



Infrastruttura
e servizi digitali per il
patrimonio culturale

Servizi di Content Processing Avanzato I.PaC

Manuale tecnico

Versione 1.0
23/12/2025

VERSIONI			
DATA	VERSIONE	DESCRIZIONE	CAP/SEZ. MODIFICATI
23/12/2025	1.0	Pubblicazione della prima versione del documento	-

Tabella 1- Tabella delle Versioni

Sommario

1. Introduzione.....	6
2. API CPA.....	8
2.1. AI Image Enhance.....	13
2.1.1. API da richiamare	13
2.2. OCR/ICR.....	15
2.2.1. API da richiamare	16
2.3. Speech To Text (STT).....	17
2.3.1. Api da richiamare	17
2.4. Ricostruzione articoli.....	18
2.4.1. API da richiamare	19
2.5. NER per Classificazione Automatica.....	20
2.5.1. API da richiamare	20
2.6. Tagging Automatico dei Testi	21
2.6.1. API da richiamare	22
2.7. Estrazione Entità Immagini	23
2.7.1. API da richiamare	23
2.8. Anagrafiche	24
2.8.1. API da richiamare	24
2.9. Ricerca/Associazione Entità a Soggettario di Firenze.....	26
2.9.1. API da richiamare	26
2.10. Associazione di topic con dei tag per un'immagine.....	27
2.10.1. API da richiamare	27
2.11. Analisi e/o Creazione struttura fisica risorsa.....	28
2.11.1. API da richiamare	28
2.12. Creazione Sommario (ToC) per Monografie.....	29
2.12.1. API da richiamare	30

2.13.	Creazione abstract da testo	31
2.13.1.	API da richiamare	32
2.14.	Creazione abstract da immagine.....	34
2.14.1.	API da richiamare	34
2.15.	Language Translation	35
2.15.1.	API da richiamare	35
2.16.	Text To Speech (TTS).....	37
2.16.1.	API da richiamare	37
2.17.	Riconoscimento delle immagini simili.....	38
2.17.1.	API da richiamare	39
2.18.	Immagini simili	39
2.18.1.	API da richiamare	40
2.19.	OMR	40
2.19.1.	API da richiamare	41
2.20.	Generazione Audio da XML Musicali	42
2.20.1.	API da richiamare	42
2.21.	Formattazione testo	43
2.21.1.	API da richiamare	43
2.22.	Adattamento Latino per Epigrafi.....	44
2.22.1.	API da richiamare	44
3.	Request e response CPA.....	46
3.1.	Request CPA	46
3.2.	Response CPA	50

Indice delle tabelle

<i>Tabella 1- Tabella delle Versioni</i>	2
Tabella 2- Sintesi Primitive e propedeuticità.....	11
Tabella 3- Schema primitive e relativi servizi, lingue e tipologie	13
Tabella 4 - Servizi disponibili per la Primitiva Image Ehnance	14
Tabella 5 - Servizi disponibili per la Primitiva OCR e la Primitiva ICR.....	17
Tabella 6 - Servizi disponibili per la Primitiva Speech To Text	18
Tabella 7 - Servizi disponibili per la Primitiva NER per Classificazione Automatica.....	21
Tabella 8 - Servizi disponibili per la Primitiva Tagging Automatico dei Testi.....	23
Tabella 9 - Servizi disponibili per la Primitiva Anagrafiche.....	25
Tabella 10 - Servizi disponibili per la Primitiva Analisi e/o Creazione Struttura Fisica.....	29
Tabella 11 - Servizi disponibili per la Primitiva Creazione Sommario (ToC) per Monografie	31
Tabella 12 - Soglie utili a definire la lunghezza dell'abstract.....	32
Tabella 13 - Servizi disponibili per la Primitiva Abstract da Testo	33
Tabella 14 - Servizi disponibili per la Primitiva Abstract da Immagini	35
Tabella 15 - Servizi disponibili per la Primitiva Language Translation.....	36
Tabella 16 - Servizi disponibili per la Primitiva Text To Speech	38
Tabella 17 - Lista delle lingue disponibili per la Primitiva Text To Speech.....	38
Tabella 18 - Servizi disponibili per la Primitiva OMR.....	42
Tabella 19 - Servizi disponibili per la Primitiva Generazione Audio da XML Musicali.....	43

1. Introduzione

Le API di Content Processing Avanzato (CPA) si integrano all'interno di I.PaC come parte delle funzionalità avanzate per l'elaborazione delle Risorse Digitali. Come già indicato all'interno del Catalogo dei Servizi di I.PaC, tali servizi avanzati di elaborazione delle Risorse Digitali consentono di generare:

- nuovi Oggetti Digitali/media, (es. testi estratti per Speech-to-Text da fonti sonore; testo elettronico estratto per ICR-ML da immagini di testi Manoscritti; ecc.);
- indici di entità estratte automaticamente dai testi elettronici (anche derivati).

