Conceptos

ONTAP Select

NetApp
April 12, 2024

Tabla de contenidos

Conceptos

Base de servicios web DE REST

La transferencia de estado representacional (REST) es un estilo para crear aplicaciones web distribuidas. Cuando se aplica al diseño de una API de servicios web, establece un conjunto de tecnologías y prácticas recomendadas para exponer recursos basados en servidor y administrar sus estados. Usa los protocolos y estándares más habituales para proporcionar una base flexible para la implementación y gestión de clústeres de ONTAP Select.

Con las restricciones clásicas y a la arquitectura

EL RESTO fue articulado formalmente por Roy Fielding en su doctorado "disertación" En UC Irvine en 2000. Define un estilo arquitectónico a través de un conjunto de restricciones, que han mejorado colectivamente las aplicaciones basadas en web y los protocolos subyacentes. Estas restricciones establecen una aplicación de servicios web RESTful basada en una arquitectura de cliente/servidor que utiliza un protocolo de comunicación sin estado.

Recursos y representación estatal

Los recursos son los componentes básicos de un sistema basado en la Web. Al crear una aplicación DE SERVICIOS web DE REST, las tareas de diseño más tempranas incluyen:

- Identificación de recursos basados en sistemas o servidores
  Cada sistema utiliza y mantiene los recursos. Un recurso puede ser un archivo, una transacción comercial, un proceso o una entidad administrativa. Una de las primeras tareas en el diseño de una aplicación basada en servicios web DE REST es identificar los recursos.
- Definición de estados de recursos y operaciones estatales asociadas
  Los recursos siempre se encuentran en uno de un número limitado de estados. Los estados, así como las operaciones asociadas utilizadas para afectar los cambios de estado, deben estar claramente definidos.

Los mensajes se intercambian entre el cliente y el servidor para acceder y cambiar el estado de los recursos según el modelo genérico CRUD (Crear, Leer, Actualizar y Eliminar).

Extremos de URI

Todos los recursos REST deben definirse y ponerse a disposición mediante un esquema de direccionamiento bien definido. Los extremos en los que se encuentran e identifican los recursos utilizan un identificador uniforme de recursos (URI). El URI proporciona un marco general para crear un nombre único para cada recurso de la red. El Localizador uniforme de recursos (URL) es un tipo de URI que se utiliza con los servicios web para identificar y acceder a los recursos. Los recursos normalmente se exponen en una estructura jerárquica similar a un directorio de archivos.

Mensajes HTTP

El Protocolo de transferencia de hipertexto (HTTP) es el protocolo utilizado por el cliente y servidor de servicios web para intercambiar mensajes de solicitud y respuesta sobre los recursos. Como parte del diseño de una aplicación de servicios web, los verbos HTTP (como GET y POST) se asignan a los recursos y a las acciones de administración de estado correspondientes.

HTTP no tiene estado. Por lo tanto, para asociar un conjunto de solicitudes y respuestas relacionadas en una transacción, se debe incluir información adicional en los encabezados HTTP transportados con los flujos de datos de solicitud/respuesta.

Formato JSON

Aunque la información se puede estructurar y transferir entre un cliente y un servidor de varias maneras, la opción más popular (y la que se usa con la API REST de deploy) es la notación de objetos JavaScript (JSON). JSON es un estándar del sector para representar estructuras de datos simples en texto sin formato y se utiliza para transferir información de estado que describe los recursos.

Cómo acceder a la API de implementación

Debido a la flexibilidad inherente de los servicios web de REST, se puede acceder a la API de implementación de ONTAP Select de varias maneras diferentes.

Implemente la interfaz de usuario nativa de la utilidad

La forma principal de acceder a la API se realiza a través de la interfaz de usuario web de implementación de ONTAP Select. El navegador realiza llamadas a la API y reformatea los datos según el diseño de la interfaz de usuario. También se accede a la API a través de la interfaz de línea de comandos de Deploy Utility.

Página de documentación en línea de ONTAP Select Deploy

La página de documentación en línea ONTAP Select Deploy proporciona un punto de acceso alternativo cuando se utiliza un explorador. Además de proporcionar una forma de ejecutar directamente llamadas API individuales, la página también incluye una descripción detallada de la API, incluidos los parámetros de entrada y otras opciones para cada llamada. Las llamadas API se organizan en varias categorías o áreas funcionales diferentes.

