jboss 7.1 Runtime”, hacer clic en “Next”; en el formulario que se muestra ingresar un nombre y especificar la ruta absoluta a la instalación de JBoss 7. Presionar el botón “Finish” para terminar.



Configuración de herramientas adicionales que utiliza el sistema informático

Está configuración ayuda y agilizan la preparación del ambiente de desarrollo, pero también se puedo optar por la opción convencional en donde tanto la base de datos como el servidor de aplicaciones se instalan localmente como servicios en las máquinas de cada desarrollador.

Docker

Automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos. Guía de instalación:

Ubuntu:

Seguir pasos de instalación en el siguiente link: https://docs.docker.com/engine/install/ubuntu/

Windows 10:

Seguir pasos de instalación de Docker Desktop en el siguiente link: https://docs.docker.com/docker-for-windows/install-windows-home/

Instrucciones para instalar ambiente e iniciar aplicación

1. Instalar última versión estable desde el siguiente enlace:

https://www.docker.com/products/docker-desktop

2. Clonar repositorio del código fuente de SAE v2 (incluye archivos de Docker):

3. Moverse al directorio de los archivos de configuración de Docker y ejecutar el siguiente comando: docker compose up

Nota: La primera vez tomará un tiempo para descargar y construir las imágenes.

Instrucciones para uso de TLS 1.2

Desde abril de 2022, el uso de versiones de TLS menores a 1.2 ha sido bloqueada por lo diferentes navegadores del mercado[1], si bien para producción se publica SAE a través de un proxy reverso o un WAF que se encarga de esto, en desarrollo la versión de Java 7u80 de Oracle no soporta dicho protocolo. Por ello para el ambiente de desarrollo se propone usar el siguiente procedimiento.

1. Usar la versión de Java 7 de Azul Systems, la versión 7u332b01 (https://www.azul.com/downloads/?version=java-7-lts&package=jdk)

[1] https://duo.com/decipher/browsers-will-block-sites-using-old-versions-of-tls-in-march

2. Actualizar el jboss-modules.jar que se encuentra en la raíz de la ruta de instalación del jboss-as-7.1.1.Final a la versión jboss-modules-1.1.5 (https://mvnrepository.com/artifact/org.jboss.modules/jboss-modules/1.1.5.GA). Cuando se reemplace se tiene que usar el mismo nombre esto es jboss-modules.jar.
3. Actualizar la entrada <ssl password="agesic" certificate-key-file="${jboss.server.config.dir}/sae.jks" protocol="**TLSv1.0**"/> del standalone.xml a <ssl password="agesic" certificate-key-file="${jboss.server.config.dir}/sae.jks" protocol="**TLSv1.2**"/>