

02. Struktura katalogu opisującego publikację

Informacje podstawowe

Katalog opisujący publikację składa się z:

- Pliku `publication.properties` opisującego publikację,
- Pliku z opisem bibliograficznym publikacji (inaczej pliku metadanych),
- Plików, które wchodzi w skład publikacji.

Plik `publication.properties`

Plik `publication.properties` jest plikiem właściwości publikacji. Składa się on z par klucz-wartość. Klucz, czyli nazwa właściwości, oddzielony jest od wartości znakiem równości (=). Każda taka para znajduje się w oddzielnej linii. Plik powinien być zapisany w kodowaniu UTF-8.

Dozwolone właściwości kluczy oraz ich wartości przedstawione są poniżej.

- `publication.actorsRights.<login użytkownika lub nazwa grupy do której przydzielamy prawa>`.
Prawa oddzielone przecinkiem. Uwaga: jeśli login zawiera spację, muszą one zostać poprzedzone znakiem '\\'.
Lista możliwych praw (szczegółowe informacje o prawach można znaleźć [tutaj](#)):
 - `pv` - dostęp tylko do opublikowanych wydań publikacji
 - `pr` - dostęp do wszystkich wydań publikacji
 - `pe` - zarządzanie publikacją
- `publication.collections`
Identyfikatory kolekcji, oddzielone przecinkiem, do których ma zostać przypisana publikacja.
- `publication.destination.directoryId`
Identyfikator katalogu, w którym ma się znaleźć publikacja.
- `publication.destination.parentPublicationId`
Identyfikator publikacji grupowej, w której ma się znaleźć publikacja.
- `publication.mainFile`
Nazwa pliku głównego publikacji (wraz z rozszerzeniem). Plik główny musi znajdować się w katalogu opisującym publikację. W przypadku publikacji wielofragmentowej, zamiast tego klucza powinny być zdefiniowane wartości `publication.mainFormat` oraz przynajmniej jednej wartości `publication.mainFile.<format>`.
- `publication.mainFile.<format>`
Nazwa/ścieżka pliku głównego w wybranym formacie. Wszystkie pliki danego formatu powinny znajdować się w podkatalogu o nazwie takiej jak format, np. PDF. Wartość klucza również musi zawierać nazwę katalogu.
- `publication.mainFormat`
Nazwa formatu, który ma być uznawany za główny (wyświetlany domyślnie jako pierwszy).
- `publication.metadataFile`
Nazwa pliku metadanych (wraz z rozszerzeniem). Plik metadanych musi znajdować się w katalogu opisującym publikację oraz być zgodny z jednym z formatów obsługiwanych przez aktualnie zainstalowane rozszerzenia do importu metadanych (zwykle jest to RDF, MARC lub Bibtex - zobacz rozdział [Import i eksport metadanych](#)). Jeśli wartość dla tego klucza nie jest zdefiniowana, domyślnie wyszukiwane są pliki o nazwie takiej samej jak nazwa pliku głównego, z rozszerzeniami domyślnymi dla każdego obsługiwanego formatu (np. `.rdf`, `.mrc`, `.bib`).
- `publication.name`
Łańcuch znaków określający nazwę publikacji. Jeśli wartość dla tego klucza nie zostanie zdefiniowana, aplikacja spróbuje wygenerować nazwę na podstawie wartości atrybutów (w taki sam sposób, jak w [drugim kroku kreatora nowej publikacji](#)).
Uwaga: Wartość ta określa również nazwę wydania. Przy dodawaniu treści do publikacji planowanej należy również uzupełnić tę wartość.
- `publication.notes`
Łańcuch znaków określający notatki/uwagi administracyjne.
- `publication.published`
Łańcuch znaków wskazujący na to, czy pierwsze wydanie tej publikacji ma zostać opublikowane. Jeśli wartość jest `true` wydanie to będzie opublikowane. W każdym innym przypadku wydanie nie będzie opublikowane. Domyślną wartością jest `false`.
- `publication.secured`
Łańcuch znaków wskazujący na to, czy publikacja ma być zabezpieczona. Jeśli wartość jest `true` publikacja będzie zabezpieczona. W każdym innym przypadku publikacja nie będzie zabezpieczona. Domyślną wartością jest `false`.
- `image.content`
Nazwa pliku, który zawiera miniaturę wydania.
- `publication.publishingDate`
Planowana data udostępnienia publikacji, np. 2020-02-29
- `publication.id`
Identyfikator publikacji (obiektu) do aktualizacji (zob. `update.mode` poniżej).
- `edition.id`
Identyfikator wydania (wersji obiektu) do aktualizacji (zob. `update.mode` poniżej).