Programa personalizado

Puede acceder a la API de implementación utilizando cualquiera de los diferentes lenguajes y herramientas de programación. Entre las opciones más populares se incluyen Python, Java y curl. Programa, script o herramienta que usa la API actúa como cliente DE servicios web REST. Utilizar un lenguaje de programación le permite comprender mejor la API y proporciona una oportunidad para automatizar las puestas en marcha de ONTAP Select.

Implemente el control de versiones de API

La API DE REST que se incluye con la implementación de ONTAP Select tiene asignado un número de versión. El número de versión de la API es independiente del número de versión de implementación. Debe conocer la versión de la API que se incluye con su versión de implementación y cómo esto puede afectar al uso que hace de la API.

La versión actual de la utilidad de administración de implementación incluye la versión 3 de la API DE REST. Las versiones anteriores de la utilidad Deploy incluyen las siguientes versiones de API:

Implemente 2.8 y posterior
: ONTAP Select Deploy 2.8 y todas las versiones posteriores incluyen la versión 3 de la API DE REST.

Puesta en marcha 2.7.2 y anteriores

ONTAP Select Deploy 2.7.2 y todas las versiones anteriores incluyen la versión 2 de la API DE REST.

> Las versiones 2 y 3 de la API REST no son compatibles. Si actualiza a Deploy 2.8 o posterior desde una versión anterior que incluya la versión 2 de la API, debe actualizar cualquier código existente que acceda directamente a la API, así como cualquier script que utilice la interfaz de línea de comandos.

Características operativas básicas

Mientras QUE REST establece un conjunto común de tecnologías y prácticas recomendadas, los detalles de cada API pueden variar en función de las opciones de diseño. Debe tener en cuenta los detalles y las características operativas de la API de implementación de ONTAP Select antes de usar la API.

Host de hipervisor frente a nodo ONTAP Select

Un *hypervisor host* es la plataforma de hardware principal que aloja una máquina virtual ONTAP Select. Cuando se implementa una máquina virtual ONTAP Select y está activa en un host de hipervisor, la máquina virtual se considera un *ONTAP Select node*. Con la versión 3 de la API DE REST de implementación, los objetos de host y nodo son distintos y separados. Esto permite una relación de uno a varios, donde uno o varios nodos ONTAP Select pueden ejecutarse en el mismo host de hipervisor.

Identificadores de objeto

A cada instancia u objeto de recurso se le asigna un identificador único cuando se crea. Estos identificadores son globalmente únicos dentro de una instancia específica de la implementación de ONTAP Select. Después de emitir una llamada API que crea una nueva instancia de objeto, el valor de id asociado se devuelve al llamador en la `location` Encabezado de la respuesta HTTP. Puede extraer el identificador y utilizarlo en llamadas posteriores cuando haga referencia a la instancia del recurso.

> El contenido y la estructura interna de los identificadores de objeto pueden cambiar en cualquier momento. Solo se deben usar los identificadores en las llamadas API aplicables según sea necesario cuando se hacen referencia a los objetos asociados.

Identificadores de solicitudes

A cada solicitud API de éxito se le asigna un identificador único. El identificador se muestra en la `request-id` Encabezado de la respuesta HTTP asociada. Puede utilizar un identificador de solicitud para hacer referencia colectivamente a las actividades de una única transacción específica de solicitud y respuesta de API. Por ejemplo, puede recuperar todos los mensajes de eventos de una transacción basándose en el ID de solicitud

Llamadas síncronas y asíncronas

Hay dos formas principales de que un servidor realice una solicitud HTTP recibida desde un cliente:

- Síncrona
  El servidor realiza la solicitud inmediatamente y responde con un código de estado 200, 201 o 204.
- Asíncrona
  El servidor acepta la solicitud y responde con un código de estado 202. Esto indica que el servidor ha

aceptado la solicitud de cliente y ha iniciado una tarea en segundo plano para completar la solicitud. El éxito o el fallo final no están disponibles de forma inmediata y se deben determinar mediante llamadas API adicionales.

