





Uwaga

Przed rozpoczęciem korzystania z tych informacji oraz z produktów, których one dotyczą, należy zapoznać się z sekcją "Uwagi" na stronie 33.

Uwaga dotycząca wydania

To wydanie odnosi się do wersji 8, wydania 4, modyfikacji 2 produktu IBM OmniFind Yahoo! Edition (numer produktu 5724-R21) oraz do wszystkich jego nowszych wydań i modyfikacji, chyba że w nowszych wydaniach pojawią się informacje, że jest inaczej.

© Copyright International Business Machines Corporation 2006, 2007. Wszelkie prawa zastrzeżone.

Spis treści

Serwis **ibm.com** i zasoby pokrewne v

Jak przesyłać swoje uwagi. v

Kontakt z firmą IBM v

Programowanie aplikacji 1

Funkcja API wyszukiwania 3

Żądania wyszukiwania 3

 Żądania wyszukiwania HTTP GET 3

 Parametry żądania wyszukiwania. 4

Wyniki wyszukiwania 7

 Dostarczane dane w formacie Atom 7

 Fragmenty kodu HTML 10

 Dokument opisu OpenSearch 11

Funkcje API do dodawania i usuwania dokumentów 15

Format żądań funkcji API do dodawania i usuwania dokumentów 15

Parametry żądań funkcji API do dodawania i usuwania dokumentów 17

Administracyjne funkcje API 19

Funkcja API do zarządzania przeszukiwaczem 19

Funkcje API dotyczące pól metadanych 21

Funkcja API generująca listę kolekcji 23

Odpowiedzi zawierające błąd 25

Kody odpowiedzi HTTP 25

Generowanie haseł funkcji API 27

Przykłady w języku Java, XSL i PHP . . . 29

Przykłady w języku Java wpisywane w wierszu komend 29

Przykładowy arkusz stylów XSL 30

Przykładowa aplikacja wyszukująca PHP 30

Uwagi 33

Uwagi 33

 Znaki towarowe 35

Indeks 37

Serwis [ibm.com](http://www.ibm.com) i zasoby pokrewne

Pomoc techniczna i dokumentacja produktu są dostępne za pośrednictwem serwisu [ibm.com](http://www.ibm.com).

Wsparcie i asysta

Wsparcie dla produktu jest dostępne za pośrednictwem sieci WWW. Aby je uzyskać, należy kliknąć odsyłacz Support w serwisie WWW produktu dostępnym pod następującym adresem:

OmniFind Yahoo! Edition

<http://www.ibm.com/software/data/enterprise-search/omnifind-yahoo/support.html>

Publikacje w formacie PDF

Zawartość plików PDF można przeglądać w trybie z połączeniem za pomocą aplikacji Adobe Acrobat Reader dla danego systemu operacyjnego. Można ją pobrać z serwisu WWW firmy Adobe dostępnego pod adresem <http://www.adobe.com>.

Należy zapoznać się z następującymi serwisami WWW z publikacjami w formacie PDF:

Produkt	Adres serwisu WWW
IBM OmniFind Discovery Edition	http://www-1.ibm.com/support/docview.wss?rs=3035&uid=swg27008552
IBM OmniFind Enterprise Edition	http://www-1.ibm.com/support/docview.wss?rs=63&uid=swg27007911
IBM OmniFind Yahoo! Edition	http://www.ibm.com/support/docview.wss?rs=3193&uid=swg27010191

Jak przesyłać swoje uwagi

Opinie użytkowników są dla nas ważne i pomagają nam dostarczać najdokładniejsze informacje o najwyższej jakości.

Można je do nas przesyłać za pomocą elektronicznego formularza na uwagi czytelników dostępnego pod adresem https://www14.software.ibm.com/webapp/iwm/web/signup.do?lang=en_US&source=swg-rcf.

Kontakt z firmą IBM

Aby skontaktować się z firmą IBM w Stanach Zjednoczonych lub Kanadzie, zadzwoń pod numer 1-800-IBM-SERV (1-800-426-7378).

Aby uzyskać informacje o dostępnych opcjach serwisowych, należy zadzwonić pod jeden z poniższych numerów:

- W Stanach Zjednoczonych: 1-888-426-4343
- W Kanadzie: 1-800-465-9600

Więcej informacji na temat sposobów kontaktowania się z firmą IBM można znaleźć w serwisie WWW Contact IBM pod adresem <http://www.ibm.com/contact/us/>.

Programowanie aplikacji

Do zintegrowania niestandardowych aplikacji z produktem IBM OmniFind Yahoo! Edition można użyć aplikacyjnych interfejsów programistycznych (API).

Interfejsy API udostępniają następujące funkcje:

- Wysyłanie zapytań i odbieranie wyników wyszukiwania. Na przykład wyniki mogą być osadzone bezpośrednio na stronie WWW lub można je sformatować zgodnie z wyglądem i zachowaniem własnej strony WWW.
- Dodawanie dokumentów do kolekcji. Na przykład można dodać treść ze źródła danych, którego nie można przeszukać przy użyciu przeszukiwaczy dołączonych do produktu, takiego jak system zarządzania treścią przedsiębiorstwa.
- Usuwanie dokumentów z kolekcji. Na przykład można usunąć dokumenty, które nie mają już być zwracane w wynikach wyszukiwania.
- Uruchamianie i zatrzymywanie przeszukiwacza. Funkcja API do zarządzania przeszukiwaczem może być użyta przez programy szeregujące w celu uruchomienia lub zatrzymania przeszukiwania o określonej godzinie.
- Przypisywanie wartości do pól metadanych podczas dodawania dokumentów do kolekcji. Jeśli dla kolekcji zostały skonfigurowane pola metadanych, można określać ich wartości.
- Wyświetlanie listy nazw wszystkich pól metadanych w kolekcji.
- Wyświetlanie listy nazw wszystkich kolekcji w systemie wyszukiwania.

Żądania funkcji API są oparte na standardzie HTTP, dzięki czemu są one niezależne od języka programowania.

Funkcja API wyszukiwania

Funkcja API search obsługuje żądania wyszukiwania w postaci komend **HTTP GET** i zwraca wyniki wyszukiwania w postaci dostarczanych danych kanałów Atom lub fragmentów kodu HTML.

Żądania wyszukiwania

Żądanie **HTTP GET** zwraca dokumenty pasujące do kryteriów wyszukiwania.

Wyniki wyszukiwania

Wyniki wyszukiwania zwracane w formacie syndykowania Atom 1.0 mogą zostać dostosowane do potrzeb użytkownika poprzez określenie w żądaniu wyszukiwania arkusza stylów. Po zwróceniu wyników wyszukiwania w postaci fragmentu kodu HTML można osadzić je w istniejącej stronie WWW.

Żądania wyszukiwania

Aplikacja wyszukująca użytkownika może udostępniać pole wyszukiwania przesyłające do serwera komendę **HTTP GET**.

Odsyłacze pokrewne

“Odpowiedzi zawierające błąd” na stronie 25

Żądania wyszukiwania HTTP GET

Żądanie wyszukiwania jest standardową komendą **GET** protokołu HTTP.

Adres URL żądania wyszukiwania można utworzyć, przygotowując kombinację następujących właściwości:

- Nazwa hosta
- Port
- Ścieżka
- Parametry żądania wyszukiwania, kolekcja par nazwa-wartość () rozdzielonych znakami ampersand (&)

Nazwa hosta to nazwa hosta serwera mechanizmu wyszukiwania. Port to numer portu dla aplikacji wyszukującej - wartość, która jest pierwotnie określana podczas instalowania serwera mechanizmu wyszukiwania. Ścieżka wysyłania żądania wyszukiwania to zawsze /api/search.

Przykłady żądań wyszukiwania HTTP GET

W poniższym przykładzie pokazano format adresu URL, który powoduje przeszukanie kolekcji domyślnej (Default) i zwrócenie pierwszych pięciu wyników zgodnych z zapytaniem *Szczecin*. Wyniki są zwracane w domyślnym formacie wyjściowym Atom.

```
http://nazwa_hosta:port/api/search?query=Szczecin&collection=Default&results=5
```

W poniższym przykładzie pokazano format adresu URL, który powoduje przeszukanie kolekcji domyślnej (Default) i zwrócenie pierwszych 20 wyników zgodnych z zapytaniem *Szczecin*. Wszystkie wyniki są zwracane w języku hiszpańskim (es) lub niemieckim (de):

```
http://nazwa_hosta:port/api/search?query=Szczecin&collection=Default  
&results=20&resultLang=pl|de
```

W poniższym przykładzie pokazano format adresu URL, który powoduje przeszukanie kolekcji domyślnej (Default) i zwrócenie pierwszych 10 wyników zgodnych z zapytaniem *fiesta*. Termin użyty w zapytaniu jest zapisany w języku hiszpańskim (es).

```
http://nazwa_hosta:port/api/search?query=fiesta&collection=Default
&queryLang=es
```

W poniższym przykładzie pokazano format adresu URL, który powoduje przeszukanie kolekcji Pracownicy i zwrócenie wyników o numerach 11-20, zgodnych z zapytaniem *kierownik*. Wyniki zwracane w formacie Atom są formatowane przy użyciu określonego arkusza stylów XSLT znajdującego się pod adresem `http://moj_serwer.com/stylesheet/atom.xsl`.

```
http://nazwa_hosta:port/api/search?query=kierownik&collection=Pracownicy
&start=10&results=10&stylesheet=http://moj_serwer.com/stylesheet/atom.xsl
```

W poniższym przykładzie pokazano format adresu URL, który powoduje przeszukanie kolekcji Pracownicy i zwrócenie wyników o numerach 11-20, zgodnych z zapytaniem *kierownik*. Wyniki są zwracane w formacie wyjściowym w postaci fragmentu kodu HTML:

```
http://nazwa_hosta:port/api/search?query=kierownik&collection=Pracownicy
&start=10&results=10&output=htmlsnippet
```

Parametry żądania wyszukiwania

W żądaniach wyszukiwania można stosować wiele opcji.