Essi possono essere utilizzati singolarmente, oppure orchestrati tramite opportune logiche e parametrizzazioni e messi a disposizione di soggetti accreditati che avranno la possibilità, attraverso l'integrazione di singole API, di scegliere singoli servizi da implementare all'interno dei propri client, rispondendo quindi alle seguenti esigenze:

- **efficienza e riduzione dei costi**, grazie all'automazione di attività ripetitive volte all'estrazione concettuale e alla soggettazione, nonché riducendo, grazie alla possibilità di gestire grandi volumi di risorse digitali in maniera scalabile, i tempi di catalogazione e indicizzazione manuale;
- **supporto all'interoperabilità e agli standard**, garantendo il costante allineamento a standard nazionali e internazionali di descrizione e indicizzazione, la facilitazione dell'integrazione con cataloghi, repository digitali e sistemi di LOD (*Linked Open Data*) e una maggiore riusabilità dei dati in contesti diversi (biblioteche, archivi, digital humanities).

Ulteriore scopo cui rispondono i servizi messi a disposizione nell'ambito del Content Processing Avanzato di I.PaC è quello di garantire la valorizzazione del patrimonio digitale culturale italiano grazie alla possibilità di arricchire semanticamente le Risorse conferite dai sistemi, rafforzando, contestualmente, il ruolo del Nuovo Soggettario come riferimento autorevole nel contesto digitale e dell'intelligenza artificiale.

Tale documento ha lo scopo di illustrare le API di CPA in modo da rendere agevole la loro fruizione da parti dei sistemi che aderiscono ad I.PaC. Esso, in particolare, è così strutturato:

- nel capitolo 2, vengono presentate le API di CPA implementate in I.PaC, fornendo sia una descrizione dei servizi con le relative propedeuticità sia una panoramica dei modelli di intelligenza artificiale che possono essere utilizzata nell'invocazione dei servizi;
- nel capitolo 3, viene illustrata la request generica utile alla chiamata delle API di CPA con la relativa response.

2. API CPA

Le API CPA descritte nei paragrafi che seguono possono essere visualizzate nello Swagger dell'ambiente di preproduzione. Tale Swagger è consultabile al link:

<https://ispc-preprod.prod.os01.ocp.cineca.it/docs>

La tabella sottostante riporta, in particolare, uno schema riassuntivo di tutte le API CPA disponibili ad oggi, che saranno dettagliate nei paragrafi successivi, indicandone la relativa propedeuticità:

API	Descrizione	Propedeuticità
AI Image Enhance	Consente di migliorare la risoluzione e la qualità delle immagini, correggendo problemi e migliorando l'esperienza visiva complessiva.	Nessuna
OCR	Consente di riconoscere e convertire caratteri stampati in testo digitale o codici di caratteri.	<i>AI Image Enhance</i> (non obbligatoria)
ICR	Consente di riconoscere e convertire caratteri scritti a mano in testo digitale o codici di caratteri.	<i>AI Image Enhance</i> (non obbligatoria)
Speech To Text (STT)	Consente di estrarre il contenuto di un file audio, o l'audio di un file video, in formato testuale.	Nessuna
Ricostruzione articoli	Consente di associare ad ogni titolo e sottotitolo di un articolo di un periodico il relativo corpo testuale.	<i>OCR</i>
NER per Classificazione Automatica	Tramite l'elaborazione del linguaggio naturale, estrae dal testo informazioni rilevanti, e ad ogniuna assegna una categoria specifica.	Dipende dal tipo di risorsa in elaborazione (Monografia = <i>OCR</i> , Periodico = <i>Ricostruzione Articoli</i> , Audio o Video = <i>Speech to Text</i>)
Tagging Automatico dei Testi	Estrae temi astratti ed entità nominali da testi, etichettando anche le risorse digitali con termini provenienti dal Nuovo Soggettario di Firenze e/o da Iconclass.	Dipende dal tipo di risorsa in elaborazione (Monografia = <i>OCR</i> , Periodico = <i>Ricostruzione Articoli</i> , Audio o Video = <i>Speech to Text</i>), con in aggiunta <i>NER per Classificazione Automatica</i>
Estrazione Entità Immagini	Consente di riconoscere e isolare, in un'immagine, delle entità presenti all'interno dell'immagine stessa.	<i>Database Vettoriale Entità</i>
Anagrafiche	Consente l'estrazione delle informazioni anagrafiche delle entità nominali individuate a partire da un file di testo fornito in input, con l'obiettivo di arricchire il grafo della conoscenza con le anagrafiche estratte.	Dipende dal tipo di risorsa in elaborazione (Monografia = <i>OCR</i> , Periodico = <i>Ricostruzione Articoli</i> , Audio o Video = <i>Speech to Text</i>)

