



# Java Content Repository

---

Peter Klimo

Rastislav Reháč



## Kto sme ?

Rastislav Rehák, architekt @ EEA s.r.o.

Peter Klimo, architekt @ EEA s.r.o.

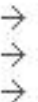
Realizované projekty s JCR

CUSTODEA (Europeana)

system na prepojenie kultúrnych inštitúcií s portálom Europeana  
prostredníctvom metadát o umeleckých dielach

Slovakiana

repozitár metadát a digitálnych reprodukcí z kultúrnych inštitúcií SR



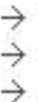


# Ra100 Rehák



**EEA communication solutions.** Hattalova 12B, 831 03 Bratislava, Slovak Republic  
Phone: +421 2 444 53690, <http://www.eea.sk>

Rastislav Rehák



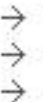


# Úvod do JCR

Java Content Repository je druh objektovej databázy  
je určená na ukladanie, vyhľadávanie a získavanie dát  
dáta sú uložené v hierarchickej štruktúre  
štruktúru ako aj spôsob prístupu k dátam normalizuje JCR špecifikácia:

JSR 170 – JCR 1.0 (RI Apache Jackrabbit 1.x, Alfresco, ModeShape (predtým JBoss DNA), eXo platform)

JSR 283 – JCR 2.0 (Apache Jackrabbit 2.0)





# Repository Model

Repository

Workspace

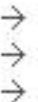
logická jednotka v rámci repozitára  
repozitár obsahuje jednu alebo viac workspaces  
jedna z workspaces je definovaná ako "default"

Items (položky)

tvoria v rámci repozitára orientovaný acyklický graf (nadmnožina stromu)  
parent-child relácia, jeden child môže mať viacero parentov

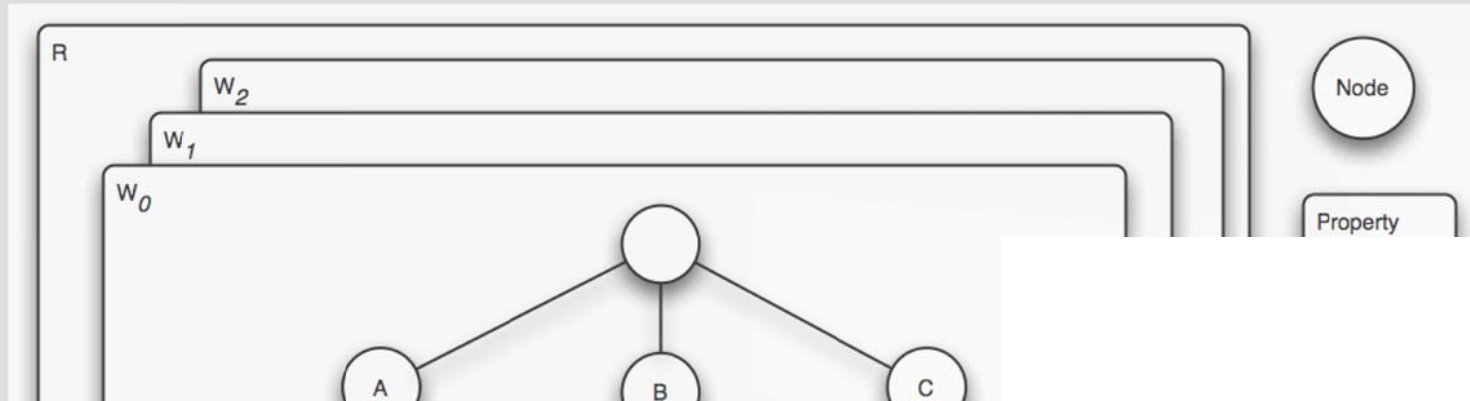
Uzol (Node) – obsahuje 0-n položiek

Property – obsahuje 0-n hodnôt, nemá však deti





# Repository Model



# Štruktúra položiek - príklad

Firma Názov, Adresa

Oddelenie Názov, Adresa

Pracovník Meno, Priezvisko, Číslo topánok

Dokumenty

Faktúra Číslo, Dátum, Zákazník, Cena

Položka Názov, Počet, Jednotková cena

Pokladňa



# Umiestnenie položky v repozitári

Umiestnenie každej položky (item) môže byť identifikované v repozitári cestou (path)

Cesta je tvorená:

vymenovaním hierarchicky nadradených položiek oddelených “/”  
uvedením identifikátora uzavretom v “[ ]”

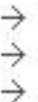
Príklady:

/

/ex:document

/ex:document/ex:paragraph[2]

[f81d4fae-7dec-11d0-a765-00a0c91e6bf6]





# Pomenovanie položiek

Každá položka v repositári je identifikovaná:  
jednoznačným identifikátorom  
menom

Meno je definované ako usporiadaná dvojica

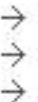
namespace:local\_name

namespace je buď prázdny reťazec alebo URI

lokálne meno je neprázdny reťazec obmedzený gramatikou definovanou  
JCR špecifikáciou

Priklady:

{<http://www.eea.sk/repository>}osoba – tzv. expanded form





# Namespaces

## Namespace Registry

perzistentné úložisko priradení prefix:uri  
platí v rozsahu celého repozitory

Minimálny obsah registra – rezervované namespaces:

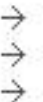
jcr = <http://www.jcp.org/jcr/1.0>

nt = <http://www.jcp.org/jcr/nt/1.0>

mix = <http://www.jcp.org/jcr/mix/1.0>

xml = <http://www.w3.org/XML/1998/namespace>

(prázdny string) = (prázdny string)



# Typy položiek / Properties

Na zabezpečenie štruktúrálnej konzistencie dát v repositári umožňuje JCR špecifikácia definovať typ položiek