Kolejność parametrów w żądaniu nie ma znaczenia. W nazwach parametrów rozróżniana jest wielkość liter i należy je podawać w formacie opisanym w dokumentacji. Wszystkie nieznanne lub nieobsługiwane parametry wysłane jako część żądania zostaną zignorowane.

W poniższej tabeli pokazano obsługiwane parametry dla żądań wyszukiwania:

Tabela 1. Parametry żądania wyszukiwania

Nazwa parametru	Opis	Wartość domyślna	Komentarze
collection	Nazwa kolekcji do przeszukania.		Wymagany. Wartość ta musi być zakodowana w formacie UTF-8 i zawierać znaki zmiany znaczenia odpowiednie dla adresów URL. W celu wyświetlenia listy wszystkich kolekcji, które można przeszukiwać, należy skorzystać z konsoli administracyjnej lub z funkcji API collections.
fields	Obowiązkowe wartości pól metadanych, które mają zostać zwrócone w każdym wyniku, niezależnie od terminów użytych w zapytaniu.		Obowiązkowe pola, które mają zostać zwrócone, należy oddzielić od siebie dwukropkiem (:). Na przykład: <code>fields=author:keywords</code> . W wartości powinno się używać znaków zmiany znaczenia odpowiednich dla adresów URL.
filter	Filtruje wyniki wyszukiwania w celu wykrycia duplikatów dokumentów.	true	Obsługiwane wartości to true i false. Po wybraniu wartości true identyczne dokumenty zostaną zwinięte do postaci pojedynczego wyniku na liście wyników wyszukiwania. Dokumenty, które mają taki sam tytuł i podsumowanie również mają na liście wyników pojedynczą reprezentację.

Tabela 1. Parametry żądania wyszukiwania (kontynuacja)

Nazwa parametru	Opis	Wartość domyślna	Komentarze
locale	Ustawienia narodowe klienta.	Ustawienia narodowe serwera	Zwraca komunikaty w języku zgodnym z ustawieniami narodowymi klienta. Obsługiwane wartości: de_DE - niemiecki en_US - angielski es_ES - hiszpański fr_FR - francuski hu_HU - węgierski it_IT - włoski ja_JP - japoński ko_KR - koreański nl_NL - holenderski pl_PL - polski pt_PT - portugalski pt_BR - portugalski (Brazylia) sv_SE - szwedzki zh_CN - chiński (uproszczony) zh_TW - chiński (tradycyjny)
output	Format treści komunikatu w odpowiedzi serwera.	atomxml	Obsługiwane wartości to atomxml i htmlsnippet.
oyeFieldFormat	Używane dla potrzeb nieaktualnych formatów odpowiedzi wyszukiwania dla pól.	false	Przed pojawieniem się wersji 8.4.2 produktu IBM OmniFind Yahoo! Edition pola metadanych w odpowiedziach wyszukiwania w formacie Atom były reprezentowane w nieaktualnym formacie /feed/entry/omnifind:field . Nadając opisywanemu parametrowi wartość true, można wymusić dalsze stosowanie tego formatu odpowiedzi. Informacje na temat nowego formatu można znaleźć w sekcji “Dostarczane dane w formacie Atom” na stronie 7.
query	Łańcuch zapytania.		Wymagany. W wartości należy używać znaków zmiany znaczenia odpowiednich dla adresów URL.

Tabela 1. Parametry żądania wyszukiwania (kontynuacja)

Nazwa parametru	Opis	Wartość domyślna	Komentarze
queryLang	Język łańcucha zapytania.	Ustawienia narodowe serwera	Obsługiwane wartości: ar - arabski cs - czeski da - duński de - niemiecki el - grecki en - angielski es - hiszpański fi - fiński fr - francuski he - hebrajski it - włoski ja - japoński ko - koreański nl - holenderski no - norweski pl - polski pt - portugalski ru - rosyjski sv - szwedzki zh_CN - chiński (uproszczony) zh_TW - chiński (tradycyjny)
queryTimeout	Maksymalny czas wartościowania żądania zapytania, wyrażony w sekundach.	0 (nieograniczony)	Określając wartość tego parametru, można nałożyć limit czasu wartościowania żądań zapytania.
resultLang	Język, w którym mają być filtrowane i zwracane wyniki wyszukiwania.		Użyj znaku pionowej kreski (), aby oddzielić łańcuchy wskazujące język. Na przykład: en de fr. W wartości należy używać znaków zmiany znaczenia odpowiednich dla adresów URL. Lista dozwolonych wartości znajduje się w sekcji komentarzy do parametru queryLang . Oprócz tego obsługiwane są następujące wartości: tr - turecki
results	Liczba wyników wyszukiwania do zwrócenia w odpowiedzi na pojedyncze żądanie wyszukiwania.	10	Minimalną wartością jest 0. Maksymalną liczbą wyników zwracanych dla każdego żądania jest 1250.
start	Przesunięcie dla pierwszego zwracanego wyniku wyszukiwania.	0	Jeśli zostanie podana liczba ujemna, przyjmuje się domyślną wartość 0. Jeśli określona wartość jest większa lub równa liczbie wyników, nie jest zwracany żaden wynik.

Tabela 1. Parametry żądania wyszukiwania (kontynuacja)

Nazwa parametru	Opis	Wartość domyślna	Komentarze
stylesheet	Pełny adres URL arkusza stylów XSL służącego do formatowania wyników wyszukiwania.		<p>Jeśli wartością parametru output jest <code>htmlsnippet</code>, wartość parametru stylesheet jest ignorowana. W wartości należy używać znaków zmiany znaczenia odpowiednich dla adresów URL.</p> <p>Wskazany arkusz stylów XSL nie jest przetwarzany na serwerze mechanizmu wyszukiwania. W aplikacji klienckiej należy zastosować reguły transformacji znajdujące się w arkuszu stylów XSL do wyników wyszukiwania w postaci dostarczanych danych w formacie Atom. Aplikacją kliencką może być zgodna z transformacjami XSLT przeglądarka WWW, czytnik dostarczanych danych lub niestandardowa aplikacja XSLT.</p>

Wyniki wyszukiwania

Funkcja API `search` obsługuje wyniki wyszukiwania w postaci kanałów Atom 1.0 lub fragmentów kodu HTML.

Dostarczane dane w formacie Atom

Wygląd dostarczanych danych w przeglądarce można dostosować, określając w żądaniu wyszukiwania arkusz stylów XSL.

Fragmenty kodu HTML

Wyniki wyszukiwania w formacie HTML można osadzić na istniejącej stronie WWW. Aby to zrobić, należy udostępnić pole wyszukiwania, za pośrednictwem którego do serwera będzie trafiało żądanie **HTTP GET**:

Funkcja API `search` udostępnia też interfejs usługi, który zwraca dokument opisu OpenSearch i umożliwia aplikacjom klienckim wykrycie interfejsu wyszukiwania produktu IBM OmniFind Yahoo! Edition.

Jeśli podczas wykonywania żądania wyszukiwania wystąpi błąd, zwracany jest komunikat zawierający ID komunikatu o błędzie i opis błędu.

Odsyłacze pokrewne

“Odpowiedzi zawierające błąd” na stronie 25

Dostarczane dane w formacie Atom

Można zażądać zwrócenia wyników wyszukiwania w postaci danych w formacie Atom.

Aby uzyskać informacje o formacie Atom 1.0, patrz dokument The Atom Syndication Format pod adresem <http://atompub.org/rfc4287.html>. IBM OmniFind Yahoo! Edition wykorzystuje formaty danych OpenSearch 1.0, aby rozszerzyć format Atom o dodatkowe metadane konieczne do zwrócenia wyników wyszukiwania. Więcej informacji na temat formatów OpenSearch 1.0 zawiera publikacja OpenSearch response elements pod adresem <http://www.opensearch.org>.

W poniższej tabeli opisano elementy, które są zwracane jako wyniki wyszukiwania funkcji API `search`:

Tabela 2. Elementy formatu Atom i OpenSearch oraz wyniki zwracane przez funkcje API.

