

RevokeToken

Para generar los casos de prueba se ha seleccionado la técnica de clases de equivalencia y valores límite. En el fichero de entrada contamos con el campo “Reason”, que tiene una restricción en cuanto a caracteres, por lo que si no utilizamos valores límite no estaríamos controlando todos los posibles casos de error de nuestro código.

Clases de equivalencia y valores límite

Las clases de equivalencia consisten en agrupar todos aquellos datos que se tratan de igual manera o que deberían proporcionar el mismo resultado. Por otro lado, los valores límite serán, como su nombre indica, los valores máximos y mínimos de esas clases de equivalencia.

ENTRADA

Criterio/Tipo de Análisis	ID de C.E./VL	Descripción
Criterio: Existencia del fichero		
	CE-RT-V-01	El fichero existe y se encuentra en la ruta especificada
	CE-RT-NV-02	El fichero NO existe o NO se encuentra en la ruta especificada.
Criterio: Sintaxis del fichero		
	CE-RT-V-03	El fichero se encuentra en formato adecuado
	CE-RT-NV-04	El fichero no se encuentra en formato adecuado, ya que le faltan ambos corchetes
	CE-RT-NV-05	El fichero no se encuentra en formato adecuado, ya que tiene algún corchete duplicado
	CE-RT-NV-06	El fichero no se encuentra en formato adecuado, ya que le falta uno de los dos corchetes
Criterio: Semántica del fichero		

	CE-RT-V-07	Token Value existe y es correcto (es una cadena de caracteres válida)
	CE-RT-NV-08	Token Value NO existe
	CE-RT-V-09	El valor del campo “Token Value” existe y es válido
	CE-RT-NV-10	El valor del campo “Token Value” no existe
	CE-RT-NV-11	El valor del campo “Token Value” existe y no es válido
	CE-RT-V-12	Type of revocation existe y es correcto
	CE-RT-NV-13	Type of revocation NO existe
	CE-RT-V-14	El valor del campo “Type of revocation ” existe y es válido
	CE-RT-NV-15	El valor del campo “Type of revocation ” no existe
	CE-RT-NV-16	El valor del campo “Type of revocation ” existe y no es válido
	CE-RT-V-17	Reason existe y es correcto
	CE-RT-NV-8	Reason no existe
	CE-RT-V-19	El valor del campo “Reason” existe y es válido
	CE-RT-NV-20	El valor del campo “Reason” no existe
	CE-RT-NV-21	El valor del campo “Reason” existe y no es válido (contiene símbolos)
	CE-RT-V-22	La longitud del valor del campo “Reason” es de menos de 100 caracteres
	CE-RT-NV-23	La longitud del valor del campo “Reason” es mayor de 100 caracteres
Criterio: funcionalidad del token		
	CE-RT-V-24	El valor de Token value existe en la TokenStore

	CE-RT-NV-25	El valor de Token value no existe en la TokenStore
	CE-RT-V-26	El valor de Token Value está vigente
	CE-RT-NV-27	El valor de Token Value ya ha caducado
	CE-RT-V-28	El valor de token value no ha sido revocado en la misma modalidad
	CE-RT-NV-29	El valor de token value ya ha sido revocado en la misma modalidad
Análisis de valores límite		
	VL-RT-V-30	La longitud del valor del campo "Reason" es 0
	VL-RT-V-31	La longitud del valor del campo "Reason" es 1
	VL-RT-V-32	La longitud del valor del campo "Reason" es 99
	VL-RT-V-33	La longitud del valor del campo "Reason" es 100
	VL-RT-NV-34	La longitud del valor del campo "Reason" es 101

SALIDA

ID de C.E./VL	Descripción
CE-RT-NV-35	El String de salida no existe.
CE-RT-NV-36	El String de salida existe y es erróneo.
CE-RT-V-37	El String de salida existe y es válido.
CE-RT-NV-38	No se encuentra el fichero con los datos de entrada.
CE-RT-NV-39	El fichero de entrada no contiene datos.
CE-RT-NV-40	El fichero de entrada no tiene el formato esperado.
CE-RT-NV-41	Error interno en la generación de la salida.