- `edition.externalId`
Zewnętrzny identyfikator powiązany z wydaniem (wersją obiektu). Wartość ta nie jest używana przez system dLibra, ale jest przechowywana w bazie danych, więc może być wykorzystana przez zewnętrzne narzędzia operujące na bazie. Identyfikator ten może też służyć do wskazania wydania (wersji obiektu) do aktualizacji (zob. `update.mode` poniżej).
- `publication.registerDOI`
Łańcuch znaków wskazujący na to, czy przy tworzeniu publikacji ma nastąpić automatyczna rejestracja numeru DOI. Jeśli wartość jest `true`, nastąpi próba rejestracji numeru DOI. W każdym innym przypadku nie zostanie wywołana żadna dodatkowa akcja. Domyślną wartością jest `false`.
- `publication.forceDOI`
Istniejący numer DOI powiązany z publikacją.
- `update.mode`
Pozwala zdefiniować tryb aktualizacji istniejącego obiektu. Wartość powinna zawierać wybrane opcje rozdzielane przecinkiem, np. `replaceContent, addMetadata`. Jednocześnie z tym kluczem musi być wskazany identyfikator istniejącego obiektu docelowego, przy pomocy jednego z kluczy: `publication.id`, `edition.id`, lub `edition.externalId`. Możliwe opcje aktualizacji to:
 - `addMetadata` - dodaje do opisu obiektu metadane załadowane z pliku wskazanego przez `publication.metadataFile`, bez usuwania wcześniej przypisanych wartości. Nie może być użyte jednocześnie z `replaceMetadata`.
 - `replaceMetadata` - zastępuje opis obiektu metadanymi załadowanymi z pliku wskazanego przez `publication.metadataFile`. Nie może być użyte jednocześnie z `addMetadata`.
 - `addContent` - dodaje do obiektu pliki treści z aktualnego katalogu, bez usuwania wcześniej istniejących plików (starsze pliki zostaną nadpisane, jeśli mają takie same nazwy jak nowe pliki). Jako plik główny publikacji ustawia zawsze plik wskazany przez `publication.mainFile`. Opcja nie może być użyta jednocześnie z `replaceContent`.
 - `replaceContent` - zastępuje wszystkie pliki z treścią w obiekcie plikami z aktualnego katalogu. Nie może być użyte jednocześnie z `addContent`.
 - `addRights` - dodaje do zbioru praw przypisanych do obiektu prawa zdefiniowane przez klucze `publication.actorsRights.*`. Nie może być użyte jednocześnie z `replaceRights`.
 - `replaceRights` - przypisuje do obiektu prawa zdefiniowane przez klucze `publication.actorsRights.*`, usuwając prawa przypisane wcześniej. Nie może być użyte jednocześnie z `addRights`.
 - `addCollections` - dodaje obiekt do kolekcji wskazanych przez `publication.collections`, bez usuwania z kolekcji przypisanych wcześniej. Nie może być użyte jednocześnie z `replaceCollections`.
 - `replaceCollections` - dodaje obiekt do kolekcji wskazanych przez `publication.collections` oraz usuwa wcześniej istniejące przypisania do kolekcji. Nie może być użyte jednocześnie z `addCollections`.
 - `image` - zastępuje miniaturę obiektu obrazkiem wskazanym przez `image.content`.
 - `adminData` - zastępuje dane administracyjne obiektu (wartości wskazane w kluczach `publication.name`, `publication.notes`, `publication.secured`, `publication.published`, `publication.publishingDate`, `edition.externalId`).
 - `full` - specjalna opcja służąca do pełnego zastąpienia treści i właściwości obiektu, jednoznaczna z użyciem `replaceMetadata`, `replaceContent`, `replaceRights`, `replaceCollections`, `image`, `adminData`

Przykładowy plik `publication.properties`:

publication.properties

```
publication.actorsRights.U\u017Cytownicy\ publiczni=pv
publication.collections=2,7,12
publication.destination.directoryId=1
publication.mainFile=index.htm
publication.metadataFile=metadata.rdf
publication.name=Przykładowa publikacja
publication.notes=Stworzone przez Jana Kowalskiego
publication.published=true
```

Powyższy przykład pliku specyfikuje publikację, do której aktor `publicGroup` będzie miał prawo przeglądania wszystkich publikacji (nadano prawo `pr`, co implikuje niejawnie prawo `pv`). Publikacja zostanie przypisana do kolekcji oznaczonych identyfikatorem 2, 7 oraz 12. Publikacja znajdzie się w katalogu o identyfikatorze 1. Plikiem głównym publikacji będzie `index.htm`, a opis metadanych znajduje się w pliku `metadata.rdf`. Nazwa publikacji to *Przykładowa publikacja*, a notatki administracyjne to *Stworzone przez Jana Kowalskiego*. Pierwsze wydanie nowo stworzonej publikacji będzie opublikowane, a sama publikacja nie będzie zabezpieczona.

Publikacje grupowe

Aby tworzona publikacja była publikacją grupową, plik główny publikacji (klucz `publication.mainFile`) **nie może** być zdefiniowany. Jeśli katalog opisujący publikację grupową będzie zawierał podkatalogi, zostaną one również przeanalizowane, a opisane w nich publikacje zostaną dodane jako publikacje podrzędne publikacji grupowej. Publikacje podrzędne również mogą być publikacjami grupowymi (jeśli nie będą miały zdefiniowanego pliku głównego), zatem można stworzyć hierarchię o dowolnej liczbie poziomów.