Elementy i atrybuty	Opis
/feed	Element kontenera dla danych i metadanych powiązanych z dostarczonymi danymi wyników wyszukiwania.
/feed/title	Wartość: Wyniki wyszukiwania dla zapytania ' <i>zapytanie</i> ' na kolekcji <i>nazwa_kolekcji</i>
/feed/link@href	Jeśli wartość atrybutu rel w elemencie href to: <ul style="list-style-type: none"> • self, wówczas odsyłacz prowadzi do adresu URL, za pomocą którego wygenerowano bieżący kanał danych. • first, wówczas odsyłacz prowadzi do pierwszego zbioru wyników wyszukiwania. • previous, wówczas odsyłacz prowadzi do poprzedniego zbioru wyników wyszukiwania, względem zbioru bieżącego. • next, wówczas odsyłacz prowadzi do następnego zbioru wyników wyszukiwania, względem zbioru bieżącego. • last, wówczas odsyłacz prowadzi do ostatniego zbioru wyników wyszukiwania. • alternate, wówczas odsyłacz prowadzi do tego zbioru wyników wyszukiwania w formacie alternatywnym. • search, wówczas odsyłacz wskazuje dokument opisu OpenSearch. • unconstrained, wówczas odsyłacz prowadzi do zbioru nieodfiltrowanych wyników wyszukiwania. Wyniki wyszukiwania mogą być odfiltrowane w celu usunięcia duplikatów lub na skutek przekroczenia limitu czasu wartościowania wprowadzonego zapytania.
/feed/author/name	Wartość: Usługa Web Service IBM OmniFind API
/feed/id	Adres URL wywołany przez aplikację kliencką w celu wygenerowania bieżącego kanału.
/feed/category	Zawiera informacje o kolekcji powiązanej z wynikami wyszukiwania.
/feed/category@term	Nazwa kolekcji, dla której wysłano to żądanie wyszukiwania (parametr collection w żądaniu wyszukiwania).
/feed/category@label	Patrz opis elementu /feed/category@term. Ten atrybut jest używany na potrzeby wyświetlania w czytnikach dostarczanych danych.
/feed/updated	Data i godzina wysłania zapytania. Wartość jest przedstawiana w formacie UTC: RRRR-MM-DDTgg:mm:ssZ.
/feed/opensearch:totalResults	Łączna liczba wyników dla wysłanego zapytania.
/feed/opensearch:Query	Zawiera informacje o zapytaniu wysłanym przez użytkownika.
/feed/opensearch:Query@role	Jeśli wartość atrybutu role to: <ul style="list-style-type: none"> • request: Wartość atrybutu searchTerms jest wysłanym zapytaniem (tylko jedno na dostarczane dane). • correction: Wartość atrybutu searchTerms reprezentuje propozycję dotyczącą pisowni. Dostarczane dane w formacie Atom mogą zawierać zero lub więcej propozycji dotyczących pisowni.
/feed/opensearch:Query@searchTerms	Reprezentuje wysłane zapytanie lub propozycję dotyczącą pisowni dla wysłanego zapytania, zwrócone przez serwer mechanizmu wyszukiwania.
/feed/opensearch:startIndex	Numer początkowego wyniku w wynikach wyszukiwania zwracanych w dostarczanych danych.

Tabela 2. Elementy formatu Atom i OpenSearch oraz wyniki zwracane przez funkcje API. (kontynuacja)

Elementy i atrybuty	Opis
/feed/opensearch:itemsPerPage	Liczba wyników wyszukiwania zwracanych w konkretnych dostarczanych danych.
/feed/entry	Zawiera informacje o pojedynczym wyniku wyszukiwania.
/feed/entry/category@term	Istnieje dla odsyłaczy wyróżnionych, a nie dla wyniku tekstowego. Wartość tego atrybutu to featured link.
/feed/entry/title	Tytuł wyniku.
/feed/entry/link	Definiuje odwołanie do zasobu wyniku wyszukiwania.
/feed/entry/link@rel	Jeśli wartość atrybutu rel to: <ul style="list-style-type: none"> • alternate: Wartość atrybutu href to identyfikator URI dokumentu wynikowego. • via: Wartość atrybutu href jest zbuforowaną wersją dokumentu wynikowego. <p>Dwa elementy link mogą istnieć w przypadku, gdy wartość atrybutu rel to via (jeśli typ oryginalnego dokumentu nie ma wartości text/html). Jeden element link reprezentuje buforowaną wersję oryginalnego dokumentu. Drugi element link reprezentuje wersję dokumentu w języku HTML (atrybut type ma wartość text/html). Elementy link z atrybutem rel o wartości via istnieją tylko wtedy, gdy włączone jest buforowanie.</p>
/feed/entry/link@href	Odsyłacz będący identyfikatorem URI dokumentu.
/feed/entry/link@type	Typ treści odsyłacza będącego identyfikatorem URI dokumentu.
/feed/entry/link@hreflang	Język odsyłacza będącego identyfikatorem URI dokumentu.
/feed/entry/opensearch:relevance	Wynik dokumentu.
/feed/entry/updated	Data ostatniej modyfikacji dokumentu. Wartość jest przedstawiana w formacie UTC: RRRR-MM-DDTgg:mm:ssZ.
/feed/entry/id	Identyfikator URI dokumentu.
/feed/entry/summary	Podsumowanie wygenerowane przez mechanizm wyszukiwania dla danego dokumentu.
/feed/entry/omnifind:field	Wartość metadanych dla wyszukiwania w polach. Możliwe wartości <i>pola</i> to: abstract, author, creator, description, doctype, fileext, keywords, language, owner, subject, title, url. Ten element jest stosowany tylko wtedy, gdy parametr oyeFieldFormat funkcji API search ma wartość true.
/feed/entry/omnifind:field	Wartość metadanych na potrzeby wyszukiwań dla pól.
/feed/entry/omnifind:field@name	Nazwa pola metadanych.

Przykład dostarczanych danych w formacie Atom

Poniższe, przykładowe wyniki wyszukiwania Atom 1.0 pokazują, co jest zwracane przez aplikację wyszukującą w odpowiedzi na zapytanie powodujące wyszukiwanie wszystkich dokumentów, które w polach metadanych keywords, author lub creator zawierają wyrażenie "narody zjednoczone". W aplikacji wyszukującej zapytanie wygląda następująco:

keywords:"narody zjednoczone" OR author:"narody zjednoczone" OR creator:"narody zjednoczone"

Format tego zapytania po zastosowaniu kodowania URL jest następujący:

```
http://nazwa_hosta:port/api/search?query=keywords%3A%22narody+
zjednoczone%22+OR+author%3A%22narody+zjednoczone%22+OR+creator%3A%22narody+zjednoczone%22
&collection=Default
```

Wynik zwrócony przez to zapytanie jest następujący:

```
<?xml version="1.0" encoding="utf-8" ?>
<feed xmlns="http://www.w3.org/2005/Atom"
  xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/"
  xmlns:omnifind="http://omnifind.ibm.yahoo.net/api/spec/1.0/">
  <title>Wyniki wyszukiwania dla zapytania 'creator:"narody zjednoczone" OR author:"narody zjednoczone"'
na kolekcji Default</title>
  <link href="http://nazwa_hosta:port/api/search?collection=
Default&query=creator:%22narody%20zjednoczone%22%20OR%20
author:%22narody%20zjednoczone%22" rel="self" type="application/atom+xml"/>
  <author>
  <name>Usługa Web Service IBM OmniFind API</name>
  </author>
  <id>http://nazwa_hosta:port/api/search?query=keywords%3A%22
narody+zjednoczone%22+40OR+author%3A%22narody+zjednoczone%22+OR+creator%3A%22narody+zjednoczone%22&collection=Default
  <category term="Default" label="Default" />
  <updated>2007-02-06T02:42:22Z</updated>
  <opensearch:totalResults>2</opensearch:totalResults>
  <opensearch:Query role="request" searchTerms="creator:"united nations" OR author:"united nations" />
  <opensearch:startIndex>1</opensearch:startIndex>
  <opensearch:itemsPerPage>2</opensearch:itemsPerPage>
  <entry>
  <link href="http://unbisnet.un.org/" rel="alternate" type="text/html" hreflang="en" />
  <link href="http://nazwa_hosta:port/search/?query=cache::http%3A%2F%2Funbisnet.un.org%2F&output=binary" rel="via" type="text/html" hreflang="en" />
  <opensearch:relevance>2.38</opensearch:relevance>
  <title type="html">UNBISnet - UN Bibliographic Information System</title>
  <updated>2006-02-06T19:21:05Z</updated>
  <id>http://unbisnet.un.org/</id>
  <summary type="html"><SPAN class="ellipsis">... </SPAN> Katalog <SPAN class="highlight">
  <SPAN class="hlTerm0">dokumentów i publikacji opracowanych przez Narody Zjednoczone</SPAN></SPAN>(UN), poindeksowanych pr
  bibliotekę UN Dag Hammarskjöld Library oraz Bibliotekę Biura Narodów Zjednoczonych w Genewie. Obejmuje publikacje komercyj
  <omnifind:creator type="html"><SPAN class="highlight"><SPAN class="hlTerm0">
  Narody Zjednoczone</SPAN></SPAN></omnifind:creator>
  <omnifind:author type="html">Autorstwo: <SPAN class="highlight"><SPAN class="hlTerm0">
  Narody Zjednoczone</SPAN>
  </SPAN></omnifind:author>
  </entry>
  <entry>
  <link href="http://testresult.un.org/" rel="alternate" type="text/html" hreflang="en" />
  <link href="http://nazwa_hosta:port/search/?query=cache::http%3A%2F%2Ftestresult.un.org%2F&output=
  binary" rel="via" type="text/html" hreflang="en" />
  <opensearch:relevance>2.08</opensearch:relevance>
  <title type="html">UN test result with only author matching</title>
  <updated>2006-02-06T19:21:05Z</updated>
  <id>http://testresult.un.org/</id>
  <summary type="html"><SPAN class="ellipsis">... </SPAN> Podsumowanie wyniku dla hasła <SPAN class="highlight">
  <SPAN class="hlTerm0">Narody Zjednoczone</SPAN></SPAN><SPAN class="ellipsis">... </SPAN>
  </summary>
  <omnifind:author type="html"><SPAN class="highlight"><SPAN class="hlTerm0">Narody Zjednoczone</SPAN>
  </SPAN></omnifind:author>
  </entry>
</feed>
```

Fragmety kodu HTML

Wyniki wyszukiwania mogą być zwracane jako fragmenty kodu HTML.

