

### Week 2 Unit 1

- 00:00:05 Hola, y bienvenido al curso de openSAP Creación de aplicaciones con el modelo de programación de aplicaciones RESTful ABAP. Semana dos, unidad uno, El escenario empresarial.
- 00:00:15 Me llamo Carine Tchoutouo Djomo y soy gestor de productos para la plataforma ABAP. Te guiaré por las unidades de esta semana.
- 00:00:23 En la semana anterior, aprendió sobre el panorama general y la arquitectura de RAP, y sobre las tecnologías y herramientas involucradas. También aprendió sobre el entorno práctico que utilizaremos durante este curso, el entorno ABAP de SAP Cloud Platform.
- 00:00:40 Además, le mostramos cómo configurar su IDE local, las herramientas de desarrollo ABAP en Eclipse, conectarlo al back end de ABAP e implementar su primera Hola, mundo. aplicación de consola.
- 00:00:54 Esta semana, crearemos una aplicación de informe de lista de viajes de SAP Fiori de solo lectura paso a paso. Comenzaremos creando las tablas de base de datos, el modelo de datos CDS y la proyección del modelo de datos y enriquecerlos con semántica de IU.
- 00:01:14 A continuación, crearemos el servicio OData y previsualizaremos la aplicación de viajes de SAP Fiori. Finalmente, implementaremos la autorización básica para el acceso a datos mediante roles CDS.
- 00:01:28 En la unidad actual, veremos el escenario empresarial subyacente a la aplicación de viajes. El escenario de referencia de vuelo ABAP es un nuevo modelo de datos, fácil de usar y comprender, diseñado para fines de demostración y utilizado en cursos de formación,
- 00:01:48 principalmente en el contexto del modelo de programación de aplicaciones RESTful ABAP. Simula una aplicación utilizada por una agencia de viajes para reservar vuelos.
- 00:02:00 Es una versión recargada y adaptada del conocido modelo de datos de vuelo de SAP NetWeaver. El escenario de referencia de vuelo ABAP ofrece varios ejemplos que pueden servir como implementaciones de referencia de las diferentes funciones proporcionadas por RAP.
- 00:02:20 Por ejemplo, implementaciones de campo verde y campo de navegación con y sin tratamiento de borrador. No es una parte integral de la plataforma ABAP, pero está disponible para descargar en GitHub.
- 00:02:34 Por lo tanto, se debe importar a los sistemas ABAP relevantes antes de su uso. El escenario de referencia de vuelo ABAP está preinstalado en SAP Cloud Platform, sistemas de prueba de entorno ABAP y listo para su uso.
- 00:02:50 Las diferentes entidades empresariales con las que trabajaremos en nuestros escenarios actuales son viajes, reservas, agencias, vuelos y clientes. Una entidad Viajes define los datos generales del viaje, como el ID de la agencia o el ID del cliente, el estado de la reserva del viaje y el precio del viaje.
- 00:03:12 Una entidad de reserva define los datos de vuelo y reserva, el ID de cliente al que está reservado el vuelo, así como el ID de viaje al que pertenece la reserva. Una entidad de agencia define los datos de agencia de viajes, como la dirección y los datos de contacto.
- 00:03:31 Una entidad Vuelo define los datos generales del vuelo para cada conexión. Por último, una entidad del Cliente proporciona una descripción detallada de un cliente o pasajero de vuelo, como el nombre, la dirección y los datos de contacto.

- 00:03:48 La principal entidad empresarial en nuestro escenario será el viaje y la reserva. Crearemos tablas de base de datos para almacenar los datos de viaje y reserva, al reutilizar las bases de datos proporcionadas por el escenario de referencia de vuelo ABAP para las otras
- 00:04:07 entidades empresariales. Al final de esta semana, tendremos una aplicación de informes de lista de viajes de elementos de SAP Fiori con funciones de búsqueda de texto completo y de filtro para visualizar datos
- 00:04:19 existentes de viajes y reservas. La aplicación consta del plano de planta del informe de lista con navegación al plano de planta de la página de detalles del objeto.
- 00:04:32 En la primera semana, unidad dos, se muestra un resumen del flujo de trabajo de desarrollo típico en RAP. El flujo de desarrollo para crear nuestra aplicación de solo lectura sería el siguiente.
- 00:04:47 Primero, crearemos las tablas de base de datos para almacenar los datos de viajes y reservas. A continuación, definimos el modelo de datos CDS para las entidades Viajes y Reserva.
- 00:05:02 En el siguiente paso, proyectaremos la parte del modelo de datos necesaria para nuestra aplicación de viaje y la enriqueceremos con información adicional a nivel de modelo de datos y de modelo de metadatos con, por ejemplo, campos adicionales y semántica de IU.
- 00:05:21 A continuación, definiremos el alcance de nuestro servicio especificando las entidades relevantes que se expondrán en nuestro servicio, vincularlo al protocolo OData y al caso de uso de la IU de SAP Fiori y previsualizarlo en las herramientas de desarrollo ABAP.
- 00:05:41 Por último, definiremos un rol CDS para restringir el acceso a los datos de la entidad Travel. Ahora estamos al final de esta unidad.
- 00:05:55 Hagamos una breve recapitulación. En esta unidad, ha aprendido sobre el escenario de referencia de vuelo ABAP y el escenario empresarial que implementaremos esta semana, qué entidades empresariales se utilizarán y el
- 00:06:09 flujo de desarrollo que cubriremos en las siguientes unidades. Para más información sobre esta unidad, véase el apéndice de la presente presentación.
- 00:06:19 Gracias por escucharle y verle en la siguiente unidad, donde crearemos las tablas de base de datos CDS.

## Week 2 Unit 2

- 00:00:05 Hola, y bienvenido al curso de openSAP Building Apps con ABAP RESTful Application Programming Model, semana dos, unidad dos, Creación de tablas de base de datos.
- 00:00:14 Me llamo Carine Tchoutouo Djomo y soy gestor de productos para la plataforma ABAP. En la unidad anterior, tuvimos un resumen del escenario empresarial que vamos a trabajar en toda
- 00:00:25 esta semana. Ahora, comenzaremos con el desarrollo de nuestra aplicación de informes de lista de viajes de SAP
- 00:00:32 Fiori de solo lectura creando las tablas de base de datos relevantes. En la pantalla, el área en la que vamos a trabajar en esta unidad se resalta en el flujo de desarrollo,
- 00:00:45 las tablas de base de datos para almacenar los datos de la aplicación. Al final de esta unidad, tendremos dos tablas de base de datos para almacenar los datos de viaje y
- 00:00:56 reserva. También los rellenaremos con datos de demostración y tendremos una vista previa de los datos en ADT.
- 00:01:02 El editor de tablas ADT basado en texto que se visualiza en la pantalla difiere considerablemente del editor de tablas basado en formularios en el Workbench ABAP, pero el mismo alcance de función
- 00:01:13 está disponible aquí. Los atributos específicos de tabla y de campo se especifican mediante anotaciones.
- 00:01:20 Los campos de tabla se especifican entre corchetes y la palabra clave "clave" se utiliza para especificar campos clave.
- 00:01:28 Los atributos técnicos de la tabla se definen en la parte superior de la definición de tabla, antes de la palabra clave "definir".
- 00:01:35 Por ejemplo, la ampliación y la categoría de tabla. En la captura de pantalla, puede ver la anotación semántica utilizada para especificar la clave de
- 00:01:46 moneda de referencia para los campos de importe, tasa de reserva, precio total y precio de vuelo. Ahora entraremos en el sistema y crearemos estos nuevos artefactos.
- 00:02:00 En la semana uno, configuramos nuestro entorno de desarrollo ABAP y creamos un paquete para agrupar todos los ejercicios de todo el curso.
- 00:02:08 También aprendimos algunas formas de acelerar el desarrollo de ABAP en ADT. Durante las demostraciones del sistema, utilizaré fragmentos de código preparados para acelerar el
- 00:02:19 desarrollo, pero, por supuesto, explicaré el código insertado. Esto nos ayudará a centrarnos en los principales aspectos de la aplicación.
- 00:02:28 Utilizaré el sufijo 1234 para los diferentes objetos que voy a crear. Vamos a crear un subpaquete para contener la aplicación de viajes.
- 00:02:40 Para ello, haga clic con el botón derecho en el paquete y seleccione Nuevo y, a continuación, Paquete ABAP en el menú contextual.
- 00:02:46 Introduzca un nombre y una descripción en el asistente de creación y pulse Siguiente. Asigne una orden de transporte y pulse Finalizar.
- 00:03:04 Añadamos el nuevo paquete a nuestros paquetes favoritos para acceder fácilmente a él. Antes de empezar con la creación de las tablas de base de datos, veamos rápidamente el escenario de
- 00:03:20 referencia de vuelo ABAP, que pertenece al área de nombres /DMO/. Para ello, añadiré su paquete principal, /DMO/FLIGHT a mis paquetes favoritos.
- 00:03:34 El escenario de referencia contiene implementaciones de ejemplo de las diferentes capacidades de RAP, así como varias implementaciones existentes, objetos de diccionario, como tablas de base de datos, dominios
- 00:03:52 y elementos de datos y datos de demostración. Echemos un vistazo a los datos de viaje y reserva proporcionados por el escenario de referencia y

