

Práca s XML – validácia XML dokumentu XSD schémou v Java

Príklad: **XmlValidator** (runtime validácia XML súboru)

Používa validačný balík: javax.xml.validation

Dokumentácia: <https://docs.oracle.com/javase/7/docs/api/javax/xml/validation/package-summary.html>

Úloha na cvičenie: napr. vytvoriť také XML aby prešlo validáciou.

Práca s BASE64 v JAVE

- BASE64 je kódovanie textu, resp. binárnych hodnôt do 64 ascii znakov: A-Z, a-z, 0-9, +, /
- tri 8 bitové symboly (napr. ascii znaky) sú kódované do štyroch BASE64 znakov ($8 \times 3 = 6 \times 4$)

• napr.:

- „Man“ je kódovaný do TWFu.
- $M=77_{10}=01001101_2$, $a=97=01100001$, $n=110=01101110$
- 01001101 01100001 01101110
- 010011 010110 000101 101110
- 19 22 5 46
-

• Príklad: **Base64json**

• Používa sa na zakódovanie binárnych informácií do textu

• Používa sa zakódovanie mena+hesla pri HTTP basic autentifikácii.

- `byte[] encodedBytes = Base64.getEncoder().encode((myUsername+":"+myUpass).getBytes());`

Base64 Encoding Table

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

Čítanie hesla z konzoly bez zobrazenia:

- Príklad: Base64json
- Metóda: GetPasswdFromConsole

Práca s HTTP Metódami (napr. GET)

- Príklad: Base64json
- Metóda: GetInfoFromLMS

Práca s JSONom

- JSON = java script object notation
- Otvorený štandard (RFC 8259) na kódovanie objektov, textový čitateľný
- Príklad jsonu:

```
{  
  "LrsAuthorization":"Basic dGVzdDpBaiMyNmpN",  
  "LearnerDisplayName":"Trainee 0 NEWTON",  
  "role":"Patient",  
  "Registration":"ec531277-b57b-4c15-8d91-d292c5b2b8f1",  
  "LearnerEmail":"trainee_newton0@local.com",  
  "LrsEndpoint":https://newtelp.eu/lrsx/xAPI/  
}
```

- Príklad: Base64json.zip (príklad na dekodovanie)
- Príklad2: jsonCreate.java (príklad na enkodovanie)

Práca s CSV

- CSV=comma separated value
- Napr. `jsonCreate.java` obsahuje
 - Class `CsvReader` (jednoduchá trieda na čítanie csv po riadkoch)
 - class `CsvUsers` extends `CsvReader` (jednoduchá trieda na overovanie hesla konkrétného užívateľa)

Ako urobiť server?

- Čo takto si urobiť vlastný HTTP server?:
 - napr. <https://www.ssaurel.com/blog/create-a-simple-http-web-server-in-java/>
 - Využíva iba sockety a io streamy ...
 - napr. <https://gist.github.com/gitawego/14e15456e4d36a26aac7695648565329>
 - Využíva `com.sun.net.httpserver`
 - Príklad: `myHttpServer.zip`
- ...

Ako Komunikovať s poznamkovacím HTTP serverom

- Napr. pomocou HTTP GET/POST a používať obsah json, xml, text
 - napr. v `jsonCreate` posielame json payload na server:
 - `con.setRequestProperty("Content-Type", "application/json; charset=UTF-8");`
- Využívať kódovanie parametrov v URL
 - <http://server/test?prikaz=dajZoznamPoznamok>
 - <http://server/test?prikaz=ZaradNovuPoznamku&text=blablabla>
 - pozor na medzery, iné znaky a max. dĺžku URL

HTTP – REST a idempodencia

- REST

- REST = REpresentational State Transfer. Je to štýl architektúry pre informačné systémy pracujúcimi s distribuovanými hypermédiami
- Hlavné vlastnosti (<https://restfulapi.net>):
 - Architektúra klient–server
 - Bezstavovosť - každá požiadavka v sebe nesie kompletnú informáciu (nepoužíva sa nejaký dočasný obsah uložený na servri – napr. „session object“)
 - Kešovateľnosť odpovedí – musí byť určená
 - ...
- hovoríme aj o RESTfull API (server poskytuje služby klientom RESTfull API)
- Ako payload sa preferuje JSON pred XML
- Porovnanie REST a SOAP
 - Rest sa viaže na HTTP a jeho metódy, SOAP nie.
 - REST sa viac viaže na základne prostriedky WWW, je jednoduchší a nenedovoľuje stavovosť na strane servra.

- Idempodencia

- v procedurálnom programovaní idempodencia znamená že ak aj viackrát zavoláme nejakú funkciu s nejakými parametrami, systém zakaždým vráti to isté a systém bude v tom istom stave ako po prvom volaní.
- Toto sa týka aj sieťových služieb, ak majú mať prívlastok idempodentné
- pozor. HTTP POST nie je idempodentný

CRUD operácie

- CRUD = Create, Read, Update, Delete
- Skratka vznikla pre základné operácie nad záznamami v perzistentnom úložisku
- skratka je používaná v informačných systémoch napr. pri operáciách pomocou SQL, HTTP a REST

Operácia	SQL	HTTP	RESTful WS
Create	INSERT	PUT / POST	POST
Read (Retrieve)	SELECT	GET	GET
Update (Modify)	UPDATE	PUT / POST / PATCH	PUT
Delete (Destroy)	DELETE	DELETE	DELETE

SOA

- SOA = service oriented architecture (architektúra orientovaná na služby)
- sada princípov a metodológií, ktoré odporúčajú skladať zložité aplikácie systémy zo skupiny na sebe nezávislých komponentov, ktoré poskytujú jednotlivé služby.
- SOA sa najčastejšie implementuje pomocou webových služieb (Web services)
 - SOAP
 - REST WS
- 8 hlavných princípov SOA (<http://www.soapprinciples.com>):
 - Štandardizovaný kontrakt služby – presne definovaný sľub okoliu, čo a ako sa poskytuje
 - Princíp abstrakcie – čo najviac skryť implementačné detaily
 - Znovu použiteľnosť – premyslený rozsah komponentov a služieb
 - Nezávislosť – izolácia služieb od ostatných služieb, minimalizácia zdieľaných zdrojov
 - ...

ESB systémy

- ESB = Enterprise Service Bus
- implementuje komunikačný systém medzi SW aplikáciami v SOA.
- Implementuje špeciálnu variantu modelu klient-server, kde sa každá aplikácia môže správať ako server alebo klient.
- Primárne povinnosti ESB sú:
 - Smerovanie správ medzi službami
 - Monitorovanie a riadenie výmeny správ
 - Riešenie sporov medzi komunikujúcimi komponentmi služieb
 - Riadenie nasadzovania a verzionovanie služieb
- Existuje veľa komerčných aj open source riešení ESB vyskúšajte napr.
 - Mule ESB (komerčné)
 - Apache Camel (open source)
- Viaceré z ESB riešení umožňujú tvoriť služby pomocou jazyka Web Services Business Process Execution (WS-BPEL), známeho aj ako BPEL
- Niekedy sa takéto systémy nazývajú aj ako „Integračná platforma“