Fragment kodu HTML różni się od pełnej strony HTML, ponieważ nie zawiera wszystkich elementów wymaganych w przypadku pełnej strony HTML. Brak w nim znaczników <HTML> czy <BODY>. Zwrócony w wynikach wyszukiwania funkcji API fragment kodu

Format żądania

Następujące żądanie pozwala pobrać format dokumentu opisu OpenSearch:

`http://nazwa_hosta:8888/api/search/opensearchdescription`

Zostanie zwrócony dokument opisu OpenSearch. Na przykład:

```
<?xml version="1.0" encoding="utf-8"?>
<OpenSearchDescription xmlns="http://a9.com/-/spec/opensearch/1.1/"
  xmlns:omnifind="http://omnifind.ibm.yahoo.net/api/spec/1.0/">
  <ShortName>OmniFind</ShortName>
  <Description>Usługa Web Service interfejsu API dla mechanizmu wyszukiwania IBM OmniFind Enterprise</Description>
  <Url type="application/atom+xml"
    indexOffset="0"
    template="http://<hostname:port>/api/search?query={searchTerms}&results={count?}
      &start={startIndex?}&
      resultLang={language?}&collection={omnifind:collection}&queryLang={omnifind:queryLang?}&
      locale={omnifind:locale?}"/>
  <Url type="text/html"
    indexOffset="0"
    template="http://<hostname:port>/api/search?query={searchTerms}&results={count?}
      &start={startIndex?}&
      resultLang={language?}&collection={omnifind:collection}&queryLang={omnifind:queryLang?}&
      locale={omnifind:locale?}&output=htmlsnippet"/>
  <Query role="example"
    searchTerms="cat"
    omnifind:collection="Default"/>
  <Query role="example"
    searchTerms="cat OR mouse"
    omnifind:collection="Default"
    omnifind:stylesheet="http://my.server.com/stylesheets/atom.xsl"
    count="20"/>
  <!--result language-->
  <Language>ar</Language>
  <Language>cs</Language>
  ...
</OpenSearchDescription>
```

Wytyczne

Dokument opisu OpenSearch jest rozszerzony o przestrzeń nazw XML `http://omnifind.ibm.yahoo.net/api/spec/1.0/`. Prefiks przestrzeni nazw to `omnifind`. Rozszerzenie to jest niezbędne do zdefiniowania określonych parametrów szablonu żądania wyszukiwania, które nie zostały zdefiniowane w podstawowym zbiorze nazw parametrów wyszukiwania OpenSearch.

Chodzi tu o elementy `Url` i `Query`. Każdy element `Url` określa atrybut szablonu. Wartość atrybutu zawiera szablon adresu URL wyszukiwania dla aplikacji klienckich. Dokument opisu OpenSearch zawiera następujące szablony:

- Szablon zwracający odpowiedź typu `application/atom+xml` w przypadku odpowiedzi w postaci dostarczanych danych Atom.
- Szablon zwracający odpowiedź typu `text/html` w przypadku odpowiedzi w postaci fragmentów kodu HTML.

Dla każdego elementu `Url` wartość atrybutu `indexOffset` jest ustawiana na 0. Dzieje się tak, ponieważ przy domyślnych ustawieniach OpenSearch pierwszy wynik wyszukiwania ma numer 1. Opis dokumentu OpenSearch nadpisuje wartość domyślną, ponieważ w produkcji OmniFind Yahoo! Edition wyniki wyszukiwania są numerowane od 0.

Każdy składnik `Query` zawiera przykładowe zapytania, które mogą być wykonywane przez klienty wyszukiwania. Przykładowe zapytania używają zdefiniowanego niestandardowego

prefiksu przestrzeni nazw, omnifind. Jedno z przykładowych zapytań wysyła żądanie wyszukania w kolekcji domyślnej (Default) słowa kluczowego `cat` (kot). Drugie wysyła żądanie wyszukania w tej kolekcji słów kluczowych `cat OR mouse` (kot LUB mysz). Do sformatowania wyników jest wykorzystywany arkusz stylów znajdujący się pod adresem <http://my.server.com/stylesheets/atom.xsl>, a w każdej odpowiedzi jest zwracanych 20 wyników.

Więcej informacji na temat składni i semantyki dokumentu opisu wyszukiwania OpenSearch można znaleźć pod adresem <http://www.opensearch.org/Specifications/OpenSearch/1.1>.

Funkcje API do dodawania i usuwania dokumentów

W aplikacji klienckiej można używać funkcji API do dodawania dokumentów do kolekcji i ich usuwania.

Żądania API odpowiedzialne za dodawanie i usuwanie dokumentów są standardowymi żądaniami HTTP. Zabezpieczanie tych żądań odbywa się przy użyciu podstawowego uwierzytelniania protokołu HTTP. Wartość ID użytkownika jest ignorowana. Hasło funkcji API można uzyskać w konsoli administracyjnej. Skontaktuj się z administratorem serwera wyszukiwania lub w konsoli administracyjnej, na stronie Zarządzaj kolekcjami kliknij opcję **Zmień hasło**.

Odpowiedziami funkcji API dla dokumentów są standardowe komunikaty odpowiedzi protokołu HTTP. Jeśli żądanie spowoduje błąd, treść komunikatu odpowiedzi będzie zawierała szczegóły dotyczące tego błędu. W przypadku powodzenia wykonania żądania treść komunikatu jest pusta.

Zadania pokrewne

“Generowanie haseł funkcji API” na stronie 27

Odsyłacze pokrewne

“Odpowiedzi zawierające błąd” na stronie 25

“Przykłady w języku Java wpisywane w wierszu komend” na stronie 29

Format żądań funkcji API do dodawania i usuwania dokumentów

Do dodawania dokumentów do kolekcji można używać żądań **HTTP POST**, a do ich usuwania - żądań **HTTP DELETE**.

Żądanie dodania dokumentu

Funkcja API `addDocument` jest żądaniem **HTTP POST**. Powoduje ona dodanie lub zastąpienie dokumentu w określonej kolekcji. Żądanie jest synchroniczne. Powrót z żądania oznacza, że dokument został pomyślnie dodany do kolekcji, chyba że zwracany jest komunikat o błędzie.

Następujący przykład pokazuje żądanie `addDocument`:

```
POST /api/document HTTP/1.1
Host: nazwa_hosta:port
action: addDocument
collection: Default
docId: document1
docType: application/x-mspowerpoint
docLang: en
lastModified: 2006-01-26T16:37:44-04:00
Authorization: Basic OnY2eEddyQWM9
Content-Length: 2048
```

[tutaj treść]

Dokumentów dodanych do kolekcji przez funkcję API `addDocument` nie można śledzić w oknie Status dokumentu w konsoli administracyjnej. Jeśli podczas dodawania dokumentu wystąpi jakikolwiek błąd, jest on odzwierciedlany w odpowiedzi HTTP.

Poza tym, jeśli wartość `docId` nie jest poprawnym identyfikatorem URI, nie będzie można kliknąć dokumentu na stronie wyników wyszukiwania.

Dodawanie do dokumentów wartości metadanych

Jeśli administrator skonfigurował pola metadanych dla kolekcji, podczas dodawania do niej dokumentów można przypisywać im wartości. Żądanie `addDocument` nie może definiować typu ani atrybutów pola metadanych (wielkości te konfiguruje się za pomocą konsoli administracyjnej), ale może przypisywać polom wybrane wartości.

W poniższym przykładzie administrator skonfigurował dwa pola metadanych, o nazwach `product` i `price`:

```
name = "product"
type = "text"
name = "price"
type = "decimal"
```

Żądanie `addDocument` może wskazać te pola metadanych za pośrednictwem dodatkowych parametrów i przypisać im żądane wartości. Nazwy pól metadanych poprzedza się parametrem **X-** w celu oznaczenia, że są to pola zdefiniowane przez użytkownika. Na przykład:

```
POST /api/document HTTP/1.1
Host: nazwa_hosta:port
action: addDocument
collection: Default
docId: document1
docType: application/x-mspowerpoint
docLang: en
X-product: movie
X-price: 19.99
lastModified: 2006-01-26T16:37:44-04:00
Authorization: Basic OnY2eEduQWM9
Content-Length: 2048
```

[tutaj treść]

Aby pobrać listę wszystkich pól metadanych dostępnych w danej kolekcji, należy użyć funkcji `API metadataFields`. W celu pobrania listy wszystkich kolekcji, do których można dodawać dokumenty, trzeba użyć funkcji `API collectionsList`.

Żądanie usunięcia dokumentu

Funkcja `API deleteDocument` jest żądaniem **HTTP DELETE**. Powoduje ono usunięcie dokumentu ze wskazanej kolekcji. Żądanie jest synchroniczne. Jednak po powrocie z funkcji nie ma gwarancji, że dokument jest już niedostępny do wyszukiwania.

Następujący przykład pokazuje żądanie `deleteDocument`:

```
DELETE /api/document HTTP/1.1
Host: nazwa_hosta:port
action: deleteDocument
collection: Default
docId: document1
Authorization: Basic OnY2eEduQWM9
```

Czas, po którym dany dokument nie jest już dostępny na potrzeby wyszukiwania zależy od obciążenia serwera wyszukiwania w momencie wysłania żądania usunięcia.

Odsyłacze pokrewne

“Funkcje API dotyczące pól metadanych” na stronie 21

Parametry żądań funkcji API do dodawania i usuwania dokumentów

W żądaniach dodania i usunięcia dokumentu można używać wielu parametrów.

W poniższej tabeli opisano obsługiwane parametry dla żądań dotyczących dokumentu:

Tabela 3. Obsługiwane parametry dla żądań dodania lub usunięcia dokumentu

Nazwa parametru	Opis	Wartość domyślna	Obsługiwane działanie	Komentarze
action	Działanie, które ma zostać przeprowadzone.		Wszystkie działania	Wymagany. Obsługiwane wartości: addDocument oraz deleteDocument.
collection	Nazwa kolekcji do zaktualizowania.		Wszystkie działania	Wymagany. Wartość ta musi być zakodowana w formacie UTF-8 i zawierać znaki zmiany znaczenia odpowiednie dla adresów URL.
Content-length	Wielkość treści dokumentu, który ma zostać dodany, wyrażona w bajtach.		addDocument	Wymagany. Wartość ta musi być równa lub większa niż zero bajtów.
docId	Identyfikator dokumentu.		Wszystkie działania	Wymagany. Jeśli użytkownicy mają mieć możliwość pobrania dokumentu poprzez kliknięcie go na liście wyników wyszukiwania, wartość ta musi być poprawnym identyfikatorem URI. W wartości należy używać znaków zmiany znaczenia odpowiednich dla adresów URL.
docKnownLang	Znany język treści komunikatu (treści dokumentu).	Określana przez serwer.	addDocument	Wartość parametru docKnownLang jest używana, aby wymusić na serwerze użycie określonego języka jako języka dokumentu. Listę obsługiwanych wartości podano w opisie parametru locale .
docLang	Zapasowy język treści komunikatu (treści dokumentu).	Określana przez serwer.	addDocument	Wartość parametru docLang jest używana w przypadku, gdy serwer nie może określić języka dokumentu i nie określono wartości parametru docKnownLang . Listę obsługiwanych wartości podano w opisie parametru locale .
docType	Zapasowy typ i podtyp treści komunikatu (treści dokumentu).		addDocument	Wymagany. Jeśli serwer nie może określić typu dokumentu, jako typu używa się wartości parametru docType . Jego format to <i>typ/podtyp</i> , np. text/html. Poprawne wartości znajdują się w dokumencie RFC1341.

