CINECA
Consorzio Interuniversitario

IRIS – REST API

Area funzionale : Iris – IR/RM

IRIS REST API

Versione 1.0.0.0

versione	data	Modifiche principali	Stato	Autore/i
0.0.0.1	12/02/2015	Prima stesura (READ-ONLY)	BOZZA	Meriggi P.
0.0.0.2	04/03/2015	Modifiche minori	BOZZA	Meriggi P.
0.0.0.3	10/04/2015	Modifiche minori – Prima revisione Bollini	BOZZA	Meriggi P.
0.0.0.4	20/04/2015	Finale API lettura – Seconda revisione Bollini	BOZZA	Meriggi P.
1.0.0.0	15/05/2015	Aggiunta API anagrafica RM	BOZZA	Meriggi P.
1.0.0.0	04/11/2015	Aggiunta API anagrafica RM per CF	FINALE	Meriggi P.

Indice

Introduzione: IRIS REST API

Attraverso queste API in formato REST si vuole dare la possibilità ad un accesso programmatico alle principali entità di IRIS, come le *Collections*, le *Communities*, gli *Items*, i *Bitstreams*, *Inputforms* e le anagrafiche.

IRIS possiede già due modalità per l'accesso ad alcune informazioni caratteristiche (OAI-PMH e i web service SOAP di consultazione catalogo ereditati da UGOV-RI) ma con queste API si vuole dare un accesso alle risorse in modo più completo utilizzando il paradigma messo a disposizione dai servizi REST.

1. REST ENDPOINTS

Lo scopo di queste API è quello di dare accesso alle principali risorse di IRIS utilizzando il concetto di RISORSA secondo i dettami dell'approccio REST.

E' naturale che l'oggetto caratteristico di IRIS venga comunque incapsulato in una entità che non è direttamente un'immagine dell'oggetto sul DB. Attraverso le API sarà perciò possibile estrarne le informazioni caratteristiche e i legami con le altre entità di IRIS.

L'implementazione segue i dettami del lavoro noto come *Richardson Maturity Model*[1] livello 3.

Attraverso la proprietà *Accept* dell'*HTTP Header* è possibile specificare il formato dati della risposta "*application/json*" per **JSON** (e "*application/xml*" per **XML** in futuro).

Attraverso invece i parametri in *query-string*, da aggiungere alla richiesta REST, è possibile variare il comportamento del servizio.

Il più comune di queste API è "*?expand*": ad ogni chiamata il servizio risponde con un sotto insieme di dati "standard" senza aggiungere informazioni che potrebbero non essere utilizzate diminuendo il costo della chiamata sia lato server che come occupazione di banda. Ogni endpoint fornirà comunque un elenco delle proprietà che possono essere "espanse" con l'intento di completare le informazioni standard con quelle aggiuntive utili caso per caso. Un esempio tipico è l'utilizzo del parametro "*?expand=all*" che provvede a restituire tutte le informazioni partendo dalla risorsa richiesta (oggetti padre, figli e metadati aggiuntivi). Altro caso è l'utilizzo del parametro con la richiesta di estensione multipla, come "*?expand=metadata,bistreams*".

Tutte le API sono protette dall'autenticazione **HTTP Basic Auth** sotto **SSL**. Ad ogni richiesta è necessario fornire nell'HEADER lo username e la password secondo lo standard citato. Inoltre, per poter avere le stesse funzionalità o permessi presenti nell'interfaccia utente di IRIS è possibile aggiungere nell'HEADER (come coppia nome/valore) della richiesta HTTP altri 3 parametri (opzionali) che "simulano" la visione personale/completa e il "login as" con la corrispettiva visione:

- **scope** è visione personale=**ROLE_USER** oppure visione completa=**ROLE_ADMIN**
- **on-behalf-of** è il "login as" ed è necessario fornire lo **username**
- **on-behalf-of-scope** come lo scope

Naturalmente l'utente che effettua la richiesta (HTTP BASIC) deve essere in possesso dell'autorizzazione per poter cambiare visione e poter effettuare il "login as".