Confirmar que se ha completado un trabajo de ejecución prolongada

Por lo general, cualquier operación que puede demorar mucho tiempo en completarse se procesa de forma asíncrona mediante a.
tarea en segundo plano en el servidor. Con la API de REST DE implementación, cada tarea en segundo plano está anclada por a.
Objeto de trabajo que realiza un seguimiento de la tarea y proporciona información, como el estado actual. Un objeto Job,
Incluido su identificador único, se devuelve en la respuesta HTTP después de crear una tarea en segundo plano.

Puede consultar el objeto Job directamente para determinar el éxito o el error de la llamada API asociada. Consulte *procesamiento asíncrono mediante el objeto Job* para obtener información adicional.

Además de utilizar el objeto Job, hay otras formas de determinar el éxito o el fracaso de un solicitud, que incluye:

- Mensajes de eventos
  Puede recuperar todos los mensajes de evento asociados a una llamada API específica mediante el identificador de solicitud devuelto con la respuesta original. Los mensajes de eventos normalmente contienen una indicación de éxito o fallo, y también pueden ser útiles al depurar una condición de error.
- Estado o estado del recurso
  Varios de los recursos mantienen un valor de estado o estado que puede consultar para determinar indirectamente el éxito o el fallo de una solicitud.

Seguridad

La API de implementación utiliza las siguientes tecnologías de seguridad:

- Seguridad de la capa de transporte
  Todo el tráfico enviado a través de la red entre el servidor de implementación y el cliente se cifra a través de TLS. No se admite el uso del protocolo HTTP a través de un canal no cifrado. Se admite la versión 1.2 de TLS.
- Autenticación HTTP
  La autenticación básica se utiliza para cada transacción de API. A cada solicitud se agrega un encabezado HTTP, que incluye el nombre de usuario y la contraseña en una cadena base64.

Transacción de API de solicitud y respuesta

Cada llamada de API de implementación se realiza como una solicitud HTTP a la máquina virtual de implementación, que genera una respuesta asociada al cliente. Este par de solicitud/respuesta se considera una transacción de API. Antes de utilizar la API de implementación, debería estar familiarizado con las variables de entrada disponibles para controlar una solicitud y el contenido del resultado de la respuesta.

Variables de entrada que controlan una solicitud API

Puede controlar cómo se procesa una llamada API mediante parámetros definidos en la solicitud HTTP.

Solicitar encabezados

Debe incluir varios encabezados en la solicitud HTTP, incluidos:

- tipo de contenido
  Si el cuerpo de la solicitud incluye JSON, esta cabecera debe establecerse en application/json.
- aceptar
  Si el cuerpo de la respuesta incluirá JSON, este encabezado debe establecerse en application/json.
- autorización
  La autenticación básica se debe establecer con el nombre de usuario y la contraseña codificados en una cadena base64.

Solicitar el cuerpo

El contenido del cuerpo de la solicitud varía en función de la llamada específica. El cuerpo de la solicitud HTTP consta de uno de los siguientes elementos:

- Objeto JSON con variables de entrada (como el nombre de un clúster nuevo)
- Vacío

Filtrar objetos

Al emitir una llamada API que utilice GET, puede limitar o filtrar los objetos devueltos en función de cualquier atributo. Por ejemplo, puede especificar un valor exacto para que coincida:

```
<field>=<query value>
```

Además de una coincidencia exacta, hay otros operadores disponibles para devolver un conjunto de objetos sobre un rango de valores. ONTAP Select admite los operadores de filtrado que se muestran a continuación.

Operador	Descripción
=	Igual a.
<	Menor que
>	Mayor que
⇐	Menor o igual que
>=	Mayor o igual que
	O.
!	No es igual a.
*	Comodín codicioso

También puede devolver un conjunto de objetos basándose en si se establece o no un campo específico utilizando la palabra clave null o su negación (!null) como parte de la consulta.