Tabela 3. Obsługiwane parametry dla żądań dodania lub usunięcia dokumentu (kontynuacja)

Nazwa parametru	Opis	Wartość domyślna	Obsługiwane działanie	Komentarze
lastModified	Data i godzina ostatniej modyfikacji dokumentu.	Data i godzina odbioru dokumentu.	addDocument	Wartość musi być określona w formacie ISO-8601: <i>RRRR-MM-DDTgg:mm:ssTZD</i> . Przykład: 2006-01-26T16:37:44-04:00 lub 2006-01-26T20:37:44Z.
locale	Ustawienia narodowe klienta.	Ustawienia narodowe serwera.	Wszystkie działania	Zwraca komunikaty w języku zgodnym z ustawieniami narodowymi klienta. Obsługiwane wartości: de_DE - niemiecki en_US - angielski es_ES - hiszpański fr_FR - francuski hu_HU - węgierski it_IT - włoski ja_JP - japoński ko_KR - koreański nl_NL - holenderski pl_PL - polski pt_PT - portugalski pt_BR - portugalski (Brazylia) sv_SE - szwedzki zh_CN - chiński (uproszczony) zh_TW - chiński (tradycyjny)
X-nazwa pola	Wartość pola metadanych.		addDocument	Zarówno nazwa parametru, jak i jego wartość musi być zakodowana w formacie UTF-8 i zawierać znaki zmiany znaczenia odpowiednie dla adresów URL.

Administracyjne funkcje API

Aplikacja kliencka może stosować administracyjne funkcje API do uruchamiania i zatrzymywania przeszukiwaczy, uzyskiwania listy wszystkich pól metadanych skonfigurowanych dla kolekcji oraz uzyskiwania nazw wszystkich kolekcji w systemie wyszukiwania.

Żądania administracyjnych funkcji API są standardowymi żądaniami HTTP. Zabezpieczenie tych żądań odbywa się przy użyciu podstawowego uwierzytelniania protokołu HTTP. Wartość ID użytkownika jest ignorowana. Hasło funkcji API można uzyskać w konsoli administracyjnej. Skontaktuj się z administratorem serwera wyszukiwania lub w konsoli administracyjnej kliknij kolejno opcje: **Zarządzaj systemem** → **Zarządzaj uwierzytelnianiem**.

Odpowiedziami funkcji API dla dokumentów są standardowe komunikaty odpowiedzi protokołu HTTP. Jeśli żądanie spowoduje błąd, treść komunikatu odpowiedzi będzie zawierała szczegóły dotyczące tego błędu. W przypadku powodzenia wykonania żądania treść komunikatu jest pusta.

Zadania pokrewne

“Generowanie haseł funkcji API” na stronie 27

Funkcja API do zarządzania przeszukiwaczem

Funkcja API do zarządzania przeszukiwaczem służy do uruchamiania lub zatrzymywania przeszukiwacza.

Żądanie uruchomienia lub zatrzymania przeszukiwacza jest standardowym żądaniem **HTTP POST**. Funkcja API wykorzystuje podstawowe uwierzytelnianie HTTP, aby zabezpieczyć żądania. Wartością hasła jest token API pobrany z konsoli administracyjnej. Wartość nazwy użytkownika jest ignorowana.

Format żądania HTTP jest następujący:

```
POST /api/admin HTTP/1.1
Host: nazwa_hosta:port
action: dzialanie
locale: ustawienia_narodowe
collection: nazwa_kolekcji
crawlType: typ_przeszukiwacza
Authorization: Basic haslo
```

Tabela 4. Parametry żądania zarządzania przeszukiwaczem

Parametry	Komentarze
action	Wymagany. Działanie, które ma zostać przeprowadzone. Obsługiwane wartości to startCrawl oraz stopCrawl.

Tabela 4. Parametry żądania zarządzania przeszukiwaczem (kontynuacja)

Parametry	Komentarze
locale	Opcjonalny. Ustawienia narodowe klienta. Domyślnie mają one tę samą wartość, co ustawienia narodowe serwera. Obsługiwane wartości: de_DE - niemiecki en_US - angielski es_ES - hiszpański fr_FR - francuski hu_HU - węgierski it_IT - włoski ja_JP - japoński ko_KR - koreański nl_NL - holenderski pl_PL - polski pt_PT - portugalski pt_BR - portugalski (Brazylia) sv_SE - szwedzki zh_CN - chiński (uproszczony) zh_TW - chiński (tradycyjny)
collection	Wymagany. Nazwa kolekcji, do której należy przeszukiwacz. Wartość ta powinna być zakodowana w formacie UTF-8 i zawierać znaki zmiany znaczenia odpowiednie dla adresów URL.
crawlType	Wymagany. Typ przeszukiwacza. Obsługiwane wartości to file, jdbc oraz web.

Narzędzie manageCrawler

Przeszukiwacze można też uruchamiać i zatrzymywać za pomocą narzędzia **manageCrawler**. Aby uzyskać informacje na temat administrowania przeszukiwaczami z poziomu wiersza komend, należy wprowadzić w wierszu komend serwera wyszukiwania komendę `manageCrawler -?` lub zapoznać się z dokumentacją na temat administrowania produktem IBM OmniFind Yahoo! Edition.

Przykład uruchamiania przeszukiwacza

W tym przykładzie funkcja API do zarządzania przeszukiwaczami jest wykorzystywana do wystosowania żądania uruchomienia przeszukiwacza systemu plików dla kolekcji domyślnej (Default):

```
POST /api/admin HTTP/1.1
Host: http://JKEnterprises.com:8888
action: startCrawl
collection: Default
crawlType: file
Authorization: Basic 6eKvCms=
```

Aby wystosować to samo żądanie z poziomu wiersza komend, należy wprowadzić następującą komendę:

```
manageCrawler -h http://JKEnterprises.com:8888 -a start -c Default -t file -p
"6eKvCms=" -o output.txt
```

Przykład zatrzymywania przeszukiwacza

W tym przykładzie funkcja API do zarządzania przeszukiwaczami jest wykorzystywana do wystosowania żądania uruchomienia przeszukiwacza sieci WWW dla kolekcji Employees z wykorzystaniem francuskich ustawień narodowych:

```
POST /api/admin HTTP/1.1
Host: http://JKEnterprises.com:8888
action: stopCrawl
locale: fr_FR
collection: Employees
crawlType: web
Authorization: Basic 6eKvCms=
```

Aby wystosować to samo żądanie z poziomu wiersza komend, należy wprowadzić następującą komendę:

```
manageCrawler -h http://JKEnterprises.com:8888 -a stop -l fr_FR -c Employees -t
web -p "6eKvCms=" -o output.txt
```

Funkcje API dotyczący pól metadanych

Za pomocą funkcji API `metadatefields` można pobrać listę wszystkich pól metadanych, które zostały skonfigurowane dla danej kolekcji.

Format żądania

Używając funkcji API `metadatefields` w połączeniu z żądaniami wyszukiwania, można ustalić, które pola są dostępne do przeszukiwania. Funkcji API `metadatefields` można też używać z żądaniami `addDocument` w celu sprawdzenia, dla jakich pól można określić wartości pól metadanych.

Żądanie pobrania nazw wszystkich pól metadanych w kolekcji jest standardowym żądaniem **HTTP GET**. Żądanie `metadatefields` składa się z kombinacji następujących właściwości:

- Nazwa hosta
- Port
- Ścieżka
- Parametry żądania, kolekcja par nazwa-wartość () rozdzielonych znakami ampersand (&)

Nazwa hosta to nazwa hosta serwera mechanizmu wyszukiwania. Port to numer portu aplikacji wyszukującej, w przypadku wykorzystywania funkcji API do określania pól metadanych, które mają zostać przeszukane, lub numer portu aplikacji administrującej, w przypadku wykorzystywania funkcji API do określania pól metadanych podczas dodawania dokumentów do kolekcji.

Ścieżka wysyłania żądania to:

- `/api/search/metadatefields`. To żądanie zwraca pola zdefiniowane przez użytkownika, predefiniowane oraz wbudowane (wszystkie pola dostępne dla wyszukiwania).
- lub `/api/document/metadatefields`. To żądanie zwraca tylko pola zdefiniowane przez użytkownika, ponieważ są to jedyne pola, którym aplikacje klienckie mogą nadawać wartości podczas dodawania dokumentów.

Format odpowiedzi

Odpowiedź na żądanie funkcji API `metadatefields` ma format XML. Dla każdego pola metadanych w kolekcji odpowiedź zawiera jego nazwę i typ. Jeśli dla pola zostały skonfigurowane atrybuty, określające na przykład, czy pole może być przeszukiwane po jego nazwie lub czy wartość pola może pojawić się w wynikach wyszukiwania, dane atrybutów są również uwzględnione w odpowiedzi.

