# DSpace API v praxi

Vlastimil Krejčíř, krejcir@ics.muni.cz

Ústav výpočetní techniky, Masarykova univerzita, Brno

CZDSUG 2010, Ostrava

# Obsah

- Architektura DSpace, kompilace.
- Public API, dostupné třídy, metody.
  - Context.
  - Community, Collection, Item, Bundle, Bitstream.
  - AuthorizeManager, Group, MetadataSchema.
- Vlastní import.
- Knihovny (.jar).
- Konfigurační soubor.
- Metadata DC, řešení.

# Architektura DSpace – datový model

Základní „objekty“ pro práci se DSpace:
Community $\rightarrow^*$ (Subcommunity) $\rightarrow$ Collection $\rightarrow$ **Item**

**Item** – kontejner zapouzdřující:
– metadata DC (popř. jiná)
– Bundle $\rightarrow$ Bitstream

**Bundle** – pojmenovaný svazek souborů (**Bitstreams**)

# Architektura DSpace – vnitřní schéma

Logicky tři (provázané) vrstvy:

- aplikační vrstva (`org.dspace.app`)
  - JSP rozhraní (`webui`)
  - import, export
  - filtrace obsahu pro fulltext, náhledy
  - OAI server
  - ...
- bussines vrstva (`org.dspace`)
  - vše kromě `app` a `storage`
  - nejdůležitější třídy pro práci s objekty
- storage vrstva (`org.dspace.storage`)
  - JDBC wrapper
  - bitstore wrapper

# Architektura DSpace – vlastnosti

- „čistá“ Java ⇒ žádný framework (viz JDBC wrapper)
- knihovny třetích stran
  - Lucene
  - PDFbox (doporučuji poslední verzi!)
  - Handle
  - PostgreSQL
  - ...
- JavaBeans v JSPUI (vlastní tagy)
- výborné **Public API** přes JavaDocs
- Apache Ant – nyní náš přítel

# Generování dokumentace

**Public API:**

```
DSPACE_SRC$ ant public_api
```

Vygenerované API je v `build/public-api`.

Alternativně použitím JavaDocs pro Manakin:

```
MANAKIN_SRC$ javadoc -public -d public-api \
-sourcepath src/ org.dspace.app.xmlui org.dspace.app.xmlui.wing
```

# Úprava build.xml

Kompilace kódu a instalace dspace.jar – nový cíl v souboru build.xml:

```
<target name="install_jar" depends="compile">
  <mkdir dir="${dspace.dir}/lib" />
  <jar jarfile="${dspace.dir}/lib/dspace.jar"
    basedir="build/classes" />
</target>
```

# Spouštění kódu – dsrun

```
$DSPACE/bin/dsrun <trida> <parametry>
```

Nastaví cesty ke knihovnám, paměť apod.

Spouštěná třída musí obsahovat metodu
`public static void main(String[] argv)`

Viz příklady v této přenášce.

# org.dspace.core.Context

Provází nás na každém kroku, nositel informací o:

- uživateli, skupinách
- spojeních do databáze
- nějakých dalších fíčurách

V podstatě naprostá většina metod pracuje s kontextem jako s jedním ze svých parametrů. Vytvořit kontext je tedy to první, co je nutné udělat.

Užitečná metoda `setIgnoreAuthorization(true)`.

Příklad: `ContextExample.java`

# org.dspace.core.Context II.

Metody `commit()` a `complete()` třídy *Context*.

DSpace provádí veškeré operace s objekty v paměti a zápis do databáze je prováděn až se zavoláním `commit()`.

Metoda `complete()` provede `commit()` a korektně ukončí spojení s databází.

Tyto metody je nutné použít vždy, kdy dochází k zápisu do databáze (změna metadat, přístupových práv apod.).

Související metoda `abort()` provede rollback dosud necommitovaných změn.

Příklad: `ContextCommitAbortExample.java`

# org.dspace.content.*

V tomto balíku se nachází třídy pro práci s objekty v DSpace:

- `DSpaceObject`
- `Community`
- `Collection`
- `Item`
- `Bundle`
- `Bitstream`

Při **změně metadat** je nutné volat metodu `update()`!

Příklady:
`CommunityCollectionExample.java`
`CommunityDeleteExample.java`

# org.dspace.content.Item

K vytvoření *Item* je potřeba třída `org.dspace.content.WorkspaceItem`. Z ní následně získáme objekt třídy `Item` pro nastavení metadat a nahrání souborů. *WorkspaceItem* se pak použije pro finální vytvoření hotové *Item*.

Zpracujeme metadata a soubory.

Pomocí třídy `org.dspace.content.InstallItem` dokončíme vytvoření *Item*. `InstallItem` přidělí identifikátor *Handle* a archivuje *Item* v kolekci.

Příklad: `ItemImportExample.java`

# Bundle

Soubory (Bitstreams) se neukládají do Item přímo, ale musí se ukládat do tzv. **Bundles**, které se připojují k Item. Tyto **Bundles** se rozlišují svým jménem.

Je možné (a mnohdy vhodné) vytvářet vlastní pojmenované bundly. Tyto se samozřejmě nijak neprojeví v defaultním rozhraní JSP, ale je výhodné je využívat při zpracování přes Manakin.