- Casos de Prueba de Entrada

ENTRADA

Test Case ID	Parámetros de entrada (Fichero xxxxx.json de nuestro código)	Clases o Valores que prueba
CP-RT-01	Fichero existe JSON es correcto Campos correcto (Correct.json)	CE-RT-V-01 CE-RT-V-03 CE-RT-V-07 CE-RT-V-09 CE-RT-V-12 CE-RT-V-14 CE-RT-V-17 CE-RT-V-19 CE-RT-V-22 CE-RT-V-24 CE-RT-V-26 CE-RT-V-28
CP-RT-02	Fichero no existe (FileIsMissing.json)	CE-RT-NV-02
CP-RT-03	Fichero no se encuentra en formato adecuado (faltan ambos corchetes) (BracketsAreMissing.json)	CE-RT-NV-04
CP-RT-04	Fichero no se encuentra en formato adecuado (corchetes duplicados) (BracketIsDuplicated.json)	CE-RT-NV-05
CP-RT-05	Fichero no se encuentra en formato adecuado (faltan un corchete) (BracketIsMissing.json)	CE-RT-NV-06
CP-RT-06	Falta "Token Value" (TokenValueIsMissing.json)	CE-RT-NV-08
CP-RT-07	Falta el valor de "Token Value" (TokenValueValueIsMissing.json)	CE-RT-NV-10
CP-RT-08	El valor del campo "Token Value" no es válido (TokenValueValueNotValid.json)	CE-RT-NV-11
CP-RT-09	Falta "Type Of Revocation" (TypeOfRevocationIsMissing.json)	CE-RT-NV-13
CP-RT-10	Falta el valor de "Type Of Revocation"	CE-RT-NV-15

	(TypeOfRevocationValueIsMissing.json)	
CP-RT-11	El valor del campo “Type Of Revocation” no es válido (TypeOfRevocationValueNotValid.json)	CE-RT-NV-16
CP-RT-12	Falta “Reason” (ReasonIsMissing.json)	CE-RT-NV-18
CP-RT-13	Falta el valor de “Reason” (ReasonIsMissing.json)	CE-RT-NV-20
CP-RT-14	El valor de “Reason” no es válido (ReasonNotValid.json)	CE-RT-NV-21
CP-RT-15	La longitud del valor del campo “Reason” no es válida (ReasonLengthNotValid.json)	CE-RT-NV-23
CP-RT-16	El valor de Token value no existe en la TokensStore (TokenValueNotInTokensStore.json)	CE-RT-NV-25
CP-RT-17	El valor de Token Value ya ha caducado (TokenValueExpired.json)	CE-RT-NV-27
CP-RT-18	El valor de token value ya ha sido revocado en la misma modalidad (TokenValueRevoked.json)	CE-RT-NV-29
Casos de prueba valores límite		
CP-RT-26	La longitud del valor del campo “Reason” es 1 (VLReason1.json)	VL-RT-V-30
CP-RT-27	La longitud del valor del campo “Reason” es 2 (VLReason2.json)	VL-RT-V-31
CP-RT-28	La longitud del valor del campo “Device name” es 99 (VLReason99.json)	VL-RT-V-32

CP-RT-29	La longitud del valor del campo "Device name" es 100 (VLReason100.json)	VL-RT-V-33
CP-RT-30	La longitud del valor del campo "Device name" es 101 (VLReason101.json)	VL-RT-NV-34

SALIDA

Test Case ID	Parámetros de entrada (Fichero xxxxx.json de nuestro código)	Clases o Valores que prueba
CP-RT-31	El String de salida no existe.	CE-RT-NV-35
CP-RT-32	El String de salida existe y es erróneo.	CE-RT-NV-36
CP-RT-33	El String de salida existe y es válido.	CE-RT-V-37
CP-RT-34	No se encuentra el fichero con los datos de entrada.	CE-RT-NV-38
CP-RT-35	El fichero de entrada no contiene datos.	CE-RT-NV-39
CP-RT-36	El fichero de entrada no tiene el formato esperado.	CE-RT-NV-40
CP-RT-37	Error interno en la generación de la salida.	CE-RT-NV-41

- Casos Prueba de Salida

Los tests de salida están condicionados directamente por los de entrada, por lo que sus casos de prueba están totalmente controlados por estos. Nos hemos dado cuenta al ir a implementarlos y al ver esto hemos decidido no codificarlos.