Parametry żądania

Tabela 5. Parametry żądania pobrania pól metadanych

Parametry	Komentarze
collection	Wymagany. Nazwa kolekcji, z której mają zostać pobrane nazwy pól metadanych. Wartość ta powinna być zakodowana w formacie UTF-8 i zawierać znaki zmiany znaczenia odpowiednie dla adresów URL.
locale	Opcjonalny. Ustawienia narodowe klienta. Domyślnie mają one tę samą wartość, co ustawienia narodowe serwera. Obsługiwane wartości: de_DE - niemiecki en_US - angielski es_ES - hiszpański fr_FR - francuski hu_HU - węgierski it_IT - włoski ja_JP - japoński ko_KR - koreański nl_NL - holenderski pl_PL - polski pt_PT - portugalski pt_BR - portugalski (Brazylia) sv_SE - szwedzki zh_CN - chiński (uproszczony) zh_TW - chiński (tradycyjny)

Przykładowe żądanie i odpowiedź

Poniższe żądanie pobiera nazwy wszystkich pól metadanych dostępnych do przeszukiwania, jakie zostały skonfigurowane dla kolekcji `Sample`:

`http://JKEnterprises.server.com:8888/api/search/metadatefields?collection=Sample`

Przykładowa odpowiedź wskazuje, że dla kolekcji `Sample` zostały skonfigurowane dwa pola metadanych (`price` i `product`):

```
<?version = 1.0 encoding="UTF-8"?>
<fields version="1.0">
<collection>Sample</collection>
<field>
  <name>author</name>
  <type>text</type>
</field>
<field>
```

```
    <name>doctype</name>
    <type>text</type>
</field>
<field>
    <name>docdate</name>
    <type>date</type>
</field>
...
</fields>
```

Odsyłacze pokrewne

“Format żądań funkcji API do dodawania i usuwania dokumentów” na stronie 15

Funkcja API generująca listę kolekcji

Za pomocą funkcji API collections można pobrać nazwy wszystkich kolekcji istniejących w systemie wyszukiwania.

Żądanie pobrania nazw wszystkich kolekcji jest standardowym żądaniem **HTTP GET**. Żądanie funkcji collections składa się z kombinacji następujących właściwości:

- Nazwa hosta
- Port
- Ścieżka

Nazwa hosta to nazwa hosta serwera mechanizmu wyszukiwania. Port to numer portu dla aplikacji wyszukiwującej. Ścieżka wysyłania żądania to zawsze /api/search/collections. Funkcji API collections można używać zarówno dla żądań wyszukiwania, jak i dla żądań dokumentów, w celu ustalenia, na której kolekcji mają zostać przeprowadzone działania.

Na przykład następujące żądanie powoduje uzyskanie nazw wszystkich kolekcji:

```
http://JKEnterprises.server.com:8889/api/search/collections
```

Odpowiedź ma format XML. W tym przykładzie widać, że w systemie znajdują się dwie kolekcje, o nazwach Marketing i Sales. Odpowiedź wskazuje, że kolekcję Marketing można przeszukiwać i że jest to domyślna kolekcja serwera wyszukiwania. Kolekcji Sales nie można przeszukiwać (o tym, czy kolekcję można przeszukiwać decyduje administrator).

```
<?xml version="1.0" encoding="UTF-8"?>
<collections version="1.0">
  <collection enabled="true" default="true">
    <name>Marketing</name>
  </collection>
  <collection enabled="false">
    <name>Sales</name>
  </collection>
</collections>
```

Odsyłacze pokrewne

“Format żądań funkcji API do dodawania i usuwania dokumentów” na stronie 15

Odpowiedzi zawierające błąd

W przypadku niepowodzenia wykonania żądania API zwracana jest odpowiedź zawierająca błąd.

Zawierające błąd odpowiedzi na żądania API są standardowymi kodami odpowiedzi protokołu HTTP. Treść odpowiedzi HTTP składa się z komunikatów o błędach, z których każdy zawiera identyfikator oraz szczegółowy opis błędu.

Wszystkie żądania API zwracają informacje o błędach w formacie XML. Funkcja API `search` może też zwracać informacje o błędach w postaci fragmentów kodu HTML, w zależności od wartości parametru **output** użytego w żądaniu wyszukiwania.

W poniższym przykładzie pokazano odpowiedź zawierającą błąd w formacie XML:

```
<APIResponse version="1.0">
  <Error>
    <Message>
      <Id>IQQR0016E</Id>
      <Text>Nie można przetworzyć żądania interfejsu API wyszukiwania.</Text>
    </Message>
    <Message>
      <Id>IQQS0032E</Id>
      <Text>Nie można przetworzyć zapytania ze względu na jego niepoprawną składnię.</Text>
    </Message>
  </Error>
</APIResponse>
```

W odpowiedzi funkcji API, która zawiera błąd, wyświetlany jest tylko tekst komunikatu. Pełna wersja komunikatu (w tym sekcje wyjaśnienia i odpowiedzi użytkownika) znajduje się w dokumentacji produktu.

Odsyłacze pokrewne

“Żądania wyszukiwania” na stronie 3

“Wyniki wyszukiwania” na stronie 7

“Funkcje API do dodawania i usuwania dokumentów” na stronie 15

Kody odpowiedzi HTTP

Standardowe kody odpowiedzi błędów HTTP określają ogólny typ zaistniałego błędu. Dodatkowe informacje na temat błędu są zawarte w treści komunikatu HTTP.

W poniższej tabeli umieszczono kody odpowiedzi HTTP określające błąd wraz z opisem zaistniałego warunku błędu.

Tabela 6. Kody odpowiedzi HTTP i sytuacje, w których mogą wystąpić dane błędy

Kod i nazwa błędu	Przyczyna	Przykłady
400 - Złe żądanie	Dane umieszczone w treści żądania nie mają oczekiwanego formatu lub nie zawierają poprawnych wartości.	Klient nie umieszcza w żądaniu wyszukiwania wymaganego parametru collection lub podaje niepoprawną nazwę kolekcji.

Tabela 6. Kody odpowiedzi HTTP i sytuacje, w których mogą wystąpić dane błędy (kontynuacja)

Kod i nazwa błędu	Przyczyna	Przykłady
401 - Brak uprawnień	Podczas przetwarzania żądania kontrola dostępu przeprowadzana przez implementację usługi API REST kończy się odmową dostępu.	W żądaniu HTTP mającym na celu dodanie dokumentu do kolekcji podano nieprawidłowe hasło interfejsu API.
404 - Nie znaleziono	<ol style="list-style-type: none"> 1. Identyfikator URI podany w żądaniu, z uwzględnieniem parametrów, nie pasuje do żadnego z identyfikatorów URI określonych w interfejsie API REST. 2. Składniowo poprawny identyfikator URI wskazuje zasób, którego implementacja usługi API REST nie może znaleźć. 	<ol style="list-style-type: none"> 1. Element ścieżki lub nazwa parametru może być zapisana z błędem. 2. Adres URL zapisany w przeglądarce jako zakładka wskazuje zasób, który został usunięty.
405 - Niedozwolona metoda	Usługa API REST nie obsługuje operacji zakładanej przez metodę HTTP dla zasobu wskazywanego przez identyfikator URI występujący w żądaniu.	Żądanie PUT na identyfikatorze URI określającym w interfejsie API REST jedynie komendy GET i POST.
500 - Błąd serwera	Podczas przetwarzania żądania wystąpił wyjątek wewnętrzny, wynikający z nieprawidłowej konfiguracji.	Taka sytuacja ma prawo zaistnieć podczas testów, ale nie powinna mieć miejsca w środowisku produkcyjnym.

Generowanie haseł funkcji API

W celu używania administracyjnych funkcji API wymagane jest hasło do interfejsu API.

Aby poznać hasło funkcji API, należy skontaktować się z administratorem serwera wyszukiwania. Hasło funkcji API jest wyświetlane na stronie Zarządzaj uwierzytelnianiem w konsoli administracyjnej.

Aby wygenerować nowe hasło funkcji API, należy skontaktować się z administratorem serwera wyszukiwania. Jeśli możesz uzyskać dostęp do konsoli administracyjnej, wykonaj następujące kroki, aby wygenerować nowe hasło funkcji API:

1. W konsoli administracyjnej kliknij kolejno opcje: **Zarządzaj systemem** → **Zarządzaj uwierzytelnianiem**.
2. W oknie Zarządzaj uwierzytelnianiem kliknij opcję **Wygeneruj nowe hasło API**.
3. Skopiuj hasło funkcji API i wklej je w kodzie aplikacji.

Po wygenerowaniu nowego hasła stare hasło interfejsu API nie jest poprawne w istniejących aplikacjach używających administracyjnych funkcji API. Jeśli aplikacja nie może uzyskać dostępu do systemu wyszukiwania, upewnij się, że hasło funkcji API w aplikacji jest zgodne z hasłem funkcji API wyświetlanym w konsoli administracyjnej.

Odsyłacze pokrewne

“Funkcje API do dodawania i usuwania dokumentów” na stronie 15

“Administracyjne funkcje API” na stronie 19

Przykłady w języku Java, XSL i PHP

Na podstawie przedstawionych przykładów w językach Java, XSL i PHP można tworzyć własne aplikacje wyszukiujące.

Przykłady w języku Java, XSL i PHP zostały umieszczone w katalogu `GŁÓWNY_KATALOG_INSTALACYJNY/examples`.

Pakiet SDK języka Java nie jest dostarczany razem z mechanizmem wyszukiwania. Nie należy tworzyć aplikacji za pomocą załączonej wirtualnej maszyny języka Java. Załączona wirtualna maszyna języka Java obejmuje tylko środowisko wykonawcze platformy Java.

Przykłady w języku Java wpisywane w wierszu komend

Można użyć udostępnionych przykładów funkcji API Java, które pomogą zbudować niestandardową aplikację wyszukiującą Java.