Selección de campos de objeto

De forma predeterminada, al emitir una llamada API mediante GET, sólo se devuelven los atributos que identifican de forma exclusiva el objeto o los objetos. Este conjunto mínimo de campos actúa como clave para cada objeto y varía según el tipo de objeto. Puede seleccionar propiedades de objeto adicionales mediante el parámetro de consulta Campos de las siguientes formas:

- Campos baratos
  Especifique `fields=*` para recuperar los campos de objeto que se mantienen en la memoria del servidor local o que requieren poco procesamiento para acceder.
- Campos caros
  Especifique `fields=**` para recuperar todos los campos de objeto, incluidos los que requieren procesamiento de servidor adicional para tener acceso.
- Selección de campo personalizado
  Uso `fields=FIELDNAME` para especificar el campo exacto que desea. Al solicitar varios campos, los valores deben separarse con comas sin espacios.

Como práctica recomendada, siempre debe identificar los campos específicos que desea. Sólo debe recuperar el conjunto de campos baratos o caros cuando sea necesario. NetApp determina la clasificación económica y cara basándose en el análisis del rendimiento interno. La clasificación de un campo determinado puede cambiar en cualquier momento.

Ordenar objetos en el conjunto de salida

Los registros de una colección de recursos se devuelven en el orden predeterminado definido por el objeto. Puede cambiar el orden utilizando el parámetro de consulta Order_by con el nombre del campo y la dirección de ordenación de la siguiente manera:
`order_by=<field name> asc|desc`

Por ejemplo, puede ordenar el campo de tipo en orden descendente seguido de id en orden ascendente:
`order_by=type desc, id asc`

Cuando se incluyan varios parámetros, los campos deben separarse con una coma.

Paginación

Al emitir una llamada API mediante GET para acceder a una colección de objetos del mismo tipo, todos los objetos coincidentes se devuelven de forma predeterminada. Si es necesario, puede limitar el número de registros devueltos mediante el parámetro de consulta max_Records con la solicitud. Por ejemplo:
`max_records=20`

Si es necesario, puede combinar este parámetro con otros parámetros de consulta para restringir el conjunto de resultados. Por ejemplo, el siguiente muestra hasta 10 eventos del sistema generados después de la hora especificada:
`time⇒ 2019-04-04T15:41:29.140265Z&max_records=10`

Puede emitir varias solicitudes para páginas a través de los eventos (o cualquier tipo de objeto). Cada llamada API posterior debe utilizar un nuevo valor de tiempo basado en el último evento del último conjunto de resultados.

Interpretar una respuesta API

Cada solicitud de API genera una respuesta al cliente. Puede examinar la respuesta para determinar si se ha realizado correctamente y recuperar datos adicionales según sea necesario.

Código de estado HTTP

A continuación se describen los códigos de estado HTTP utilizados por la API de REST de despliegue.

Codificación	Significado	Descripción
200	DE ACUERDO	Indica que las llamadas que no crean un objeto nuevo se han realizado correctamente.
201	Creado	Se ha creado correctamente un objeto; el encabezado de respuesta de ubicación incluye el identificador único del objeto.
202	Aceptado	Se inició un trabajo en segundo plano de ejecución prolongada para realizar la solicitud, pero la operación aún no se ha completado.
400	Solicitud incorrecta	La entrada de la solicitud no se reconoce o no es apropiada.
403	Prohibido	Se deniega el acceso debido a un error de autorización.
404	No encontrado	El recurso al que se hace referencia en la solicitud no existe.
405	Método no permitido	El verbo HTTP de la solicitud no es compatible con el recurso.
409	Conflicto	Error al intentar crear un objeto porque el objeto ya existe.
500	Error interno	Se ha producido un error interno general en el servidor.
501	No implementada	El URI es conocido pero no es capaz de realizar la solicitud.

Encabezados de respuesta

Se incluyen varios encabezados en la respuesta HTTP generada por el servidor de implementación, entre los que se incluyen:

- id de solicitud
  A cada solicitud API correcta se le asigna un identificador de solicitud único.
- ubicación
  Cuando se crea un objeto, la cabecera de ubicación incluye la dirección URL completa del nuevo objeto, incluido el identificador de objeto único.

Cuerpo de respuesta

El contenido de la respuesta asociada a una solicitud API varía en función del objeto, el tipo de procesamiento y el éxito o el fallo de la solicitud. El cuerpo de la respuesta se representa en JSON.