Si ricorda inoltre che nelle API in cui è prevista la paginazione esiste un limite pre-impostato rispetto al quale si limita la cardinalità del recupero dati per evitare eccessivi sovraccarichi del sistema.

Per i dettagli implementativi, gli esempi e la descrizione esaustiva delle API, vedere IRIS IR 1.0.0.0 REST API documentation (CINECA_IR_REST_API_Doc.htm).

[1] http://martinfowler.com/articles/richardsonMaturityModel.html

1.1 Communities endpoint

Le Communities sono le macrotipologie o nodi non foglia nella vecchia alberatura UGOV.

Metodo	Endpoint	Descrizione
GET	/rest/api/v1/communities	Ritorna l'elenco delle communities in IR (paginato)
GET	/rest/api/v1/communities/{communityId}	Ritorna la community specificata dato il suo ID
GET	/rest/api/v1/communities/{communityId}/collections	Ritorna l'elenco delle collections a partire dalla community specificata dal suo ID
GET	/rest/api/v1/communities/{communityId}/communities	Ritorna l'elenco delle communities a partire dalla community specificata dal suo ID (paginato)

1.2 Collections endpoint

Le collections in IRIS sono i nodi foglia dell'intero albero delle communities che contengono gli items (o prodotti secondo la terminologia UGOV).

Metodo	Endpoint	Descrizione
GET	/rest/api/v1/collections	Ritorna l'elenco delle collections in IR (paginato)
GET	/rest/api/v1/collections/{collectionId}	Ritorna la collection specificata dato il suo ID
GET	/rest/api/v1/collections/{collectionId}/items	Ritorna l'elenco degli items a partire dalla collection specificata dal suo ID (paginato)

1.3 Items endpoint

Gli item (prodotti) rappresentano in IRIS i metadati di una particolare entità (prodotto) che, unitamente ai bitstream, completano le informazioni sull'oggetto.

Metodo	Endpoint	Descrizione
GET	/rest/api/v1/items	Ritorna l'elenco degli item in IR (paginato)
GET	/rest/api/v1/items/{itemId} *oppure* /rest/api/v1/items/{namingAuthority}/{localname}	Ritorna l'item specificato dato il suo ID
GET	/rest/api/v1/items/{itemId}/metadata *oppure* /rest/api/v1/items/{namingAuthority}/{localname}/metadata	Ritornano i metadati dell'item specificato
GET	/rest/api/v1/items/{itemId}/metadata/{metadataKey} *oppure* /rest/api/v1/items/{namingAuthority}/{localname}/metadata/{metadataKey}	Ritorna il metadato richiesto per l'item specificato
GET	/rest/api/v1/items/{itemId}/streams *oppure* /rest/api/v1/items/{namingAuthority}/{localname}/streams	Ritornano i bitstream dell'item specificato
GET	/rest/api/v1/items/{itemId}/history *oppure* /rest/api/v1/items/{namingAuthority}/{localname}/histories	Ritorna l'item specificato dal proprio handle
POST	/rest/api/v1/items/search	Ritorna una lista di item secondo il DTO di ricerca (paginato)
POST	/rest/api/v1/items/ids	Ritorna una lista id di item a partire da un valore noto (sequenziale)

1.4 Bitstreams endpoint

I Bitstreams sono file. Hanno un nome, una dimensione (in byte), e un formato.

Tipicamente in IRIS, il Bitstream rappresentano il cosiddetto FULL-TEXT o altri MEDIA TYPE.

Alcuni file sono il file effettivo che è stato caricato da interfaccia utente (con etichetta bundleName: ORIGINAL), altri invece sono file generati da IRIS per utilizzi interni o per procedure di ottimizzazione/gestione.

Per sapere a chi appartiene il bitstream è possibile richiedere l'entità padre attraverso l'utilizzo dell'expand (ad esempio "?expand=parent").

Metodo	Endpoint	Descrizione
GET	/rest/api/v1/streams	Ritorna l'elenco dei bitstream in IR (paginato)
GET	/rest/api/v1/streams/{bitstreamId}	Ritorna il bitstream specificato dato il suo ID
GET	/rest/api/v1/streams/{bitstreamId}/attachment	Ritornano gli attachment del bitstream specificato dato il suo ID

1.5 Inputform endpoint

L'input form determina come gli item sono visualizzati e inseriti in IRIS.