00:04:07 hagamos una vista previa de los datos para ello. El alcance de la función del escenario de referencia se mejora regularmente para cubrir nuevas

00:04:18 capacidades y escenarios de RAP. Las guías de desarrollo para las diferentes capacidades de RAP y ejemplos de escenarios de

00:04:29 referencia se proporcionan en la documentación de RAP en el SAP Help Portal. Ahora vamos a crear la tabla de la base de datos de viajes.

00:04:46 Para ello, haga clic con el botón derecho en el nuevo paquete y seleccione Nuevo y, a continuación, Otro objeto de ABAP Repository en el menú contextual.

00:04:54 Busque Tabla de base de datos en el asistente de creación, selecciónela de la lista y pulse Siguiente.

00:05:02 Introduzca un nombre y una descripción y pulse Siguiente para continuar. Asigne una orden de transporte y pulse Finalizar.

00:05:24 La tabla creada se abre en el editor. El campo de mandante se ha añadido automáticamente porque es una tabla específica de un mandante.

00:05:32 Podemos conservar los atributos técnicos específicos de tabla predeterminados especificados en la parte superior antes de la sentencia de definición de tabla y añadir los campos de tabla entre corchetes.

00:05:44 He insertado mi primer fragmento de código para eso. La tabla ahora consta de los campos clave, client y Travel\_uuid y los campos de tabla, como un ID de

00:05:56 viaje legible para humanos, agency\_id, el customer\_id, el total\_price y el status global, así como los datos de administración estándar, como el usuario correspondiente o el momento de la creación

00:06:12 de datos. El campo de tabla currency\_code se especifica como el campo de referencia para los campos de importe

00:06:19 resering\_honorarios y total\_price mediante una anotación semántica. Ahora podemos guardar y activar la tabla de viajes.

00:06:27 Vamos a crear ahora la tabla de base de datos de reservas. Haga clic con el botón derecho en la carpeta Tablas de base de datos y seleccione Nueva tabla de base de

00:06:37 datos en el menú contextual. Introduzca un nombre y una descripción en el asistente y pulse Siguiente para continuar.

00:06:47 La asignación de paquete se ha efectuado automáticamente. Asigne una orden de transporte y pulse Finalizar.

00:06:57 Se crea la nueva tabla específica de mandante. También podemos conservar los atributos técnicos específicos de tabla por defecto y ir entre

00:07:07 corchetes para añadir los campos de tabla utilizando otro fragmento de código. La tabla ahora consta de los campos clave client y Booking\_uuid, y otros campos, como un resering\_id

00:07:22 legible para humanos, el viaj\_uuid al que pertenece la reserva y el carrier\_id, un registro de reserva. Además, los datos de gestión estándar, como el usuario correspondiente de la última modificación.

00:07:41 Vamos a activar y guardar nuestra vista. Como podemos ver, ambas tablas están vacías actualmente.

00:07:52 Un informe de lista de viajes sin datos para visualizar sería bastante aburrido. Por lo tanto, vamos a crear un generador de datos simple para rellenar la tabla de base de datos con

00:08:01 algunos datos de demostración. Para ello, creemos una clase ABAP.

00:08:05 Haga clic con el botón derecho en el paquete y seleccione Nuevo y, a continuación, Clase ABAP en el menú contextual.

00:08:12 Proporcione un nombre y una descripción en el asistente de creación, añada la interfaz IF\_OO\_ADT\_RUN y pulse Siguiente.

00:08:34 Asigne una orden de transporte y pulse Finalizar. Tenemos una advertencia en el editor.

- 00:08:48 Puede utilizar la corrección rápida ADT para añadir la implementación de método y resolver el problema. Ahora utilizaré otro fragmento de código para proporcionar la implementación y luego explicarla.
- 00:09:15 Pero, en primer lugar, permítanme sustituir el marcador de posición de hashes por el sufijo que he elegido, 1234.
- 00:09:28 Ahora examinemos con más detenimiento el código insertado. En primer lugar, se borran las entradas existentes en ambas tablas.
- 00:09:41 A continuación, se seleccionan los datos de la tabla /dmo/Travel y se insertan en la nueva tabla de viajes.
- 00:09:47 La función SQL uuid( ) se utiliza para fijar el valor del campo clave Travel\_uuid. A continuación, se ejecuta la sentencia Commit Work para conservar los datos.
- 00:09:59 La selección de datos se ha limitado a un máximo de 200 registros de viaje, pero se puede modificar si lo desea.
- 00:10:05 En el siguiente bloque de código, los registros de reserva correspondientes se seleccionan de la tabla de reservas /dmo/reserve, unidos con la tabla de viajes rellenada anteriormente, y los datos se
- 00:10:17 insertan en la nueva tabla de reservas. La función SQL uuid( ) también se utiliza para establecer el valor del campo clave Booking\_uuid, y
- 00:10:26 la sentencia commit work se ejecuta para conservar los datos. Y, por último, se escribe un mensaje de éxito en la consola.
- 00:10:35 Hemos grabado y activado la clase ABAP y ahora podemos ejecutarla como una aplicación de consola. Ahora podemos obtener una vista previa de los datos de ambas tablas.
- 00:11:28 Ya hemos terminado. Eso fue con la demostración del sistema y ahora estamos al final de esta unidad.
- 00:11:34 Como breve resumen, en esta unidad, ha aprendido a utilizar las herramientas de desarrollo ABAP para crear tablas de base de datos para su aplicación de demostración y una forma de rellenarlas con algunos
- 00:11:46 datos ficticios. Para más información sobre esta unidad, véase el apéndice de esta presentación.
- 00:11:53 Gracias por escuchar y verle en la siguiente unidad, donde vamos a crear el modelo de datos CDS.