Przykłady funkcji API Java i powiązanych plików klasy Java są udostępniane w katalogu `INSTALL_ROOT/examples/java/commandline`, gdzie `INSTALL_ROOT` to katalog instalacyjny IBM OmniFind Yahoo! Edition. Aby uruchomić przykład Java, należy użyć wiersza komend w celu przejścia do katalogu `INSTALL_ROOT/examples/java`.

Przed uruchomieniem dowolnego przykładu dla wiersza komend należy dodać plik `whitney_core.jar` do używanej instrukcji `CLASSPATH`. Plik `whitney_core.jar` znajduje się w katalogu `GŁÓWNY_KATALOG_INSTALACYJNY/lib`.

Search

Przykładowa komenda wiersza komend **Search** uruchamia wyszukiwanie i zwraca wyniki wyszukiwania w postaci danych w formacie Atom, który jest wyświetlany w oknie wiersza komend. Jeśli lokalny plik XSL jest określony jako argument, arkusz stylów XSL zostanie zastosowany do zwróconych danych w formacie Atom, a sformatowany rezultat zostanie również wyświetlony w oknie wiersza komend.

Opis składni:

```
Search nazwa_hosta port nazwa_kolekcji zapytanie lokalna_ścieżka_do_pliku_XSL
```

Na przykład:

```
commandline.Search localhost 8080 Default NFL "C:\\Program Files\\IBM\\OmniFindYahooEdition\\exam
```

AddDocument

Przykładowa komenda wiersza komend **AddDocument** dodaje dokument do kolekcji.

Opis składni:

```
AddDocument nazwa_hosta port nazwa_kolekcji identyfikator_dokumentu  
plik_lokalny typ_mime nazwa_użytkownika hasło
```

Na przykład:

```
commandline.AddDocument localhost 8080 Default "My MS Word Document" "C:\\temp\\My Document.doc" a
```

DeleteDocument

Przykładowa komenda wiersza komend **DeleteDocument** usuwa dokument z kolekcji.

Opis składni:

```
DeleteDocument nazwa_hosta port nazwa_kolekcji identyfikator_dokumentu  
nazwa_uzytkownika haslo
```

Na przykład:

```
commandline.DeleteDocument localhost 8080 Default "My MS Word Document" admin "fhWJhgo="
```

Odsyłacze pokrewne

“Funkcje API do dodawania i usuwania dokumentów” na stronie 15

Przykładowy arkusz stylów XSL

Arkusze stylów XSL definiują standardowy format wyświetlania danych wyjściowych XML, taki jak format Atom.

Przykładowy plik arkusza stylów XSL znajduje się w katalogu *INSTALL_ROOT/examples/xsl*. Przykładowy arkusz stylów XSL przekształca format Atom na format tekstowy.

Przykładowa aplikacja wyszukująca PHP

Można użyć udostępnionego przykładu aplikacji PHP, aby utworzyć niestandardową aplikację wyszukującą PHP.

Przykładowa aplikacja wyszukująca PHP znajduje się w katalogu *INSTALL_ROOT/examples/php*.

W celu uruchomienia przykładowej aplikacji wymagana jest aplikacja PHP oraz kompatybilny z aplikacją PHP serwer WWW zainstalowany w używanym systemie. Po zainstalowaniu wymienionych komponentów należy utworzyć kontekstowy katalog główny aplikacji wyszukującej PHP w katalogu głównym serwera WWW. Przykładowo, utwórz katalog OYE w katalogu głównym serwera WWW. Następnie skopiuj i wklej zawartość katalogu *INSTALL_ROOT/examples/php* do nowego katalogu OYE. Edytuj plik *search.php*, aby zmienić zmienną *\$oyeUrl* na adres URL używanego systemu IBM OmniFind Yahoo! Edition.

Przykładowa aplikacja wyszukująca PHP zawiera arkusz stylów, dwa obrazy i dwa pliki PHP: *search.php* i *oye.php*. Arkusz stylów zawiera klasy CSS sterujące wyglądem przykładowej aplikacji wyszukującej PHP. Aplikacja wyszukująca PHP wykorzystuje dwa pliki graficzne: *fp_bg.png* i *front-page-header.png* na banerze strony aplikacji. Plik *search.php* zawiera kod HTML służący do wyświetlania formularza i wyników wyszukiwania. Plik *oye.php* zawiera funkcje służące do przeprowadzania wyszukiwania przy użyciu funkcji API wyszukiwania REST oraz do przetwarzania wyników.

Przykładowo, aby wyświetlić odsyłacze wyróżnione w używanej aplikacji wyszukującej PHP, należy uruchomić wyszukiwanie przy użyciu funkcji *search(\$queryString)*, a następnie przenieść zmienną *\$feed* do funkcji *getFeaturedLinks(\$feed)*.

Następujące funkcje są dostępne w pliku włączanym *oye.php*:

search(\$queryString)

Zwraca obiekt wskazujący na początek danych XML.

getTotalResults(\$feed)

Zwraca łączną liczbę wyników w postaci liczby całkowitej.

getSearchTerms(\$feed)

Zwraca łańcuch terminów wyszukiwania.

getSpellCorrections(\$feed)

Zwraca tablicę łańcuchów przedstawiających poprawki pisowni.

getStartIndex(\$feed)

Zwraca pierwszy wynik w postaci liczby całkowitej.

getItemsPerPage(\$feed)

Zwraca liczbę wyników wyszukiwania wyświetlanych na każdej stronie w postaci liczby całkowitej.

getSearchResults(\$feed)

Zwraca tablicę obiektów wyników przedstawiających wyniki wyszukiwania.

getFeaturedLinks(\$feed)

Zwraca tablicę obiektów wyników przedstawiających odsyłacze wyróżnione.

Uwagi

Uwagi

Niniejsza publikacja została przygotowana z myślą o produktach i usługach oferowanych w Stanach Zjednoczonych.

IBM może nie oferować w innych krajach produktów, usług lub opcji, omawianych w tej publikacji. Informacje o produktach i usługach dostępnych w danym kraju można uzyskać od lokalnego przedstawiciela firmy IBM. Jakakolwiek wzmianka na temat produktu, programu lub usługi firmy IBM nie oznacza, że może być zastosowany jedynie ten produkt, ten program lub ta usługa firmy IBM. Zamiast nich można zastosować ich odpowiednik funkcjonalny, pod warunkiem, że nie narusza to praw własności intelektualnej firmy IBM. Jednakże cała odpowiedzialność za ocenę przydatności i sprawdzenie działania produktu, programu lub usługi pochodzących od producenta innego niż IBM spoczywa na użytkowniku.

IBM może posiadać patenty lub złożone wnioski patentowe na towary i usługi, o których mowa w niniejszej publikacji. Używanie tego dokumentu nie daje żadnych praw do tych patentów. Wnioski o przyznanie licencji można zgłaszać na piśmie pod adresem:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Zapytania dotyczące zestawów znaków dwubajtowych (DBCS) należy kierować do lokalnych działów własności intelektualnej firmy IBM (IBM Intellectual Property Department) lub wysłać je na piśmie na adres:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

Poniższy akapit nie obowiązuje w Wielkiej Brytanii, a także w innych krajach, w których jego treść pozostaje w sprzeczności z przepisami prawa miejscowego: FIRMA INTERNATIONAL BUSINESS MACHINES CORPORATION DOSTARCZA TĘ PUBLIKACJĘ W TAKIM STANIE, W JAKIM SIĘ ZNAJDUJE, "AS IS", BEZ UDZIELANIA JAKIKOLWIEK GWARANCJI (W TYM TAKŻE RĘKOJMI), WYRAŻNYCH LUB DOMNIEMANYCH, A W SZCZEGÓLNOŚCI DOMNIEMANYCH GWARANCJI PRZYDATNOŚCI HANDLOWEJ ORAZ PRZYDATNOŚCI DO OKREŚLONEGO CELU LUB GWARANCJI, ŻE PUBLIKACJA NIE NARUSZA PRAW OSÓB TRZECICH. Ustawodawstwa niektórych krajów nie dopuszczają zastrzeżeń dotyczących gwarancji wyraźnych lub domniemanych w odniesieniu do pewnych transakcji, w takiej sytuacji powyższe zdanie nie ma zastosowania.

Informacje zawarte w niniejszej publikacji mogą zawierać nieścisłości techniczne lub błędy drukarskie. Informacje te są okresowo aktualizowane, a zmiany te zostaną ujęte w kolejnych wydaniach tej publikacji. IBM zastrzega sobie prawo do wprowadzania ulepszeń i/lub zmian w produktach i/lub programach opisanych w tej publikacji w dowolnym czasie, bez wcześniejszego powiadomienia.

Jakiegokolwiek wzmianki na temat stron internetowych nie należących do firmy IBM zostały podane jedynie dla wygody użytkownika i nie oznaczają, że firma IBM w jakikolwiek sposób firmuje te strony. Materiały dostępne na tych stronach nie są częścią materiałów opracowanych dla tego produktu IBM, a użytkownik korzysta z nich na własną odpowiedzialność.

IBM ma prawo do korzystania i rozpowszechniania informacji przysłanych przez użytkownika w dowolny sposób, jaki uzna za właściwy, bez żadnych zobowiązań wobec ich autora.

Licencjobiorcy tego programu, którzy chcieliby uzyskać informacje na temat programu w celu: (i) wdrożenia wymiany informacji między niezależnie utworzonymi programami i innymi programami (łącznie z tym opisywanym) oraz (ii) wykorzystywania wymienianych informacji, powinni skontaktować się z:

IBM Corporation J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003 U.S.A.

Informacje takie mogą być udostępnione, o ile spełnione zostaną odpowiednie warunki, w tym, w niektórych przypadkach, uiszczenie odpowiedniej opłaty.