Metodo	Endpoint	Descrizione
GET	/rest/api/v1/inputforms/{inputformId}	Ritorna l'inputform specificata dato il suo ID
GET	/rest/api/v1/inputforms/{inputformId}/form	Ritornano i dati dell'inputform specificato dato il suo ID
GET	/rest/api/v1/inputforms/{inputformId}/valuepairs	Ritornano i valuepairs dell'inputform specificato il suo ID

1.6 Ance endpoint

Ritorna l'anagrafica della rivista dato il suo crisid.

Metodo	Endpoint	Descrizione
GET	/rest/api/v1/ance/{crisId}	Ritorna la rivista con il crisId specificato
POST	/rest/api/v1/ance/search	Effettua una ricerca dato il crisId o l'anceId (oggetto Ance Search DTO)

1.7 Persons endpoint

Ritorna l'anagrafica delle persone dato il suo crisid o l'id o il codice fiscale.

Metodo	Endpoint	Descrizione
GET	/rm/restservices/api/v1/personsbyrpid/{crisId} *oppure* /rm/restservices/api/v1/personsbyid/{id} *oppure* /rm/restservices/api/v1/personsbycf/{cf}	Ritorna la persona con il crisId/id specificato
GET	/rm/restservices/api/v1/personsbyrpid/{crisId}/positions *oppure* /rm/restservices/api/v1/personsbyid/{id}/positions *oppure* /rm/restservices/api/v1/personsbycf/{cf}/positions	Ritorna la carriera completa della persona specificata

Metodo	Endpoint	Descrizione
GET	/rm/restservices/api/v1/personsbyrpid/{crisId}/positioncurrent *o* /rm/restservices/api/v1/personsbyid/{id}/positioncurrent oppure /rm/restservices/api/v1/personsbycf/{cf}/positioncurrent	Ritorna la carriera attuale della persona specificata

2. Modelli/Oggetti

Ecco l'elenco degli oggetti (alcuni possono essere parziali). Per le versioni complete vedere la documentazione html.

Community Object

{"copyrightText","introductoryText","sidebarText","shortDescription","collections","subcommunities","name","handle","type","id","expand","self","parentCommunity","collections","logo""}

Collection Object

{"parentCommunities","copyrightText","introductoryText","sidebarText","shortDescription","name","handle","type","id","expand","parentCommunityList","license","logo","countItems","self"}

Item Object

{"id","owner","lastModified","collection","submitterID","submitterNetID","identifierToDisseminate","workFlowValidationStatus","simpleItemStatus","workFlowValidationRule","collectionHandle","inputFormId","withdrawn","archived","snapshot","inWorkFlow","inWorkspace","earlyDraft","submitter","name","type","self","expand","metadata","bitstreams","history","lookupValues"}

MetadataEntry Object

{"key", [{"value", "authority", "place"}]}

Owner Object

{ "id","lastName","firstName","netid","email"}

Submitter Object

{ "id","lastName","firstName","netid","email"}

Bitstream Object

{"format","description","mimeType","bundleName","checkSum","sizeBytes","sequenceId","retrieveLink","name","type","id","expand","parent","licenseType","self"}

InputForm Object

{"formName","rows","numberPages","valuePairs"}

Row Object

{"required","pageNumber","repeatable","hint","inputType","warning""dcSchema","dcElement","label","dcQualifier","typeBind","noupdate","rowNumber","id","expand","self"}

Value Pairs Object

{"name",["key",”value”]}

Ance Object

{"casaEditrice","codice","natura","validoAl","validoDal","dateSent","titoli","issn","paese","submitter","eissn","lingua","codiceCRIS"}

Person Object (parziale)

{"type","id","crisid","firstName","lastName","addressSet","contactSet","personElementSet","dateMap","gaDictionaryMap","parentPersonLinkElementSet","childPersonLinkElementSet","uniqueIdentifier","uuid"}

Career Person Object (parziale)

{"type","temporary","discriminator","startDate","endDate","organizationUnit","positionType"}