## Week 2 Unit 3

- 00:00:05 Hola, y bienvenido al curso de openSAP "Building Apps with the ABAP RESTful Application Programming Model", semana dos, unidad tres, Creación del modelo de datos de Core Data Services.
- 00:00:16 Me llamo Carine Tchoutouo Djomo y soy gestor de productos para la plataforma ABAP. En la unidad anterior, creamos las tablas de base de datos para almacenar los datos de viaje y reserva y los rellenamos con algunos datos ficticios.
- 00:00:29 En esta unidad, ahora vamos a crear nuestro modelo de datos CDS sobre ellos. En la pantalla, el área en la que vamos a trabajar en esta unidad se resalta en el flujo de desarrollo.
- 00:00:44 CDS es la piedra angular del modelo de programación de aplicaciones RESTful ABAP. Lo usaremos para definir el modelo de datos del business object de viaje y el objeto empresarial de reserva.
- 00:01:00 Como ya se introdujo en la semana uno, CDS es la infraestructura de modelado de datos de próxima generación en la plataforma ABAP. Proporciona un enfoque declarativo para definir modelos de datos específicos de dominio y ricos semánticamente cerca del pensamiento conceptual, lo que facilita la definición,
- 00:01:19 comprensión y consulta de los modelos de datos. CDS ofrece un conjunto enriquecido de funciones y expresiones integradas y tiene su propio lenguaje de control de datos para restringir el acceso a los datos.
- 00:01:32 Las relaciones entre entidades empresariales a nivel de modelo de datos se pueden definir mediante asociaciones CDS, y la semántica específica de dominio se puede añadir a los modelos de datos mediante anotaciones CDS.
- 00:01:46 ABAP CDS admite varias entidades de modelado CDS, por ejemplo, vistas CDS para la creación de vistas estándar, ampliaciones de vista CDS para la ampliación sin modificaciones del modelo de datos CDS y funciones de tabla CDS para acceder a las capacidades avanzadas de SAP HANA como Data Mining o el análisis predictivo.
- 00:02:13 Al final de esta unidad, tendremos nuestro árbol de composición de business object definido mediante dos entidades de vista CDS para definir el BO raíz de viaje y el BO subordinado de reserva y la relación entre ellas.
- 00:02:28 Las entidades de vista CDS son el sucesor de las vistas basadas en Dictionary ABAP CDS. Ofrecen muchas mejoras en el área de gestión del ciclo de vida El editor CDS basado en fuente en ADT ofrece funciones como verificaciones de sintaxis, resaltado de sintaxis, información
- 00:02:47 de elemento y correcciones rápidas para dar un mejor soporte al desarrollo de modelos de datos CDS. También se proporciona el framework doble de test CDS para el test de unidad y las herramientas de soporte.
- 00:03:05 Ahora entraremos en el sistema y continuaremos con el desarrollo de nuestra aplicación de viajes de SAP Fiori. Primero definimos el modelo de datos CDS clásico y, a continuación, lo ampliaremos para definir la estructura de business object.
- 00:03:21 Durante los ejercicios, sustituiré todas las ocurrencias con marcadores de posición con el sufijo que he elegido, 1234. Comencemos con la creación de la vista de viaje.
- 00:03:31 Haga clic con el botón derecho en la tabla de viajes y seleccione Nueva definición de datos en el menú contextual. Introduzca un nombre y una descripción y pulse Siguiente para continuar.
- 00:03:42 El proyecto, el paquete y el objeto referenciado se han asignado automáticamente en el asistente de creación. Asigne la orden de transporte y pulse Siguiente.
- 00:03:52 Se proporcionan varios modelos de definición de datos para su comodidad. Seleccione el modelo de entidad Definir vista de la lista y, a continuación, pulse Finalizar.

- 00:04:03 La nueva definición de datos aparece en el editor. La tabla de viajes se utiliza como fuente de datos y los campos de tabla se han insertado automáticamente en la lista de proyección entre corchetes excepto el campo de mandante que
- 00:04:17 trata implícitamente la plataforma de aplicación porque la vista es una vista específica de mandante. Definamos alias para la fuente de datos y las columnas de vista utilizando la palabra clave "como" y ampliamos el modelo de datos CDS con asociaciones para definir la relación con otras
- 00:04:33 entidades. He preparado un fragmento de código para eso.
- 00:04:37 Insertémoslo. Las asociaciones a las entidades Reserva, Agencia, Cliente y Moneda se definen y exponen en la lista de proyección.
- 00:04:50 Una asociación se define con una cardinalidad, una entidad CDS de destino, un alias opcional y una condición ON. El motivo del error actual es que la vista CDS de reserva especificada aún no existe.
- 00:05:05 En la lista de proyección, el elemento de vista overall\_status se ha renombrado como TravelStatus. Ahora podemos continuar y añadir anotaciones semánticas a los campos de moneda y administrativos para garantizar un procesamiento de datos uniforme en el lado del consumidor.
- 00:05:27 Podemos especificar currencyCode como el campo de referencia de los campos de moneda BookingRate y TotalPrice mediante una anotación semántica. La anotación de los campos administrativos CreatedBy, CreatedOn, LastChangedBy, LastChangedOn y LocalLastChangedOn es un paso de preparación para la semana tres, donde se activará el
- 00:06:06 comportamiento transaccional de la aplicación Informe de lista de viajes. Estas anotaciones son necesarias para permitir la actualización automática de los campos de administración en cada operación.
- 00:06:57 Grabe las modificaciones, pero aún no active la vista. Podemos continuar y crear la vista de reserva.
- 00:07:06 Haga clic con el botón derecho en la tabla de reservas y seleccione Nueva definición de datos en el menú contextual. Introduzca un nombre y una descripción y pulse Siguiente para continuar.
- 00:07:16 El proyecto, el paquete y el objeto referenciado se han asignado automáticamente. Asigne una orden de transporte y pulse Siguiente.
- 00:07:30 Seleccione la plantilla Definir entidad de vista y pulse Finalizar. La nueva definición de datos aparece en el editor.
- 00:07:39 La tabla de base de datos de reserva ya se utiliza como fuente de datos. Proporcionemos un alias para ello.
- 00:07:45 Permítanme insertar otro fragmento de código para actualizar la definición de datos de reserva. Las asociaciones a las entidades Viajes, Cliente, Transportista, Conexión, Vuelo y Moneda se definen y exponen en la lista de proyección.
- 00:08:09 En la lista de proyección, las columnas de vista ahora tienen alias y las anotaciones @semántica también se especifican para el campo de moneda FlightPrice y los campos administrativos CreatedBy, LastChangedBy y LocalLastChangedOn.
- 00:08:30 Grabe las modificaciones. Ahora podemos continuar y activar ambas vistas CDS juntas.
- 00:08:43 Las vistas ahora están activadas, pero las advertencias se visualizan en la vista Problemas. El motivo de estas advertencias es que la verificación de autorización se ha activado para ambas vistas CDS mediante la anotación de vista de control de acceso, pero aún no se ha
- 00:09:01 definido ningún rol CDS para las entidades CDS. Trabajaremos en esta cuestión en la unidad siete de esta semana, donde implementaremos el control de acceso.
- 00:09:10 Ahora previusualizaremos los datos en ADT. La vista previa de datos en ADT ofrece varias funciones, incluida la posibilidad de navegar a datos asociados al previusualizar vistas CDS.

00:09:46 Por último, definamos la estructura de nuestro business object. Este paso no es obligatorio al crear aplicaciones de solo lectura, sino más un paso de preparación para la activación transaccional de nuestra aplicación de viaje en la semana

00:10:00 tres. Para ello, vaya a la vista de viaje y especifique la entidad de viaje como nodo raíz de la composición de BO utilizando la palabra clave "raíz" en la declaración de definición, y

00:10:12 también especifique la vista de reserva como BO inferior utilizando la asociación especial, composición. Grabe las modificaciones, pero aún no active la vista.

