

**sdgjes**

**Import danych z pliku**

# Agenda

## CZĘŚĆ TEORETYCZNA

- ✓ Jakie formaty można importować?
- ✓ Przygotowanie pliku CSV
- ✓ Struktura pól
- ✓ Mapowanie pól
- ✓ Import danych

## CZĘŚĆ PRAKTYCZNA

# Jakie formaty plików można importować?

- ✓ MARC
- ✓ Aleph
- ✓ plik CSV

# Przygotowanie pliku CSV – na co zwrócić uwagę

- ✓ kodowanie pliku w UTF-8
- ✓ jako separator kolumn powinien być stosowany średnik
- ✓ w pierwszej kolumnie powinien znajdować się identyfikator, po którym rozpoznajemy poszczególne rekordy i przed mapowaniem tego pola wpisujemy IDENTIFY: (dotyczy sytuacji, gdy aktualizujemy dane w Omedze)
- ✓ mapowanie pól zapisywane jest zawsze w drugim wierszu w pliku
- ✓ dane w kolumnach nie powinny zawierać żadnych niepotrzebnych znaków np. spacji

# Struktura pól

Strukturę poszczególnych pól można znaleźć pod adresem:

**adresbazy//doc/nazwapola/nazwapola.html**

gdzie:

**adresbazy** – to adres url dla danej instalacji Omegi – np. dla IMP jest to <https://ppm.imp.lodz.pl>

**nazwa pola** – tutaj podstawiamy nazwę pola złożonego, którego strukturę chcemy zobaczyć np. author, affiliation itp.

# Struktura pól

Wpisując w oknie adresu w przeglądarce:

<http://ppm.imp.lodz.pl//doc/author/author.html>

otrzymamy:

## Struktura typu *author*

Reprezentuje osobę

Nazwa pola	Typ	Ograniczenia	Znaczenie
id	ID	Pole identyfikatora	id
recordStatus	string	Pole wyboru: { partial, completed, verified, }	Status rekordu
owner	ID	Pole identyfikatora	Właściciel rekordu
affiliationowner	ID	Pole identyfikatora	Kolekcja
verificationDate	date	Data	
verifier	ID	Pole identyfikatora	
active	boolean	Wartosc Logiczna	Aktywny (dostępny w wyszukiwaniu publicznym)?
status	string	Pole wyboru: { PRACOWNIK, BYLY-PRACOWNIK, PRACOWNIK-UCP, BYLY-PRACOWNIK-UCP, DOKTORANT, PRACOWNIK-DOKTORANT, DYPLOMANT, EMERYT, GOSC, ZEWNETRZNY, NIEZYJACY, }	Status
officialAcademicDegree	Obiekt zlozony: term	Pojedynczy	Tytuł naukowy / Stopień naukowy / tytuł zawodowy
academicDegree	string	Zwykle pole tekstowe	Inne stopnie/tytuł
academicCategory	Obiekt zlozony: term	Powtarzany	Kategorie naukowe
specialization	Obiekt zlozony: term	Powtarzany	Specjalizacja zawodowa
belonging2N	boolean	Wartosc Logiczna	Uwzględniany przy określaniu liczby 'N'
name	string	Zwykle pole tekstowe Wymagane Minimum: 2Znaki Maximum: 1024Znaki	Imię
surname	string	Zwykle pole tekstowe Wymagane Minimum: 2Znaki Maximum: 1024Znaki	Nazwisko
presentedFullName	string	Zwykle pole tekstowe	
otherName	string	Powtarzany / Zwykle pole tekstowe	Inne formy nazwiska
email	string	Zwykle pole tekstowe Zgodne z wyrażeniem: <code>^[A-Za-z0-9!#\$%*./=?^_`{}~\+]*@[A-Za-z0-9!#\$%*./=?^_`{}~\+]*(\.[A-Za-z0-9!#\$%*./=?^_`{}~\+])*(\.[A-Za-z]{2,})\$</code>	Email

# Struktura pól

- Obiekt typu String jest ciągiem znaków
- Obiekt złożony posiada swoją własną strukturę, w mapowaniu korzystając z pól obiektu złożonego musimy się odwołać do obiektu nadrzędnego
- Obiekt złożony Term jest daną słownikową
- Obiekt typu boolean jest zmienną logiczną przyjmującą wartości true or false

# Mapowanie pól

W mapowaniu używa się następujących znaków:

<< - przypisz do danego pola

<> - pobierz istniejący obiekt z bazy danych

() – stwórz nowy obiekt

[] – to pole jest kolekcją

| - oznacza „gdzie”, po niej następuje zapytanie do bazy danych

++ - służy do oddzielania wartości w komórce z danymi, która zawiera dane do więcej niż

jednego pola



# Mapowanie pól - przykłady

- `status=PRACOWNIK` – w polu „Status” przypisz każdemu wartość Pracownik
- `authorprofile/prefixPL=mgr` – w profilu autora w polu „Tekst przed nazwiskiem w języku polskim” ustaw każdemu wartość mgr
- `authorprofile/positionPL={0}` - w profilu autora w polu „Stanowisko w języku polskim” wpisz wartość z pliku CSV
- `affiliation<<Affiliation<> | acronymPL={0}` – znajdź obiekt typu afiliacja o takim samym akronimie w języku polski jak w komórce w pliku, a następnie wpisz go w polu „Afiliacja (główna)” w rekordzie danej osoby

# Mapowanie pól - przykłady

- `authorprofile/extid[]<<$(staffId)` - „Identyfikator w systemach zewnętrznych” (w Profilu autora), dla którego tworzymy nowy obiekt o nazwie „Identyfikator pracownika” i wpisujemy mu wartość z pliku CSV
- `staffGroup<<Term<> | termttype/code=staffGroup&&namePL={0}` - do pola „Grupa zatrudnienia” dodaj obiekt typu Term gdzie pole „Kod systemowy typu danych słownikowych” jest równe Grupie zatrudnienia i „Nazwa w języku polskim” jest brana z pliku CSV

# Import danych – URL

Przy imporcie pliku do Omegi należy wybrać jeden z URL, w zależności od operacji jaka będzie wykonywana:

- `<protocol>://<domain>/entity-service/rest/entityaccesspoint/csv/save` – jeżeli będziemy dodawać jedynie nowe rekordy do bazy
- `<protocol>://<domain>/entity-service/rest/entityaccesspoint/csv/update` – używany w przypadku kiedy chcemy zaktualizować istniejące rekordy w bazie; jeżeli w czasie importu pojawi się nowy rekord to zostanie on dodany do bazy

w miejsce `<protocol>` wpisujem `https`

a w miejsce `<domein>` domenę w jakiej znajduje się nasza baza czyli dla IMP będzie to `ppm.imp.lodz.pl`

# Import danych

Import pliku do bazy może odbyć się przy użyciu:

- wiersza poleceń
- programów graficznych taki jak POSTMAN lub SoapUI

sdgjes

**Dziękujemy za uwagę**