ExecuteAction

Se ha seleccionado la técnica de prueba de análisis sintáctico para generar los casos de prueba. Esta técnica es muy metódica, de esta manera conseguiremos reducir al máximo la posibilidad de cometer un error durante su realización. Además, cubre una gran cantidad de fallos posibles.

Análisis Sintáctico

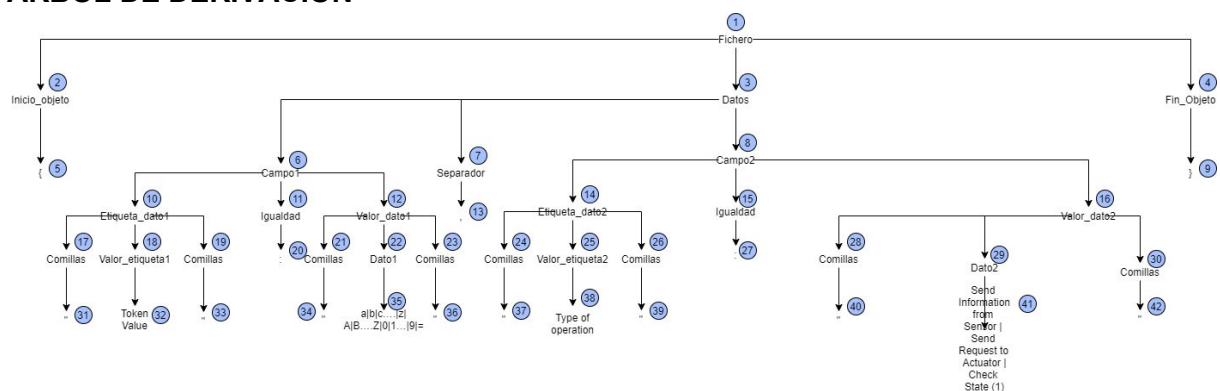
En este caso primero es necesario definir la gramática para representar los valores esperados en el fichero de entrada descritos en el enunciado. A continuación, se creará el árbol de

derivación a partir de esta gramática que acabamos de obtener, y ya por último se codificarán los tests necesarios siguiendo este árbol.

GRAMÁTICA

- Fichero ::= Inicio_objeto Datos Fin_Objeto
- Inicio_objeto ::= {
- Fin_Objeto ::= }
- Datos ::= Campo1 Separador Campo2
- Campo1 ::= Etiqueta_dato1 Igualdad Valor_dato1
- Campo2 ::= Etiqueta_dato2 Igualdad Valor_dato2
- Etiqueta_dato1 ::= Comillas Valor_etiqueta1 Comillas
- Comillas ::= “”
- Igualdad ::= ‘:’
- Separador ::= ‘,’
- Valor_etiqueta1 ::= Token Value
- Valor_dato1 ::= Comillas Dato1 Comillas
- Dato1 ::= a|b|c...|z|A|B...Z|0|1...|9|=
- Etiqueta_dato2 ::= Comillas Valor_etiqueta2 Comillas
- Valor_etiqueta2 ::= Type of operation
- Valor_dato2 ::= comillas Dato2 comillas
- Dato2 ::= Send Information from Sensor | Send Request to Actuator | Check State (1)

ÁRBOL DE DERIVACIÓN



- Casos de Prueba de Entrada

ENTRADA

Test Case ID	Parámetros de entrada (Fichero xxxxx.json de nuestro código)	Nodos Cubiertos
AS-EA-V-01	Fichero existe JSON es correcto Campos correcto (Correct.json)	
AS-EA-NV-02	Nodo no terminal 1 eliminado	1, 2, 3, 4
AS-EA-NV-03	Nodo no terminal 2 eliminado	2, 5