00:10:25 Ahora cambie a la vista de reserva y especifique la entidad Viajes como nodo superior mediante la asociación especial, asociación con superior. Ahora podemos guardar y activar ambas vistas CDS juntas.

00:10:53 Estas modificaciones no afectan a los datos. Eso es todo.

00:11:12 Eso fue con la demostración del sistema. Ahora estamos al final de esta unidad.

00:11:16 Permítanme hacer una breve recapitulación. En esta unidad, ha aprendido a definir un modelo de datos y el árbol de composición de business object con CDS y cómo previsualizar los datos en ADT.

00:11:33 Para más información sobre esta unidad, véase el apéndice de la presente presentación. Gracias por escuchar y verle en la siguiente unidad en la que vamos a proyectar la parte relevante de nuestro modelo de datos CDS para nuestra aplicación de informes de lista de

00:11:49 viajes y comenzar a enriquecerla.



## Week 2 Unit 4

- 00:00:05 Hola y bienvenido al curso de openSAP "Building Apps with the ABAP RESTful Application Programming Model", semana dos, unidad cuatro - Definición de la "Proyección de modelo de datos CDS".
- 00:00:16 Me llamo Carine Tchoutouo Djomo y soy gestor de productos para la plataforma ABAP. En la unidad anterior creamos el modelo de datos subyacente a nuestra aplicación de informe de lista de
- 00:00:27 viajes y definimos el modelo de composición que incluye los viajes y los business objects de reserva con CDS.
- 00:00:34 En esta unidad, proyectaremos la parte relevante de nuestro modelo de composición CDS necesaria para nuestra aplicación de informes de lista de viajes y comenzaremos a enriquecerla.
- 00:00:50 El área en la que vamos a trabajar en esta unidad se resalta en el flujo de desarrollo RAP: la proyección del modelo de datos basado en CDS que permite la desnormalización y el enriquecimiento del
- 00:01:03 modelo de datos subyacente. La enriqueceremos con datos adicionales y algunos metadatos relacionados con la estructura de objetos y
- 00:01:15 consumo. Al final de esta unidad, tendremos el modelo de datos CDS creado a partir de dos proyecciones de vista CDS:
- 00:01:24 una vista de previsión de viaje y una vista de previsión de reserva. Como puede ver en la captura de pantalla, habilitaremos la proyección diferente para la búsqueda
- 00:01:33 de texto completo y especificaremos el elemento de vista que se debe tener en cuenta en una búsqueda de estilo libre mediante las anotaciones @Search.
- 00:01:41 Permitiremos la ampliación sin modificaciones de ambas vistas de proyección con anotaciones de IU en un documento separado, la extensión de metadatos CDS (en resumen MDEs), para separarlas de las anotaciones
- 00:01:55 relacionadas con el business object. Utilizaremos CDS MDEs en la siguiente unidad.
- 00:02:03 Añadiremos algunos campos nuevos como Agency\_Name, customer\_Name y Carrier\_Name en la lista de proyección y estableceremos la conjunción de un campo con sus unidades de texto descriptivas independientes del
- 00:02:18 idioma utilizando las anotaciones @ObjectModel. También definiremos una ayuda para entradas para campos como ID de agencia, ID de cliente e ID de
- 00:02:29 transportista utilizando anotaciones de @Consumo y redireccionaremos los nodos de composición Viaje y Reserva a las entidades correspondientes de la capa de proyección de business object.
- 00:02:42 Ahora entraremos en el sistema y sigamos con la definición de la proyección del modelo de datos CDS. La estructura de business object definida previamente es independiente del servicio y está prevista para el
- 00:02:55 uso general. Con las vistas de proyección, solo podemos exponer aquellos elementos que son relevantes para el servicio
- 00:03:02 específico, incluida la desnormalización del modelo de datos subyacente. También se puede definir el ajuste preciso, como las ayudas para entradas, la búsqueda y la semántica de
- 00:03:12 IU. Estamos siguiendo la convención para fijar nombres del modelo de datos virtual de SAP S/4HANA en el que los
- 00:03:18 nombres de las vistas de consumo comienzan con el área de nombres, seguida de la letra mayúscula C. Comenzaremos con la definición de la vista de proyección del business object de viaje.
- 00:03:29 Haga clic con el botón derecho en la vista BO de viaje creada anteriormente y seleccione Nueva definición de datos en el menú contextual.

00:03:36 Introduzca un nombre y una descripción y pulse Siguiente para continuar. El proyecto, el paquete y el objeto referenciado se asignan automáticamente en el asistente de creación.

00:03:49 Asigne una orden de transporte y pulse Siguiente. Seleccione la plantilla Definir vista de proyección de la lista y pulse Finalizar.

00:03:59 La nueva definición de datos aparece en el editor. Especifique la vista de proyección BO como entidad raíz y proporcione un alias para la vista de viaje

00:04:12 proyectada que ya está asignada. Todos los elementos de vista y asociaciones se han insertado automáticamente en la lista de proyección.

00:04:24 Aquí podemos añadir los elementos AgencyName de la asociación \_Agency y CustomerName de la asociación \_Customer a la lista de proyección.

00:04:51 Ahora añadamos algunas semánticas específicas de servicio al modelo de datos proyectado. En primer lugar, permita que la vista de proyección se amplíe con extensiones de metadatos separadas y

00:05:09 habilite la búsqueda de texto completo utilizando las anotaciones de vista @Metadata.AllowExtensions y @Search.searchable.

00:05:23 Regrese a la lista de proyección. Habilite la búsqueda de estilo libre para las columnas de vista TravelID, AgencyID y CustomerID utilizando la

00:05:47 anotación de elemento @Search.DefaultSearchElement. Especifique los elementos previamente añadidos AgencyName y CustomerName como descripciones textuales

00:06:21 para los elementos AgencyID y CustomerID respectivamente utilizando la anotación de elemento @ObjectModel.text.element.

00:06:49 Defina las ayudas para entradas para los elementos de vista AgencyID, CustomerID y CurrencyCode utilizando la anotación de elemento @Consumption.ValueHelpDefinition.

00:07:14 Aquí, se debe especificar el nombre de la entidad CDS de destino que actúa como proveedor de ayuda para entradas y el nombre de su elemento vinculado al elemento local.

00:07:30 Puede utilizar la compleción de código al especificar la entidad de proveedor de ayuda para entradas de destino.

00:08:16 También vamos a especificar el elemento CurrencyCode como campo de referencia para los campos de moneda BookingRate y TotalPrice utilizando la anotación de elemento @Semantics.amount t.Currency.

00:09:08 Mantendremos todas las asociaciones expuestas para nuestro caso de uso pero redirigiremos la entidad BO subordinada de reserva a la vista de previsión BO de reserva adecuada.

00:09:21 La vista de proyección indicada aún no existe. Grabe las modificaciones, pero no active la vista CDS.

00:09:32 Ahora crearemos la vista de previsión BO de reserva. Haga clic con el botón derecho en la vista BO de reserva creada anteriormente y seleccione Nueva

00:09:42 definición de datos Introduzca un nombre y una descripción y pulse Siguiente para continuar. El proyecto, el paquete y el objeto referenciado se asignan automáticamente en el asistente de creación.

00:09:54 Asigne una orden de transporte y pulse Siguiente. Seleccione la plantilla Definir vista de proyección de la lista y pulse Finalizar.

00:10:02 La nueva definición de datos aparece en el editor. Proporcione un alias para la vista BO de reserva que se especifica como la vista proyectada.

00:10:17 Además, aquí, permita que la vista de proyección se amplíe con extensiones de metadatos separadas y habilite la vista para la búsqueda de texto completo utilizando las anotaciones de vista @Metadata y

00:10:33 @Search adecuadas. Ahora mejoraremos la lista de proyección.