## Properties

STRING (`java.lang.String`), URI (`java.lang.String` zodpovedajúce URI-reference RFC 39861)

BOOLEAN (Java primitive `boolean`), LONG (Java primitive `long`), DOUBLE (Java primitive `double`)

DECIMAL (`java.math.BigDecimal`), BINARY (`javax.jcr.Binary`), DATE (`java.util.Calendar`)

NAME (inštancie JCR mien), PATH (inštancie JCR ciest, bez zabezpečenia referenčnej integrity, len string v tvare JCR cesty),



# Typy položiek / Uzly I.

Typ uzla definuje požadované alebo povolené child položky (items)

Každý uzol má:

Práve jeden **primárny typ uzla** (primary node type)

Musí byť uvedený pri vytváraní uzla

0-n **mixin** typov

Je možné ho aplikovať už na vytvorený uzol neskôr

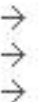
## Mixin

Je analógiou "marker" interface v Jave, alebo anotácie

Slúži na vyjadrenie istého špecifického aspektu uzla v repozitári

Príklady: `mix:referenceable`, `mix:versionable`, `mix:lockable`

Domain špecifické: `myapp:emailable`





# Typy položiek / Uzly II.

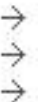
## Primárny typ uzla

Definuje štruktúru child položiek, ich typy, počet a ostatné atribúty

Primárne typy uzlov sú v JCR organizované hierarchicky, podporujú dedičnosť (mixiny nepodporujú dedičnosť)

JCR špecifikácia definuje štandardné typy uzlov a zároveň umožňuje tvorbu vlastných typov

Základným primárnym typom uzlov je typ `nt:base`





# Definovanie typov uzlov I.

Co vsetko obsahuje definicia typov uzlov ?

meno typu

supertyp (v zmysle dedenia)

abstraktný/konkrétny typ

mixin(y)

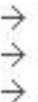
príznak "queryable"

príznak "orderable" pre detské uzly

primárny item

definícia detských uzlov

required





# Definovanie typov uzlov II.

...

Definovanie child properties

Typ property

Defaultna hodnota

Dostupné operátory pri dotazovaní

Možnosť vyhľadávania formou Full-Text-u

Zoraditeľnosť v dotazoch

Obmedzenia nadobúdaných hodnôt (constraints)

Multi-single value príznak





# Ako definíciu typu zapísať ?

Compact Node Type Definition (CND)

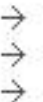
Priklad:

```
<eea = 'http://www.eea.sk/demo' >
```

```
[eea:Firma]
```

- nazov (String)
- adresa (String)
- + \* (eea:Oddelenie)

```
[eea: Oddelenie]
```





# Základné primárne typy uzlov

nt:base

nt:hierarchyNode

nt:file (obsahuje skutočný obsah súboru)

nt:linkedFile (obsahuje referenciu na súbor)

nt:folder

nt:resource

nt:address

nt:unstructured





# Základné typy mixinov

mix:title

mix:created

mix:lastModified

mix:language (od JCR 2.0)

mix:mimeType

mix:referenceable

mix:lockable

mix:shareable

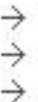
mix:lifecycle





## JCR API

- Aplikačné rozhranie k Java Content Repozitárom umožňuje:
  - Pripojenie k repozitáru, autentifikácia cez login-logout
  - Čítanie z repozitára – prístup k položkám
    - Priamo – zadaním cesty k položke
    - Traverzovaním
  - Získavanie objektov prostredníctvom dotazov (query)
  - Zápis položiek do repozitára
  - Spracovanie založené na eventoch generovaných repozitárom (tzv. observations)
  - Zisťovanie vlastností repozitáru (discovery) ako napr:
    - typy uzla (použitý typ uzla, dostupné typy uzlov)





# Queries

Dotazy podprované JCR špecifikáciou sú založené na Abstract Query Modeli (AQM)

AQM nie je konkrétnym jazykom, ale logickou definíciou sémantiky dotazov

AQM dotaz obsahuje:

- selektory

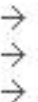
- joins (inner, left-outer, right-outer)

- konštanty (cesta, meno, hodnota atď)

- orderings

Každé JCR repozitory podporujúce dotazy musí podporovať nasledovné implementácie AQM:

- JCR-SQL2 – jazyk “SQL like”





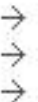
# Podporované dotazovacie jazyky

## JCR-SQL2

```
Query query = qm.createQuery("SELECT 'nt:unstructured'",  
    Query.JCR_SQL2);
```

## JCR-OQM

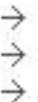
```
QueryObjectModelFactory qomf = qm.getQOMFactory();  
Selector selector = qomf.selector("nt:unstructured");  
// qomf.createQuery(selector, constraint, orderings, columns)
```





# Podporné služby JCR repozitára

- Import z XML formátu
- Export do XML formátu
- Verziónovanie položiek
  - Simple – priame verziónovanie v hlavnej línii
  - Full – vytváranie branches a merges
- Manažment prístupových práv
- Transakčnosť





# Úroveň JCR compliance

## Basic

Podporuje všetky povinné features definované JCR špecifikáciou:

Pripojenie k repozitory, autentifikácia a autorizácia používateľa

Čítanie prostredníctvom cesty, identifikátora alebo traverzovania

Dotazovanie (Queries)

Export do XML

Zisťovanie podporovaných a použitých typov uzlov prostredníctvom API

Zisťovanie oprávnení (permissions) a schopností (capabilities) repozitára

## Full

Všetky ostatné features definované JCR špecifikáciou. Iba takáto implementácia sa označuje ako "fully-compliant"





## Apache **Jackrabbit**

Úplná a referenčná implementácia JCR

Open source

Deployment:  
Samostatný server  
JCA resource

Prístup:  
Priamy  
RMI