DSpace samotný (resp. rozhraní JSP) si rezervuje 4 pojmenované bundly (je možné je v Manakinu ignorovat nebo použít na jiné účely – nedoporučuji to však, protože takové Items nebudou z rohraní JSP moc čitelné).

# Bundle – rezervovaná jména v DSpace

- ORIGINAL – v něm jsou hlavní dokumenty, které se ukazují uživatelům
- TEXT – do tohoto Bundlu si DSpace si (spuštěním *filter-media*) ukládá text, extrahovaný z dokumentů v *bundlu* ORIGINAL
- THUMBNAIL – zde si DSpace ukládá náhledy obrázků z *bundlu* ORIGINAL (opět spuštěním *filter-media*)
- LICENSE – uložené soubory s licencemi (např. CC)

# Item – související třídy

Spolu s Item souvisejí další užitečné třídy:

- DCValue
- FormatIdentifier
- ItemComparator
- ItemIterator
- BitstreamFormat
- ...

Příklad: ItemInfoExample.java

# Jiné užitečné třídy

Práce s identifikátory:

- `org.dspace.handle.HandleManager`

Třídy pro autorizaci (omezení přístupu apod.):

- `org.dspace.authorize.AuthorizeManager`
- `org.dspace.eperson.EPerson`
- `org.dspace.eperson.Group`

Práce s metadatovým formátem:

- `org.dspace.content.MetadataSchema`
- `org.dspace.content.MetadataField`

Příklad: `MetadataSchemaExample.java`

# Jiné užitečné třídy II

Interní indexy pro procházení (viz `bin/index-all`):

- `org.dspace.browse.*`

Nastavení *BrowseScope* `setFocus(<hledaný_výraz)`, který se předloží jako argument některé z metod třídy *Browse*:

- `getItemsBySubject()`
- `getAuthors()`
- `getItemsByTitle()`
- `...`

Získáme objekt třídy *BrowseInfo* a z něj výsledky (`getItemResults()`, `...`).

Příklad: `BrowseExample.java`

# Knihovny `.jar`

Je možné použít další knihovny třetích stran. Např. v DML-CZ `com.ibm.icu.text.*` pro korektní převod UTF-8 do ASCII apod.

Stačí nakopírovat příslušný `.jar` do `$DSPACE_SRC/lib/` a `$DSPACE/lib/`.

Doporučuji přejít na PDFbox verze 1.1.0.

# Konfigurační soubor dspace.cfg

Třída `org.dspace.core.ConfigurationManager`:

- `propertyNames()`
- `getProperty()`
- `getBooleanProperty()`
- `...`

Vlastní prefix – pro DML-CZ je to `dml.*`.

```
ConfigurationManager.getProperty("dml.genid")
```

# Nepříjemnosti v DSpace API

- konstruktory hlavních tříd jsou `private`
- chybějící metody přidělení vlastního identifikátoru pro `Community` a `Collection`
- některé věci by mohly být pohodlnější
- ...

# Řešení DML-CZ

- strukturovaná metadata v DC (Similar Articles)
- DmlItem, DmlCollection, DmlCommunity
- počty článků u autorů
- MSC kódy
- přidělování handlů (a problém mazání)
- rychlost Manakinu - cache
- ...

`http://projekt.dml.cz`

`http://dml.cz`

# Řešení v DML-CZ II

**Strukturovaná metadata v DC**

Do normálního záznamu Dublin Core ukládáme data ve formátu ala CSV, tedy kódovaný řetězec, oddělený tabulátorem.

# Řešení v DML-CZ III

**Počty článků, MSC**

Vlastní databáze (SQL) s uloženými počty článků pro jednotlivé autory, aktualizace periodicky (cron). Totéž pro MSC (které je ve stromové struktuře, ale v podstatě je to totéž) – MSC kódy do *subject.msc*.

Využití již hotových interních indexů DSpace a tříd `org.dspace.browse.*` .

# Řešení v DML-CZ IV

**Třídy Dml[Item‖Collection‖Community]**

Vlastní třídy pro práci s objekty DML-CZ. Jako atributy nesou samotné třídy *Item*, *Collection*, *Community*, dále infromace o typu (časopis, sborník, monografie) a rodiči.

Konstruktory vytvářející celý strom, vlastní komparátory, zpracování vlastních *Bundles* a další metody usnadňující život.

# Přidělování Handle identifikátoru

Identifikátor Handle je přidělen právě jednou a zůstává perzistentní v databázi i v případě, že je objekt (např. *Item*) smazán (pomocí dostupných metod).

V DML-CZ přidělujeme vlastní jednoznačné handly (negeneruje nám je DSpace), kvůli synchronizaci s Metadatovým Editorem*. Smažu-li Item, handle zůstává a při jejím znovuvytvoření dostávám Exception.

Pro nutné případy máme vlastní metody mazání, které odstraňují i handle (přímo i z tabulky *handle*).

* Viz `http://project.dml.cz/`

# Další témata volně k diskusi

Máte-li cokoli, co chcete probrat, sem s tím!

Materiály a slajdy na:

# http://dspace.cz/