00:10:38 He preparado un fragmento de código para eso. ¿Qué ha cambiado?

00:10:52 Las columnas de vista BookingID y CustomerID están activadas para la búsqueda de estilo libre. La lista de proyección se ha ampliado con los elementos CustomerName y CarrierName de las

00:11:06 asociaciones \_Customer y \_Carrier respectivamente. Estos elementos proporcionan la descripción textual para los elementos CustomerID y CarrierID y se

00:11:18 especifican como tales mediante la anotación de elemento @ObjectModel.text.element. Las ayudas para entradas se definen para los elementos CustomerID, CarrierID, ConnectionID y CurrencyCode.

00:11:33 En la ayuda para entradas del elemento ConnectionID, se define una condición de vinculación adicional para devolver valores del registro de ayuda para entradas seleccionado para los elementos de vista local

00:11:47 CarrierID, FlightDate, FlightPrice y Currency. El campo de vista CurrencyCode se especifica como campo de referencia para el campo de moneda

00:11:55 FlightPrice. Las asociaciones relevantes se exponen y la asociación BO superior se ha redirigido a la previsión BO de

00:12:04 viaje. Grabe las modificaciones.

00:12:08 Ahora podemos activar ambas vistas de proyección de BO juntas al mismo tiempo. Ahora previsualicemos los datos.

00:12:29 Podemos ver que las vistas de proyección de BO se utilizan cuando se sigue una asociación en la vista previa de datos.

00:12:55 Eso es todo. Eso fue para la demostración.

00:13:00 Ahora estamos al final de esta unidad. Permítanme hacer una breve recapitulación:

00:13:06 En esta unidad, ha aprendido sobre los motivos para proyectar un modelo de datos CDS existente y cómo definir una proyección de modelo de datos CDS para business objects.

00:13:20 Para más información sobre esta unidad, véase el apéndice de la presente presentación. Gracias por escucharle y verle en la siguiente unidad en la que vamos a enriquecer aún más la proyección del

00:13:32 modelo de datos CDS con la semántica de IU.

## Week 2 Unit 5

- 00:00:05 Hola y bienvenido al curso de openSAP "Building Apps with the ABAP RESTful Application Programming Model", semana dos, unidad cinco: "Enriquecimiento del modelo de datos proyectado con metadatos de IU".
- 00:00:17 Me llamo Carine Tchoutouo Djomo y soy gestor de productos para la plataforma ABAP. En la unidad anterior definimos la proyección de modelo de datos CDS necesaria para nuestra aplicación de informe de lista de viajes de SAP Fiori.
- 00:00:31 En esta unidad, continuaremos y enriqueceremos aún más con la semántica de IU para la generación de nuestra aplicación de informes de lista de viajes de elementos de SAP Fiori en la siguiente unidad.
- 00:00:46 El área en la que vamos a trabajar en esta unidad se resalta en el flujo de desarrollo RAP. Esta vez utilizaremos ampliaciones de metadatos CDS para enriquecer aún más la proyección del modelo de datos CDS con anotaciones de IU.
- 00:01:04 Como ya se mencionó en la semana uno, la unidad tres, ampliaciones de metadatos CDS (en forma breve MDEs) permiten la ampliación sin modificaciones de modelos de datos CDS o proyecciones de modelo de datos CDS a nivel de metadatos.
- 00:01:20 Con un CDS MDE, puede añadir semántica para diferentes dominios a una entidad CDS existente utilizando anotaciones, normalmente, anotaciones de IU. Al utilizar CDS MDEs, logramos una separación del modelo de datos con todas las semánticas de business object de la semántica de IU.
- 00:01:41 Esta separación de las preocupaciones permite una gestión de modificaciones simplificada y una ampliación sin modificaciones. Gracias al enfoque de capas de CDS MDEs, puede definir más de una ampliación de metadatos CDS para una entidad CDS.
- 00:02:03 Al final de esta unidad, tendremos dos extensiones de metadatos CDS que enriquecerán las vistas de proyección CDS definidas en la unidad anterior: un MDE CDS para la proyección de vista de viaje y otro para la vista de proyección de reserva.
- 00:02:18 Se puede ver información diversa en las capturas de pantalla. En la parte superior, la especificación de la capa de metadatos.
- 00:02:25 Cada extensión de metadatos se asigna a una capa, como CORE#, PARTNER# o CUSTOMER#. La capa determina la prioridad de la evaluación.
- 00:02:37 A continuación, podemos ver el nombre de la entidad CDS que se está enriqueciendo y la vista diversa y las anotaciones de IU específicas de campo. Ahora entraremos en el sistema y crearemos las extensiones de metadatos CDS para definir el diseño de nuestra aplicación de viajes de elementos de SAP Fiori.
- 00:02:55 En la unidad anterior, hemos especificado la anotación de vista @Metadata.AllowExtensions:true en las vistas de proyección BO para permitir su ampliación sin modificaciones mediante ampliaciones de metadatos CDS separadas.
- 00:03:09 El objetivo de este curso es la implementación de aplicaciones de back end RESTful. Por lo tanto, solo utilizaremos algunas anotaciones de IU para enriquecer el modelo de datos CDS proyectados para la última generación de nuestra aplicación de viaje de elementos de SAP Fiori. Comencemos con la vista de proyección de BO de viaje.
- 00:03:25 Haga clic con el botón derecho en el Explorador de proyectos y seleccione Nueva extensión de metadatos en el menú contextual. Introduzca un nombre y una descripción y pulse Siguiente para continuar.
- 00:03:29 El proyecto, el paquete y el objeto referenciado se asignan automáticamente en el asistente de creación. Asigne una orden de transporte y pulse Siguiente.
- 00:03:40 Se proporcionan plantillas de extensión de metadatos para su comodidad. Seleccione la plantilla Anotar vista y pulse Finalizar.