Ricerca/Associazione entità a Soggettario di Firenze	Consente, a partire da un insieme di entità passate in input, di trovare la corrispondenza con quelle presenti all'interno del soggettario di Firenze. Questo permette di avere una rappresentazione di entità validata dal Soggettario.	<i>Estrazione Entità Immagini, Filtro Entità Rappresentative</i>
Associazione di topic con dei tag per un'immagine	Consente di trovare le coordinate delle entità all'interno di un'immagine, con l'obiettivo di visualizzare le entità all'interno dell'immagine o di estrarne i relativi box per processamenti futuri.	Nessuna
Analisi e/o Creazione struttura fisica risorsa	Consente di estrarre la sequenza numerica delle pagine della risorsa digitale.	OCR
Creazione Sommario (ToC) per Monografie	Consente di creare il sommario di una monografia, sia esso già presente nel testo (in questo caso il sommario viene identificato all'interno della risorsa digitale e da essa estratto) o non presente (in questo caso viene ricostruito, analizzando la monografia e incrociando titoli dei capitoli, o paragrafi e sottocapitoli, con i numeri di pagina).	OCR, Analisi e/o Creazione Struttura Fisica
Creazione abstract da testo	Consente di estrarre le informazioni più rilevanti e significative da un testo e presentarle in modo sintetico.	Dipende dal tipo di risorsa in elaborazione (Monografia = OCR, Periodico = Ricostruzione Articoli, Audio o Video = Speech to Text)
Creazione abstract da immagine	Consente di sintetizzare quanto rappresentato in un'immagine.	<i>Estrazione Entità Immagini</i>
Language Translation	Consente di convertire il testo scritto digitale da una lingua in un'altra lingua.	Dipende dal tipo di risorsa in elaborazione (Monografia = Abstract da Testo, Periodico = Creazione abstract da immagine)
Text To Speech (TTS)	Consente di convertire il testo scritto digitale in audio, sfruttando le informazioni sulla lingua contenute nei metadati.	Dipende dal tipo di risorsa in elaborazione (Monografia = Abstract da Testo, Periodico = Creazione abstract da immagine)
Riconoscimento delle immagini simili (Database Vettoriale Immagini)	Rappresentando le immagini all'interno di un database vettoriale, permette di ottimizzare le operazioni sulle immagini stesse, come quelle di confronto o di recupero immagini.	Nessuna
Immagini simili	Permette di cercare immagini simili a quella fornita in input, secondo politiche di similarità di varia natura.	<i>Database Vettoriale Immagini</i>
OMR	Consente di convertire spartiti musicali stampati in formato "music xml" e di generarne un'immagine pdf. Inoltre, viene confrontata l'immagine pdf prodotta in output con	Nessuna

	l'immagine originale per verificare la qualità dell'OMR.	
Generazione Audio da XML Musicali	Consente di creare un file audio a partire da file in formato musicxml, generato dalla primitiva OMR.	OMR
Formattazione testo estratto (schede)	Consente di formattare correttamente e correggere un testo scritto, ad esempio estratto tramite OCR.	OCR
Adattamento Latino per Epigrafi	Permette di specializzare un modello linguistico (LLM) al latino, utilizzando un dizionario di parole latine validate per poter correggere dei testi con potenziali errori, allineandoli alle parole corrette.	OCR

Tabella 2- Sintesi Primitive e propedeuticità

La tabella seguente riporta le peculiarità di ciascuna primitiva, ordinate per 'Codifica primitiva' crescente.

Per ciascuna API è riportato:

- Codifica: l'identificativo della primitiva;
- Primitiva: il nome della primitiva. Il termine primitiva viene utilizzato come sinonimo di API;
- Servizio: servizio/modello utilizzato dalla primitiva;
- Lingue gestite dal servizio: lingue consentite dallo specifico servizio;
- Tipologia: tipologia della risorsa digitale gestita.