- Un solo objeto
  Un solo objeto se puede devolver con un conjunto de campos basados en la solicitud. Por ejemplo, se puede usar GET para recuperar las propiedades seleccionadas de un clúster mediante el identificador único.
- Varios objetos
  Se pueden devolver varios objetos de una colección de recursos. En todos los casos, existe un formato coherente utilizado, con `num_records` indica el número de registros y registros que contienen una matriz

de las instancias de objeto. Por ejemplo, puede recuperar todos los nodos definidos en un clúster específico.

- Objeto de trabajo
  Si una llamada API se procesa de forma asíncrona, se devuelve un objeto Job que ancla la tarea en segundo plano. Por ejemplo, la solicitud POST utilizada para implementar un clúster se procesa de forma asíncrona y devuelve un objeto Job.
- Objeto de error
  Si se produce un error, siempre se devuelve un objeto error. Por ejemplo, recibirá un error al intentar crear un clúster con un nombre que ya existe.
- Vacío
  En ciertos casos, no se devuelven datos y el cuerpo de la respuesta está vacío. Por ejemplo, el cuerpo de respuesta está vacío después de utilizar DELETE para eliminar un host existente.

Procesamiento asíncrono mediante el objeto de trabajo

Algunas de las llamadas de implementación de API, especialmente las que crean o modifican un recurso, pueden tardar más tiempo en completarse que otras llamadas. La implementación de ONTAP Select procesa estas solicitudes de ejecución prolongada de forma asíncrona.

Solicitudes asincrónicas descritas mediante el objeto Job

Después de realizar una llamada API que se ejecuta de forma asíncrona, el código de respuesta HTTP 202 indica que la solicitud se ha validado y aceptado correctamente, pero que aún no se ha completado. La solicitud se procesa como una tarea en segundo plano que continúa ejecutándose después de la respuesta HTTP inicial al cliente. La respuesta incluye el objeto Job anclando la solicitud, incluyendo su identificador único.

Consulte la página de documentación en línea de implementación de ONTAP Select para determinar qué llamadas API funcionan de forma asíncrona.

Consulte el objeto Job asociado a una solicitud API

El objeto Job devuelto en la respuesta HTTP contiene varias propiedades. Puede consultar la propiedad state para determinar si la solicitud se completó correctamente. Un objeto Job puede estar en uno de los siguientes estados:

- En cola
- Ejecutando
- Correcto
- Fallo

Existen dos técnicas que se pueden utilizar al sondear un objeto Job para detectar un estado de terminal para la tarea, ya sea con éxito o con un error:

- Solicitud de sondeo estándar
  El estado actual del trabajo se devuelve inmediatamente
- Solicitud de sondeo larga

El estado del trabajo solo se devuelve cuando se produce alguna de las siguientes situaciones:

- El estado ha cambiado más recientemente que el valor de fecha y hora proporcionado en la solicitud de sondeo
- El valor de tiempo de espera ha caducado (de 1 a 120 segundos)

Sondeo estándar y sondeo largo Utilice la misma llamada API para consultar un objeto Job. Sin embargo, una solicitud de sondeo larga incluye dos parámetros de consulta: `poll_timeout` y.. `last_modified`.

Siempre debe utilizar los sondeos largos para reducir la carga de trabajo en la máquina virtual de implementación.

Procedimiento general para emitir una solicitud asincrónica

Puede utilizar el siguiente procedimiento de alto nivel para completar una llamada API asíncrona:

1. Emita la llamada de API asíncrona.
2. Reciba una respuesta HTTP 202 que indique la aceptación correcta de la solicitud.
3. Extraiga el identificador del objeto Job del cuerpo de respuesta.
4. Dentro de un bucle, realice lo siguiente en cada ciclo:
   a. Obtener el estado actual del trabajo con una solicitud de sondeo largo
   b. Si el trabajo se encuentra en un estado que no es terminal (en cola, en ejecución), vuelva a realizar el bucle.
5. Deténgase cuando el trabajo alcance un estado terminal (correcto, fallo).

Información de copyright

Información de la marca comercial