- 00:04:02 La nueva extensión de metadatos aparece en el editor. Tenemos la anotación @Metadata.layer en la parte superior, el nombre de la vista de proyección BO después de la sentencia ANNOTATE VIEW y una entrada ficticia entre corchetes.
- 00:04:16 Especifique #CORE como capa de metadatos, ya que somos el proveedor de aplicaciones en el escenario actual. Cuando se definen varias extensiones de metadatos para una entidad CDS determinada, la capa determina la prioridad de los metadatos.
- 00:04:32 #CORE tiene la prioridad más baja y #CLIENTE. Ahora definamos la lista separada por punto y coma de los elementos que queremos anotar.
- 00:04:43 Algunas de las anotaciones de elemento específicas de consumo especificadas en las vistas de proyección BO hacen redundantes las anotaciones de IU. Este es el caso de los elementos AgencyName y CustomerName, que son las descripciones textuales para AgencyID y CustomerID respectivamente, y para el elemento CurrencyCode, que es el campo de
- 00:05:04 referencia para BookingRate y TotalPrice. Estos tres elementos se visualizarán automáticamente junto con sus elementos relacionados en la IU.
- 00:05:14 Por lo tanto, eliminémoslos de la lista de elementos y también de las asociaciones. Volvamos a la parte superior del editor y definamos información de cabecera, como el nombre y el título del tipo, y la variante de presentación para nuestra aplicación de informe de lista
- 00:05:31 utilizando anotaciones de vista @UI. Los datos de viaje consultados se ordenarán en orden descendente según el elemento TravelID de la lista.
- 00:05:41 A continuación, en los corchetes, utilizamos las anotaciones @UI.faceta para definir la navegación a la página de objeto y su layout. La página de objeto de viaje tiene dos facetas:
- 00:05:53 la referencia de identificación de la entidad Viaje y la referencia de partida individual de la entidad Reserva, con la composición \_reserve especificada como elemento de destino. Ahora podemos continuar y especificar una posición predeterminada y, posiblemente, una etiqueta para cada elemento utilizando las anotaciones de elemento @UI correspondientes.
- 00:06:15 Siempre que estén disponibles y no se sobrescriban, las etiquetas de los elementos de datos subyacentes se utilizan como etiquetas en la IU. La anotación @UI.item se utiliza para especificar la información de diseño de cada elemento que se muestra como una columna en la lista.
- 00:06:33 La anotación @UI.Identification se utiliza para especificar la información de layout de cada elemento que se muestra en la sección de identificación de la página de objeto. Y la anotación @UI.selectionField se utiliza para activar un elemento para la selección en la barra de filtros.
- 00:06:51 Si se define una ayuda para entradas para el elemento indicado, se expone automáticamente a la IU. Para la secuencia de emplazamiento se puede utilizar cualquier número decimal como valor para la posición de la propiedad.
- 00:07:09 Con la anotación @UI.oculto:true, evitamos que los elementos se visualicen en la IU o en el diálogo de opciones de personalización. Como resultado de esta ampliación de metadatos, el elemento TravelUUID solo se visualizará en la sección de identificación de la página de objeto.
- 00:07:28 Los elementos TravelID, AgencyID y CustomerID estarán disponibles para su selección en la barra de filtros y también se mostrarán en la lista de viajes y en la sección de identificación de la página de objeto.
- 00:07:41 Excepto los campos de administración que no son relevantes en la IU, los otros elementos aparecerán en la lista y en la sección de identificación de la página de objeto. Grabe y active la ampliación de metadatos.
- 00:08:12 Ahora, vamos a enriquecer la vista de proyección de BO de reserva. Haga clic con el botón derecho en el Explorador de proyectos y seleccione Nueva extensión de metadatos en el menú contextual.

- 00:08:23 Introduzca un nombre y una descripción en el asistente de creación y pulse Siguiente para continuar. El proyecto, el paquete y el objeto referenciado se han asignado automáticamente.
- 00:08:45 Asigne una orden de transporte y pulse Siguiente. Seleccione Anotar plantilla de vista y haga clic en Finalizar.
- 00:08:52 La nueva extensión de metadatos aparece en el editor y con la vista de proyección de reserva especificada como la sentencia Vista anotada. Especifique #CORE como capa de metadatos y defina la lista de elementos que se enriquecerán con semántica de IU.
- 00:09:08 Elimine los elementos CustomerName, CarrierName y CurrencyCode de la lista porque se visualizarán automáticamente junto con sus elementos relacionados en la IU gracias a algunas anotaciones específicas de consumo especificadas en la vista de proyección de reserva.
- 00:09:24 Elimine también las asociaciones. Ahora añadimos información de cabecera y definimos el diseño de la página del objeto de reserva y el diseño de la lista de reservas en la página del objeto de viaje.
- 00:09:35 He preparado un fragmento de código para eso. Como resultado de esta ampliación de metadatos, la página de objeto de reserva solo incluye una faceta de la referencia de identificación de tipo.
- 00:09:58 El UUID de reserva solo se mostrará en la página del objeto de reserva. El UUID de viaje y los campos de administrador se fijan en ocultos y, por lo tanto, no aparecerán en la IU ni en el diálogo de opciones de personalización.
- 00:10:15 Los otros campos se mostrarán en la lista de reservas, en la página del objeto de viaje y en la sección de identificación de la página del objeto de reserva. Grabe y active la ampliación de metadatos.
- 00:10:29 Hemos realizado el enriquecimiento de nuestra proyección de modelo de datos CDS. Vamos a las perspectivas de proyección y examinémoslas rápidamente.
- 00:10:38 Pulse F5 para actualizar el editor si es necesario. Ahora tenemos un marcador en la parte izquierda de la sentencia DEFINE VIEW que muestra que la vista se ha ampliado.
- 00:10:49 En la información del elemento, podemos ver la estructura de vista completa, incluidos los nodos de composición, los objetos generados y las extensiones. También podemos echar un vistazo a las anotaciones activas de una determinada vista CDS.
- 00:11:10 Para ello, haga clic con el botón derecho en la entidad relevante en el Explorador de proyectos y vaya al menú contextual Abrir con y, a continuación, en Anotaciones activas. En la vista Anotaciones activas, podemos ver las diferentes anotaciones activas, sus valores y orígenes.
- 00:11:28 El origen puede ser la propia entidad CDS o las entidades CDS subyacentes, las tablas de base de datos, los elementos de datos o las extensiones existentes. Eso es todo.
- 00:11:43 Eso fue con la demostración del sistema. Ahora estamos al final de esta unidad.
- 00:11:48 Vamos a resumirlo con una breve recapitulación: En esta unidad, ha aprendido a enriquecer un modelo de datos CDS con anotaciones específicas de IU y cómo crear ampliaciones de metadatos CDS.
- 00:12:03 Para más información sobre esta unidad, véase el apéndice de la presente presentación. Gracias por escucharle y verle en la siguiente unidad, donde crearemos el servicio OData y previsualizaremos nuestra aplicación de viajes de elementos de SAP Fiori.

## Week 2 Unit 6

- 00:00:05 Hola, y bienvenido al curso de openSAP Creación de aplicaciones con el modelo de programación de aplicaciones RESTful ABAP. Semana dos, unidad seis, Creación y vista previa del servicio de IU OData.
- 00:00:17 Me llamo Carine Tchoutouo Djomo y soy gestor de productos para la plataforma ABAP. En la unidad anterior, hemos enriquecido aún más la proyección del modelo de datos CDS con la semántica de IU necesaria para la generación de nuestra aplicación de viajes de elementos de SAP
- 00:00:34 Fiori. En esta unidad, ahora crearemos el servicio OData y previsualizaremos la aplicación.
- 00:00:44 Las áreas en las que vamos a trabajar en esta unidad se destacan en el flujo de desarrollo RAP. Esta vez crearemos una definición de servicio para especificar el alcance de nuestro servicio OData y, a continuación, un enlace de servicio para vincular nuestra definición de servicio al
- 00:01:02 protocolo OData como servicio de IU. El alcance del servicio se define exponiendo las vistas CDS relevantes y sus metadatos y comportamiento se exponen automáticamente.
- 00:01:14 Finalmente, previsualizaremos la aplicación resultante. Al final de esta unidad, habremos creado una definición de servicio y un servicio vinculante.
- 00:01:27 Nuestro alcance de servicio incluirá la vista de proyección CDS de los objetos empresariales de viaje y reserva y las vistas CDS adicionales como agencia, cliente y vuelo, que se utilizan para las ayudas para entradas y los elementos de texto asociados.
- 00:01:45 Especificaremos alias locales para las distintas vistas CDS para permitir un consumo más sencillo del servicio OData. Previsualizaremos la aplicación de viaje en las herramientas de desarrollo ABAP mediante la vista previa de la aplicación de elementos Fiori y la utilizaremos.
- 00:02:03 Ahora pasemos al sistema y hagamos el trabajo. Primero, crearemos la definición de servicio.
- 00:02:10 En el Explorador de proyectos, haga clic con el botón derecho en la vista de proyección de BO de viaje y seleccione Nueva definición de servicio en el menú contextual. Introduzca un nombre y una descripción en el asistente de creación y pulse Siguiente para continuar.
- 00:02:23 El proyecto, el paquete y la entidad expuesta se asignan automáticamente. Asigne una orden de transporte y pulse Siguiente.
- 00:02:36 Seleccione la plantilla Definir servicio y pulse Finalizar. La nueva definición de servicio aparece en el editor.
- 00:02:43 La vista de proyección de BO de viaje ya está expuesta entre corchetes. Proporcione un alias local para facilitar el consumo del servicio.
- 00:02:53 Ahora podemos exponer la vista de proyección de BO de reserva y otras vistas útiles como proveedores de ayuda para entradas para la agencia, el cliente, el vuelo, las entidades de transportista y especificar alias.
- 00:03:22 Puede utilizar la función de ayuda de contenido para especificar las entidades CDS relevantes. Grabe y active el servicio.
- 00:04:31 Ahora, podemos seguir adelante y crear el servicio vinculante. Haga clic con el botón derecho en la definición de servicio que acaba de crear en el explorador de proyectos y seleccione Nueva vinculación de servicio.
- 00:04:43 El proyecto, el paquete y la definición de servicio se asignan automáticamente en el asistente de creación. Introduzca un nombre y una descripción, seleccione el tipo de vinculación adecuado para servicios de IU OData y pulse Siguiente para continuar.
- 00:04:59 Actualmente solo se admite OData V2; sin embargo, se planifica OData V4. Asigne una orden de transporte y pulse Finalizar.
- 00:05:14 La nueva vinculación de servicio aparece en el editor. Ahora podemos activar el punto final del servicio local.