Codifica	Primitiva	Servizio	Lingue gestite dal servizio	Tipologia
PR 01/02	OCR / ICR	default, tesseract	Default: documenti in lingua inglese e di origine latina (italian, french, spanish)	monografia (obbligatorio)
PR 04	Speech to Text (STT)	whisper	italian, english, french, german (obbligatorio)	audio, video (obbligatorio)
PR 05	Text to Speech (TTS)	kokoro-tts	Kokoro: italian, english, french, german (obbligatorio)	monografia, periodico (obbligatorio)

PR 06	Language Translation	llama-3-3-70b, llama-3-1-8b di cinese	Llama-3-3-7B-Instruct: italian, english, french, (obbligatorio) Llama-3-1-8b di cinese: italian, english, french, german	monografia, periodico, audio (obbligatorio)
PR 07	NER per Classificazione Automatica	llama-3-3-70b	italian, english, french, german (obbligatorio)	monografia, periodico, audio, video (obbligatorio)
PR 08	Tagging Automatico dei Testi	llama-3-3-70b	italian, english, french, german (obbligatorio)	monografia, periodico, audio, video (obbligatorio)
PR 09	AI Image Enhance	default, realesrgan	N/A	monografia (obbligatorio)
PR12	OMR	audiveris	italian, english, french, german, spanish, portuguese (opzionale)	monografia (opzionale)
PR14	Ricostruzione articoli	llama_3_3_70B	N/A	N/A
PR15	Creazione Sommario (ToC) per Monografie	llama-3-3-70b	Llama-3-3-7B-Instruct: italian, english, french, german (obbligatorio)	monografia (obbligatorio)
PR 17	Generazione Audio da XML Musicali	partitura	N/A	N/A
PR18	Analisi e/o Creazione struttura fisica	llama-3-3-70b	Llama-3-3-7B-Instruct: italian, english, french, german (obbligatorio)	monografia (obbligatorio)
PR 19	Creazione abstract da Testo	llama-3-3-70b	Llama-3-3-7B-Instruct: italian, english, french, (obbligatorio)	monografia, periodico, audio (obbligatorio)
PR 21	Estrazione Entità Immagini	MISTRAL_MEDIUM_2505	italian, english, french, german (obbligatorio)	N/A
PR23	Associazione di topic con dei tag per un'immagine	llama_3_3_70B	N/A	immagine
PR25	Ricerca/ Associazione Entità a Soggettario di Firenze	llama_3_3_70B	N/A	N/A
PR32	Formattazione testo estratto (schede)	llama_3_3_70B	italian	testo

PR 35	Adattamento Latino per Epigrafi	MISTRAL_MEDIUM_2505	N/A	N/A
PR 36	Creazione abstract da immagine	mistral-medium-2505, qwen2	N/A	N/A

Tabella 3- Schema primitive e relativi servizi, lingue e tipologie

2.1. AI Image Enhance

L'AI *Image Enhance* è una tecnica di miglioramento della risoluzione delle immagini basata sull'intelligenza artificiale. Utilizzando algoritmi sofisticati, la primitiva è in grado di analizzare e ottimizzare automaticamente le immagini per migliorarne la qualità visiva. Questa tecnologia può correggere problemi come sfocatura, contrasto e luminosità, portando ad un'immagine finale più nitida, vivida e dettagliata, ottenendo risultati di alta qualità e migliorando l'esperienza visiva complessiva.

2.1.1. API da richiamare

Per eseguire l'API di *Image Enhance* sarà necessario richiamare la seguente API: </api/v1/batch/tasks/creazione-image-enhance>.

È di seguito riportato un esempio di chiamata all'API:

```
{
  "codiceOwnerSpecifico": "",
  "descrizione": "IMAGE ENHANCE",
  "publicTask": false,
  "uuidBitstream": {
    "35191B0CE8A7E258E0630204FE0A0702": {}
  },
  "crossInfo": {
    "service": null,
    "language": "italian",
    "typology": "Monografia",
    "scale": 3,
    "tile": 512,
    "tile_pad": 10,
    "noise_level": 2
  }
}
```

Di seguito sono elencati i parametri che caratterizzano questa API, come possono essere popolati e la loro eventuale obbligatorietà. Quando si utilizza la primitiva è necessario considerare alcuni parametri fondamentali:

- *service*, obbligatoria, è il servizio di Image Enhance da utilizzare per estrarre file testuali dalle immagini di input. Nella tabella che segue, le cui colonne riportano le seguenti informazioni:
 - Servizio: il nome del servizio che la primitiva può utilizzare;
 - Parametro: il parametro da passare in input;
 - Disponibile ad elaborazione backend: se il servizio è disponibile negli ambienti di test e produzione o meno,

sono elencati i servizi utilizzabili per la primitiva in oggetto.