Uwaga: jeśli dodawana publikacja nie jest publikacją grupową, określenie pliku głównego publikacji jest wymagane.

Publikacje planowane

Tworzona publikacja może być publikacją planowaną. Aby tak było, w katalogu z publikacją mogą być **tylko** 2 pliki: `publication.properties` i `description.rdf`. Dodatkowo w pliku `publication.properties` można umieścić klucz `publication.publishingDate`, którego wartością powinna być planowana data opublikowania. Dozwolony format daty to `yyyy-MM-dd`, gdzie `yyyy` oznacza rok, `MM` miesiąc a `dd` dzień. W przypadku późniejszego uzupełniania publikacji planowanej o treść, należy pamiętać o uzupełnieniu wartości `publication.name`, ponieważ jej wartość posłuży jako nazwa wydania.

Definiowanie umiejscowienia publikacji

Do stworzenia publikacji niezbędne jest określenie, gdzie w strukturze katalogów i publikacji grupowych systemu dLibra ma być ona dodana. Można posłużyć się w tym celu właściwością `publication.destination.directoryId` (katalog docelowy) albo `publication.destination.parentPublicationId` (nadrzędna publikacja grupowa). Druga z tych właściwości ma wyższy priorytet, zatem jeśli zostaną zdefiniowane obie właściwości, to katalog docelowy zostanie zignorowany (publikacja zostanie dodana do tego samego katalogu, w którym znajduje się nadrzędna publikacja grupowa).

W przypadku dodawania publikacji grupowej, dodawane wraz z nią publikacje podrzędne nie muszą mieć bezpośrednio zdefiniowanego położenia docelowego. Ponieważ nadrzędna publikacja grupowa jest od razu znana, właściwości `publication.destination.directoryId` i `publication.destination.parentPublicationId` zostaną zignorowane.

Definiowanie nazwy publikacji

Dodawana publikacja musi mieć określoną nazwę. Specyfikacja nazwy publikacji możliwa jest na dwa sposoby:

- bezpośrednio: specyfikacja właściwości `publication.name`,
- pośrednio: jeżeli powyższa właściwość nie jest wyspecyfikowana, zostanie sprawdzony plik metadanych w celu skomponowania nazwy publikacji, w taki sam sposób jak w [drugim kroku kreatora publikacji](#). W takim przypadku należy więc zadbać o to, aby metadane wykorzystywane do skomponowania nazwy były zdefiniowane.

Dziedziczenie właściwości w plikach `publication.properties`

Mechanizm dziedziczenia został wprowadzony w celu ułatwienia definicji właściwości w sytuacji, gdy wiele dodawanych publikacji ma takie same wartości. Właściwości takie nie muszą być definiowane bezpośrednio w każdym katalogu opisującym publikację, ale można je umieścić w pliku `publication.properties` w katalogu nadrzędnym (tym, który jest wskazywany w [kreatorze masowego ładowania publikacji](#)). Zostaną one odziedziczone do wszystkich dodawanych publikacjach. W poszczególnych publikacjach można zdefiniować wybrane właściwości jeszcze raz, co spowoduje nadpisanie odziedziczonych wartości.

Dziedziczenie właściwości działa również w ramach hierarchii zagnieżdżonych publikacji grupowych. Zatem jeśli tworzona jest publikacja grupowa, właściwości we wszystkich tworzonych wraz z nią publikacjach podrzędnych są dziedziczone z tej publikacji grupowej, chyba że zostaną nadpisane. Są jednak wyjątki od tej reguły:

- właściwości definiujące miejsce docelowe (`publication.destination.directoryId` i `publication.destination.parentPublicationId`) — publikacja nadrzędna i katalog wynikają ze struktury katalogów,
- identyfikatory kolekcji (`publication.collections`) — opisywany mechanizm je pomija, ponieważ [mechanizm grupowania publikacji](#) zapewnia dziedziczenie tych właściwości,
- prawa dostępu do publikacji (`publication.actorRights`) — jak wyżej.

Dziedziczenie metadanych

Mechanizm ten działa podobnie jak w przypadku [dziedziczenia plików `publication.properties`](#) z tym wyjątkiem, że plik metadanych w katalogu nadrzędnym nazywać się powinien `publication.rdf`. Zatem, jeśli w katalogu zawierającym katalogi z opisami publikacji zostanie znaleziony plik `publication.rdf`, zawarte w nim metadane zostaną przypisane wszystkim tworzonym publikacjom. Dziedziczenie to działa tylko na pierwszym poziomie katalogów — nie dotyczy publikacji zgrupowanych w nowo tworzonych publikacjach. Dziedziczenie metadanych z publikacji grupowych jest obsługiwane przez [mechanizm grupowania publikacji](#).

Generowanie przykładowego pliku właściwości

Używając Zapisz plik właściwości z menu Narzędzia możliwe jest zapisanie informacji opisujących publikację wybraną na liście elementów do pliku. Plik taki może służyć jako wzór pliku jaki powinien znaleźć się w katalogu opisującym publikację dla [masowego ładowania publikacji](#).