- 00:05:26 Esto puede tardar unos segundos. El servicio ahora está activado y listo para el consumo a través de un cliente OData.
- 00:05:49 En el Explorador de proyectos, podemos ver los artefactos que se han generado durante la activación. No vamos a profundizar en ellos.
- 00:05:58 Actualice el paquete si es necesario. En el editor de vinculación de servicios, la URL de servicio relativa, los conjuntos de entidades expuestos y las asociaciones ahora se muestran en el área de información de servicio en el lado
- 00:06:13 derecho. Hagamos clic en el enlace URL de servicio para ver rápidamente el documento de metadatos de servicio en el navegador.
- 00:06:24 Volvamos al enlace de servicio en ADT y previsualicemos la aplicación. Para ello, seleccione la entidad Viajes y pulse el botón Vista previa o haga clic con el botón derecho en ella y seleccione Abrir vista previa de aplicación de elementos Fiori en el menú
- 00:06:44 contextual. La aplicación de informe de lista de viajes ahora está abierta en el navegador.
- 00:06:49 Pulse Ir para cargar los datos back end. En la parte superior, tenemos el campo de entrada para la búsqueda de texto completo en la parte izquierda y junto a ella, la barra de filtros.
- 00:07:01 La lista de viajes se muestra a continuación en orden descendente. Podemos navegar a la página de objeto de detalles de cada registro de viaje.
- 00:07:10 En la página de objeto, tenemos el identificador de viaje que se muestra en el encabezado y las dos facetas, Viajes y Reservas. En la faceta Reserva , tenemos un campo de búsqueda de texto completo y una lista de reservas.
- 00:07:22 Podemos navegar a la página de objetos de cada registro de reservas. Volvamos a la página de entrada de nuestra aplicación de informes de lista y realizamos una búsqueda de texto completo.
- 00:07:34 Introduzca un valor, por ejemplo, una pantalla, pulse Intro y verifique el resultado. Se han encontrado cuatro registros.
- 00:07:44 Las columnas ID de agencia e ID de cliente se han buscado según las anotaciones de búsqueda definidas en la vista de previsión BO de viaje. Ahora filtremos las entradas, por ejemplo, por el ID de agencia.
- 00:07:58 Aquí podemos utilizar la ayuda para entradas definida para este elemento. Pulse Intro para visualizar el resultado.
- 00:08:06 Podemos personalizar la lista añadiendo, moviendo y eliminando columnas en el diálogo de opciones de personalización. Por ejemplo, añadamos el UUID, muévelo a la primera posición, elimine algunos campos y confirme el diálogo.
- 00:08:33 Podemos ver la nueva disposición de la lista de viajes. Podemos restaurar fácilmente el diseño predeterminado.
- 00:08:44 Eso es todo. Eso fue con la demostración del sistema.
- 00:08:48 Ahora estamos al final de esta unidad. Permítanme hacer una breve recapitulación.
- 00:08:52 En esta unidad, ha aprendido sobre las definiciones de servicio y la vinculación de servicio y cómo crearlas, y cómo previsualizar la aplicación de elementos de SAP Fiori en las herramientas de desarrollo ABAP.
- 00:09:07 Para más información sobre esta unidad, véase el apéndice de la presente presentación. Gracias por escuchar y verle en la siguiente unidad en la que implementaremos una autorización básica para el acceso a datos mediante roles CDS.



## Week 2 Unit 7

- 00:00:05 Hola, y bienvenido al curso de openSAP Creación de aplicaciones con el modelo de programación de aplicaciones RESTful ABAP. Semana dos, unidad siete, Implementación de autorizaciones básicas.
- 00:00:15 Me llamo Carine Tchoutouo Djomo y soy gestor de productos para la plataforma ABAP. En la unidad anterior creamos el servicio OData y previsualizamos la aplicación de informes de lista de viajes de los elementos de SAP Fiori en las herramientas de desarrollo ABAP.
- 00:00:32 En esta unidad, implementaremos una autorización básica para el acceso a datos mediante roles CDS para el business object de viaje. Las áreas en las que vamos a trabajar en esta unidad se destacan en el flujo de desarrollo RAP.
- 00:00:50 Esta vez, crearemos un objeto de autorización y roles CDS para la entidad Travel. Luego verificaremos el efecto de la verificación de autorización en la aplicación de viajes.
- 00:01:06 Como ya se mencionó en la semana uno, ABAP CDS proporciona un lenguaje de control de datos, en pocas palabras, DCL, para definir el acceso de autorización a los datos mediante un rol CDS.
- 00:01:18 El enfoque de autorización clásico utilizado en la plataforma ABAP requiere verificaciones de autorización codificadas explícitamente en los datos seleccionados, que se almacenan en tablas internas en la capa de aplicación.
- 00:01:30 CDS DCL, en su lugar, ofrece un enfoque declarativo basado en verificaciones de autorización implícitas que ocurren durante los intentos de acceso a la entidad CDS. Los roles CDS se pueden definir utilizando condiciones PFCG, condiciones literales, condiciones de usuario y condiciones de herencia.
- 00:01:53 Deben definirse explícitamente para cada entidad CDS, cuando sea necesario. Forman parte del modelo de datos y, por lo tanto, también se transfieren a la base de datos para que solo se devuelvan los datos autorizados al leer un modelo de datos CDS.
- 00:02:11 Los roles CDS se definen una vez y se utilizan automáticamente en todas partes al acceder a las entidades CDS. Al final de esta unidad, habremos definido un objeto de autorización y dos roles CDS para la entidad de viaje, uno en la capa de modelado de datos y otro en la capa de aprovisionamiento
- 00:02:34 de servicios empresariales para la proyección del modelo de datos. El rol CDS en la capa de modelado de datos se definirá con condiciones literales y PFCG, y crearemos un objeto de autorización para este fin.
- 00:02:50 Por último, el segundo rol CDS heredará la condición de la capa subyacente. Volvamos al sistema e implementemos las autorizaciones básicas.
- 00:03:05 Ahora definimos el control de acceso para la entidad de viaje. Las reglas de acceso consistirán en una condición literal para el elemento de vista CurrencyCode y una condición PFCG para el elemento de vista TravelStatus.
- 00:03:19 Para la definición de la condición PFCG, crearemos un objeto de autorización, incluido el campo de autorización y el elemento de datos, desde cero. Comencemos con la creación del elemento de datos.
- 00:03:31 Para ello, haga clic con el botón derecho en la carpeta Diccionario en el Explorador de proyectos y seleccione Nuevo, Elemento de datos del menú contextual Introduzca un nombre y una descripción en el asistente de creación y pulse Siguiente para continuar.
- 00:03:50 Asigne una orden de transporte y pulse Finalizar. El elemento de datos nuevo aparece en el editor correspondiente.
- 00:03:58 El dominio debe seleccionarse en el campo Categoría. Introduzca el nombre del tipo.
- 00:04:10 Actualice las etiquetas de los campos, por ejemplo, Status para el breve y Status de viaje para el resto. La actualización de propiedades adicionales no es necesaria para el escenario actual.
- 00:04:31 Grabe y active el nuevo elemento de datos. Ahora, vamos a crear el campo de autorización.