Licencjonowany program opisany w tym dokumencie oraz wszystkie inne licencjonowane materiały dostępne dla tego programu są dostarczane przez IBM na warunkach określonych w Umowie IBM z Klientem, Międzynarodowej Umowie Licencyjnej IBM na Program lub w innych podobnych umowach zawartych między IBM i użytkownikami.

Wszelkie dane dotyczące wydajności zostały zebrane w kontrolowanym środowisku. W związku z tym rezultaty uzyskane w innych środowiskach operacyjnych mogą się znacząco różnić. Niektóre pomiary mogły być dokonywane na systemach będących w fazie rozwoju i nie ma gwarancji, że pomiary te wykonane na ogólnie dostępnych systemach dadzą takie same wyniki. Niektóre z pomiarów mogły być estymowane przez ekstrapolację. Rzeczywiste wyniki mogą być inne. Użytkownicy powinni we własnym zakresie sprawdzić odpowiednie dane dla ich środowiska.

Informacje dotyczące produktów innych firm zostały uzyskane od dostawców tych produktów z opublikowanych przez nich zapowiedzi lub innych powszechnie dostępnych źródeł. Firma IBM nie testowała tych produktów i nie może potwierdzić dokładności pomiarów wydajności, kompatybilności ani żadnych innych danych związanych z tymi produktami. Pytania dotyczące możliwości produktów innych firm należy kierować do dostawców tych produktów.

Publikacja ta zawiera przykładowe dane i raporty używane w codziennej działalności biznesowej. W celu kompleksowego zilustrowania tej działalności podane przykłady zawierają nazwy osób, firm i ich produktów. Wszystkie te nazwiska/nazwy są fikcyjne i jakakolwiek ich zbieżność z prawdziwymi nazwiskami/nazwami jest całkowicie przypadkowa.

LICENCJA NA PRAWA AUTORSKIE:

Niniejsza publikacja zawiera przykładowe aplikacje w kodzie źródłowym, ilustrujące techniki programowania w różnych systemach operacyjnych. Użytkownik może kopiować, modyfikować i rozpowszechniać te programy przykładowe w dowolnej formie bez uiszczania opłat, w celu rozbudowy, użytkowania, handlowego lub w celu rozpowszechniania aplikacji zgodnych z aplikacyjnym interfejsem programowym dla tego systemu operacyjnego, dla którego napisane były programy przykładowe. Programy przykładowe nie zostały gruntownie

przetestowane. Firma IBM nie może zatem gwarantować lub sugerować niezawodności, użyteczności i funkcjonalności tych programów.

Każda kopia lub dowolna część programów przykładowych, albo też dowolna praca pochodna, musi zawierać poniższą informację o prawach autorskich:

Oracle[®] Outside In Search Export, Copyright © 1992, 2007, Oracle. Wszelkie prawa zastrzeżone.

Oracle[®] Outside In HTML Export, Copyright © 1992, 2007, Oracle. Wszelkie prawa zastrzeżone.

Znaki towarowe

Ta sekcja zawiera spis znaków towarowych firmy IBM i niektórych znaków towarowych firm trzecich.

Więcej informacji na temat znaków towarowych IBM zawiera strona WWW <http://www.ibm.com/legal/copytrade.shtml>.

Poniższe nazwy są znakami towarowymi lub zastrzeżonymi znakami towarowymi innych firm:

Adobe, Acrobat, PostScript i wszystkie znaki towarowe oparte na Adobe są zastrzeżonymi znakami towarowymi lub znakami towarowymi Adobe Systems Incorporated w Stanach Zjednoczonych i/lub innych krajach.

Java i wszystkie znaki towarowe i logo związane z językiem Java są znakami towarowymi lub zastrzeżonymi znakami towarowymi firmy Sun Microsystems, Inc. w Stanach Zjednoczonych i/lub innych krajach.

Microsoft, Windows, Windows NT i logo Windows są znakami towarowymi firmy Microsoft Corporation w Stanach Zjednoczonych i/lub innych krajach.

Intel, logo Intel, Intel Inside, logo Intel Inside, Intel Centrino, logo Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium i Pentium są znakami towarowymi lub zastrzeżonymi znakami towarowymi firmy Intel Corporation lub firm zależnych w Stanach Zjednoczonych i innych krajach.

Linux jest znakiem towarowym Linusa Torvaldsa w Stanach Zjednoczonych i/lub innych krajach.

Nazwy innych firm, produktów i usług mogą być znakami towarowymi lub znakami usług firm trzecich.

Indeks

A

- addDocument, funkcja API
 - format żądania 15
 - odpowiedzi 15
 - opis 15
 - parametry żądania 17
 - zabezpieczenia 15
- administracyjne funkcje API
 - odpowiedzi 19
 - opis 19
 - zabezpieczenia 19
- API, przykłady
 - aplikacje wyszukujące PHP 30
 - funkcje PHP 30
 - Java 29
 - PHP 29
 - XSL 29
 - XSL, arkusz stylów 30

C

- collections, funkcja API
 - pobieranie nazw kolekcji 23

D

- deleteDocument, funkcja API
 - format żądania 16
 - odpowiedzi 15
 - opis 15
 - parametry żądania 17
 - zabezpieczenia 15

E

- elementy dostarczania danych 7

F

- format wyników
 - dokument opisu OpenSearch 11
 - elementy dostarczanych danych w formacie Atom 7
 - przykład dostarczanych danych w formacie Atom 9
 - przykładowy fragment kodu HTML 10
- format wyników w postaci danych Atom
 - elementy 7
 - przykład 9
 - rozszerzenia OpenSearch 7
- format wyników wyszukiwania
 - dokument opisu OpenSearch 11
 - elementy dostarczanych danych w formacie Atom 7
 - przykład dostarczanych danych w formacie Atom 9
 - przykładowy fragment kodu HTML 10
- format żądania
 - collections, funkcja API 23

- format żądania (*kontynuacja*)
 - dodawanie dokumentów 15
 - dodawanie wartości metadanych 16
 - funkcja API do zarządzania przeszukiwaczem 19
 - metadatafields, funkcja API 21
 - usuwanie dokumentów 16
 - wyszukiwanie dokumentów publicznych 3
 - zarządzanie przeszukiwaczem 19
- format żądania dodania dokumentu
 - dodawanie dokumentów 15
 - dodawanie wartości metadanych 16
 - komenda POST protokołu HTTP 15
- format żądania usunięcia dokumentu
 - komenda POST protokołu HTTP 16
 - usuwanie dokumentów 16
- format żądania wyszukiwania
 - dokumenty publiczne 3
 - GET, komenda protokołu HTTP 3
 - parametry 4
 - przykłady adresów URL 3
 - przykłady żądań HTTP GET 3
- funkcja API wyszukiwania
 - dokument opisu OpenSearch 11
 - format wyników 7
 - format wyników w postaci danych Atom 7
 - GET, żądania protokołu HTTP 3
 - odpowiedzi 7
 - opis 3
 - parametry żądania 4
 - przykład wyników w postaci danych Atom 9
 - przykładowy fragment kodu HTML 10

G

- GET, komenda protokołu HTTP
 - collections, funkcja API 23
 - format wyszukiwania 3
 - metadatafields, funkcja API 21
 - parametry żądania wyszukiwania 4
 - pobieranie nazw kolekcji 23
 - pobieranie nazw pól metadanych 21
 - przykłady wyszukiwania 3

H

- hasło funkcji API
 - addDocument, funkcja API 15
 - administracyjne funkcje API 19
 - deleteDocument, funkcja API 15
 - generowanie 27

I

- interfejs API
 - addDocument 15
 - administrowanie 19

- interfejs API (*kontynuacja*)
 - deleteDocument 16
 - kody odpowiedzi HTTP 25
 - kolekcje 23
 - metadatafields 21
 - odpowiedzi zawierające błąd 25
 - przegląd 1
 - REST 1
 - wyszukiwanie 3
 - zarządzanie przeszukiwaczem 19

K

- kody odpowiedzi HTTP 25
- komenda POST protokołu HTTP
 - format dodawania dokumentów 15
 - format dodawania metadanych 16
 - format usuwania dokumentów 16
 - format zarządzania przeszukiwaczem 19
 - parametry zarządzania przeszukiwaczem 19
 - parametry żądania dodania dokumentu 17
 - parametry żądania usunięcia dokumentu 17
 - przykłady dodawania dokumentów 15
 - przykłady dodawania metadanych 16
 - przykłady usuwania dokumentów 16
 - przykłady zarządzania przeszukiwaczem 19

M

- manageCrawler, komenda 19
- metadatafields, funkcja API
 - pobieranie nazw pól metadanych 21

O

- odpowiedzi zawierające błąd
 - format wyjściowy 25
 - kody odpowiedzi HTTP 25
 - przykład 25
- OpenSearch
 - dokument opisu 11
 - format wyników w postaci danych Atom 7
 - format żądania 11

P

- parametry żądania
 - addDocument, funkcja API 17
 - collections, funkcja API 23
 - deleteDocument, funkcja API 17
 - funkcja API wyszukiwania 4
 - metadatafields, funkcja API 21
- przeszukiwacze
 - manageCrawler, komenda 19

- przeszukiwacze (*kontynuacja*)
 - przykładowe żądania API 19
 - uruchamianie 19
 - zatrzymywanie 19
- przykładowa aplikacja wyszukująca PHP 30
- przykłady w języku Java wpisywane w wierszu komend
 - dodawanie dokumentów 29
 - usuwanie dokumentów 30
 - wyszukiwanie 29

W

- wartości metadanych
 - dodawanie do dokumentów
 - addDocument, funkcja API 16
- wyniki wyszukiwania w postaci fragmentów kodu HTML
 - przykład 10

Ż

- żądania wyszukiwania
 - przykłady żądań HTTP GET 3



Drukowane w USA