AS-EA-NV-04	Nodo no terminal 2 duplicado	2, 5
AS-EA-NV-05	Nodo no terminal 3 eliminado	3, 6, 7, 8
AS-EA-NV-06	Nodo no terminal 3 duplicado	3, 6, 7, 8
AS-EA-NV-07	Nodo no terminal 4 eliminado	4, 9
AS-EA-NV-08	Nodo no terminal 4 duplicado	4, 9
AS-EA-NV-09	Nodo no terminal 6 eliminado	6, 10, 11, 12 8, 14, 15, 16
AS-EA-NV-10	Nodo no terminal 6 duplicado	6, 10, 11, 12 8, 14, 15, 16
AS-EA-NV-11	Nodo no terminal 7 eliminado	7, 13
AS-EA-NV-12	Nodo no terminal 7 duplicado	7, 13
AS-EA-NV-13	Nodo no terminal 10 eliminado	10, 17, 18, 19 12, 21, 22, 23 14, 24, 25, 26 16, 28, 29, 30
AS-EA-NV-14	Nodo no terminal 10 duplicado	10, 17, 18, 19 12, 21, 22, 23 14, 24, 25, 26 16, 28, 29, 30
AS-EA-NV-15	Nodo no terminal 11 eliminado	11, 20 15, 27
AS-EA-NV-16	Nodo no terminal 11 duplicado	11, 20 15, 27
AS-EA-NV-17	Nodo no terminal 17 eliminado	17, 31 19, 33 21, 34 23, 36 24, 37 26, 39 28, 40 30, 42
AS-EA-NV-18	Nodo no terminal 17 duplicado	17, 31 19, 33 21, 34 23, 36 24, 37

		26, 39 28, 40 30, 42
AS-EA-NV-19	Nodo no terminal 18 eliminado	18, 32
AS-EA-NV-20	Nodo no terminal 18 duplicado	18, 32
AS-EA-NV-21	Nodo no terminal 22 eliminado	22, 35
AS-EA-NV-22	Nodo no terminal 22 duplicado	22, 35
AS-EA-NV-23	Nodo no terminal 25 eliminado	25, 38
AS-EA-NV-24	Nodo no terminal 25 duplicado	25, 38
AS-EA-NV-25	Nodo no terminal 29 eliminado	29, 41
AS-EA-NV-26	Nodo no terminal 29 duplicado	29, 41
Nodos terminales		
AS-EA-NV-27	Nodo terminal 5 no es “{“	5, 9
AS-EA-NV-28	Nodo terminal 31 no es “	31, 33, 34, 36, 37, 39, 40, 42
AS-EA-NV-29	Nodo terminal 32 no es “Token Value”	32
AS-EA-NV-30	Nodo terminal 20 no es “.”	20, 27
AS-EA-NV-31	Nodo terminal 35 no es una cadena de caracteres [a-z][A-Z][0-9][=]	35
AS-EA-NV-32	Nodo terminal 13 no es “,”	13
AS-EA-NV-33	Nodo terminal 38 no es “Type of operation”	38
AS-EA-NV-34	Nodo terminal 41 no es “Send Information from Sensor” o “Send Request to Actuator” o “Check State”	41
Específicos de la funcionalidad		
AS-EA-NV-35	El token que se quiere utilizar no existe.	35
AS-EA-NV-36	El token que se quiere utilizar no es válido.	35

AS-EA-NV-37	La operación solicitada no se puede realizar con el token adjuntado.	35, 41
-------------	--	--------

OBSERVACIONES DE LOS TEST

Ante un par de casos en los que se duplican nodos el test no se pasa satisfactoriamente. Esto es debido a que el parser proporcionado es relajado y únicamente comprueba su primera aparición, si ese nodo está le da igual que esté duplicado, pues al encontrar el primero finaliza.

Esos test han sido comentados como corresponde en el código.

SALIDA

Test Case ID	Parámetros de entrada (Fichero xxxxx.json de nuestro código)
CP-EA-38	El booleano/excepción de salida no existe.
CP-EA-39	El booleano/excepción de salida existe y es erróneo.
CP-EA-40	El booleano/excepción de salida existe y es válido.
CP-EA-41	No se encuentra el fichero con los datos de entrada.
CP-EA-42	El fichero de entrada no contiene datos.
CP-EA-43	El fichero de entrada no tiene el formato esperado.
CP-EA-44	Error interno en la generación de la salida.

- Casos de Prueba de Salida

Hemos decidido devolver true en caso de que la operación sea aceptada para el token adjuntado. En caso contrario en vez de devolver false se retorna una excepción.

Los tests de salida están condicionados directamente por los de entrada, por lo que sus casos de prueba están totalmente controlados por estos. Nos hemos dado cuenta al ir a implementarlos y al ver esto hemos decidido no codificarlos.