- 00:04:39 Haga clic con el botón derecho en su paquete en el Explorador de proyectos y seleccione Nuevo, Otro objeto de repository ABAP del menú de contexto. Filtre la entrada en el diálogo que aparece, seleccione Campo de autorización de la lista y pulse Siguiente para continuar.
- 00:04:56 Introduzca un nombre en el asistente de creación y pulse Siguiente. Asigne una orden de transporte y pulse Finalizar.
- 00:05:08 El nuevo campo de autorización aparece en el editor. Actualice el nombre del elemento de datos creado anteriormente y grabe el nuevo campo de autorización para activarlo.
- 00:05:26 A continuación, haga clic en el enlace Crear un nuevo objeto de autorización y asígnele el campo de autorización en la siguiente área. Introduzca un nombre y una descripción en el asistente de creación y pulse Siguiente para continuar.
- 00:05:47 Asigne una orden de transporte y pulse Finalizar. El nuevo objeto de autorización aparece en el editor correspondiente.
- 00:05:56 El campo de autorización creado anteriormente y el campo de actividad por defecto se enumeran en el área Campos de autorización. Actualice las actividades permitidas en el área correspondiente en el editor, Añadir o Crear, Modificar, Visualizar y Borrar.
- 00:06:20 Grabe las modificaciones para activarlas. La clase de objeto se actualiza automáticamente.
- 00:06:29 Ahora, definamos el rol CDS para la vista BO de viaje. Haga clic con el botón derecho en el Explorador de proyectos y seleccione Nuevo Control de acceso en el menú contextual.
- 00:06:42 Introduzca un nombre y una descripción y pulse Siguiente para continuar. El proyecto, el paquete y la entidad protegida se han asignado automáticamente en el asistente de creación.
- 00:07:02 Asigne una orden de transporte y pulse Siguiente. Se proporcionan varias plantillas de control de acceso para su comodidad.
- 00:07:10 Seleccione el modelo Definir rol con condiciones simples de la lista y pulse Finalizar. El rol creado aparece en el editor adecuado.
- 00:07:20 La anotación @MappingRole: true se define en la parte superior para asignar el rol CDS a cada usuario independientemente del cliente. El nombre de la entidad CDS protegida se especifica después de conceder la sentencia select on y los literales ficticios y las reglas de acceso de usuario se definen en la cláusula
- 00:07:40 WHERE. Definamos una condición literal en el elemento CurrencyCode.
- 00:07:45 Sólo se deben seleccionar registros con el código de moneda EURO. Grabe el rol CDS.
- 00:07:53 Antes de activarlo, veamos rápidamente el conjunto de resultados actual en la vista previa de datos. Seleccione la vista en el Explorador de proyectos y pulse F8.
- 00:08:03 Podemos ver registros con dólares estadounidenses como código de moneda y también podemos comprobar el número de entradas seleccionadas. También podemos ver las monedas disponibles en el diálogo de filtro.
- 00:08:25 Ahora activemos el rol. Volvamos a la vista previa de datos y actualicela.
- 00:08:44 Ahora sólo se han seleccionado registros Euro de la base de datos. Vamos a continuar y ampliar la regla de acceso con una condición PFCG para asociar el elemento de vista TravelStatus con el campo de autorización adecuado del objeto de
- 00:09:21 autorización creado anteriormente, y también restringir la verificación de control de acceso a los usuarios que tienen el valor 03 visualizado en el campo de actividad. Grabe y active el rol CDS.
- 00:09:47 Ahora volvamos a verificar la presentación preliminar de datos. Actualicelo.
- 00:09:56 No se han seleccionado datos esta vez. El motivo es que mi usuario no tiene las autorizaciones necesarias.
- 00:10:06 Pero no trataremos la creación de modelos de autorización en este curso debido a restricciones técnicas en la prueba del entorno ABAP. Vamos a realizar una vista previa de los datos en la aplicación de viajes.
- 00:10:24 Los datos aún se recuperan. El motivo es que aún no hemos definido reglas de acceso para la vista de proyección BO de viaje, que se utiliza en la definición de servicio.

- 00:10:36 Como ya se ha explicado, un rol CDS debe definirse explícitamente para cada entidad CDS. No hay herencia implícita de reglas de acceso.
- 00:10:46 Por lo tanto, ahora definamos las reglas de acceso para la vista de proyección de BO de viaje. Haga clic con el botón derecho en el Explorador de proyectos y seleccione Nuevo Control de acceso en el menú contextual.
- 00:10:58 Introduzca un nombre y una descripción y pulse Siguiente para continuar. El proyecto, el paquete y la entidad protegida se han asignado automáticamente.
- 00:11:11 Esta vez, seleccione la plantilla Definir rol con condiciones heredadas y pulse Finalizar. El rol creado aparece en el editor.
- 00:11:22 La anotación @MappingRole: true se define en la parte superior. El nombre de la entidad CDS protegida se especifica después de la sentencia select on de concesión y, en la cláusula de condición WHERE, podemos definir la entidad CDS de la que se
- 00:11:36 heredarán las condiciones. Esto significa que no es necesario volver a escribir las reglas de acceso.
- 00:11:43 Introduzca el nombre de la vista BO de viaje subyacente. Grabe y active el rol CDS.
- 00:11:51 Ahora podemos obtener una vista previa de los datos en ADT y probar la aplicación de viajes. No se han seleccionado datos ahora.
- 00:12:09 Como ya se ha mencionado, no gestionaremos la creación de modelos de autorización para otorgar a un usuario las autorizaciones necesarias en este curso. Por lo tanto, para recuperar el acceso completo a los datos, podemos añadir una condición o verdadera en la cláusula WHERE del control de acceso para la vista BO de viaje como solución
- 00:12:30 alternativa. No es necesario realizar una modificación en la capa de proyección de BO porque esta regla ampliada se heredará.
- 00:12:40 Grabe y active el rol CDS. Ahora podemos obtener una vista previa de los datos en ADT y en la aplicación de viajes.
- 00:13:08 Eso es todo. Eso fue para la demostración.
- 00:13:13 Ahora estamos al final de esta unidad. Permítanme hacer una breve recapitulación.
- 00:13:18 En esta unidad, ha aprendido cuál es el enfoque de autorización declarativa del lenguaje de control de datos CDS y cómo crear roles CDS para controlar el acceso a los datos. Encontrará más información sobre esta unidad en el apéndice de esta presentación.
- 00:13:39 Ahora estamos al final de la semana dos del curso de openSAP Creación de aplicaciones con el modelo de programación de aplicaciones RESTful ABAP. Gracias por escucharle y verle la próxima semana donde habilitaremos el comportamiento transaccional de nuestra aplicación de informes de lista de viajes de SAP Fiori.

[www.sap.com/contactsap](http://www.sap.com/contactsap)

© 2020 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See [www.sap.com/copyright](http://www.sap.com/copyright) for additional trademark information and notices.