Servizio	Parametro	Disponibile ad elaborazione backend
Default Super Resolution	Default	SI

Tabella 4 - Servizi disponibili per la Primitiva Image Enhance

- *typology*, obbligatoria, che indica la tipologia della risorsa (e.g., monografia o periodico). Attualmente, l'unico valore accettato da vocabolario è 'monografia';
- *language*, obbligatoria, che indica la lingua della risorsa/bitstream. I valori ammissibili sono in lowercase e in inglese (e.g., 'italian', 'english', 'french', 'german', ...);
- *scale*: opzionale, indica il valore di upscaling da applicare all'immagine originale quando il servizio scelto è quello di default (e.g., 3);
- *tile*: opzionale, indica i valori con cui il servizio lavora per il rendering immagine;
- *tile_pad*: opzionale, indica i valori con cui il servizio lavora per il rendering immagine;
- *noise_level*: opzionale, valore per soppressione del rumore.

2.2. OCR/ICR

Per *OCR* (Optical Character Recognition) si intende un sistema di riconoscimento di testo che consente di convertire immagini di documenti testuali in testo digitale.

L'OCR utilizza algoritmi di intelligenza artificiale e machine learning per analizzare le immagini e identificare le lettere e i simboli presenti all'interno di esse, in modo da ricostruire il testo presente nel modo più accurato possibile.

La bontà dell'output, oltre che dal modello di OCR stesso, dipende anche dalla qualità delle immagini fornite in input: è necessario fornire, ove possibile, immagini di buona qualità. Per questo, in caso di immagini non nitide o distorte, è sempre buona norma pre-processarle in modo da migliorare la qualità delle immagini stesse.

Per *ICR* (Intelligent Character Recognition) si intende un tipo di tecnologia di riconoscimento ottico del carattere (OCR) che è in grado di processare testi scritti a mano. Questo software estende la proprietà di riconoscimento dei caratteri standard dell'OCR e consente di utilizzare diversi stili e caratteri di scrittura.

Essendo un'estensione dell'OCR, valgono le stesse considerazioni fatte in precedenza per quanto riguarda la qualità delle immagini in input.

Attualmente, l'OCR di default può processare solo immagini di documenti in lingua inglese e di origine latina (italiano, latino, francese e spagnolo). I documenti di lingua non inclusa in questo elenco non potranno essere processati da questo servizio.

L'OCR di default può esaminare testi dattiloscritti o stampati; testi manoscritti, o con stili calligrafici artistici: non verranno riconosciuti correttamente. Inoltre, tale servizio OCR attualmente non è in grado di rilevare il numero di pagina dell'immagine processata.

Infine, la primitiva OCR di default prevede due tipologie di miglioramento della risoluzione delle immagini. La prima sfrutta una rete neurale per rimuovere artefatti JPEG (che dunque è possibile attivare esclusivamente se è presente questo formato), cioè eliminare tutte le aberrazioni dovute alla compressione del formato. In questo caso l'input all'interno della pipeline è un'immagine originale e l'output è l'immagine ripulita.

La seconda utilizza un modello di super risoluzione dell'immagine, nel caso in cui quest'ultima sia stata scansionata ad una bassa risoluzione. In questo caso l'input è l'immagine originale a bassa risoluzione e l'output è l'immagine super risolta.

2.2.1. API da richiamare

Per eseguire un *OCR/ICR* sarà necessario richiamare la seguente API:

</api/v1/batch/tasks/creazione-ocr>

È di seguito riportato un esempio di chiamata all'API:

```
{
  "codiceOwnerSpecifico": "",
  "descrizione": "OCR",
  "publicTask": false,
  "uuidRisorseDigitali": {
    "858DB94E1DBE4F2BBCAE0459A28AC4A6": {}
  },
  "crossInfo": {
    "service": "document_intelligence",
    "language": "italian",
    "typology": "Immagine"
  }
}
```

Di seguito sono elencati i parametri che caratterizzano questa API, come possono essere popolati e la loro eventuale obbligatorietà. Quando si utilizza la primitiva è necessario considerare alcuni parametri fondamentali:

- *service*, per estrarre file testuale dalle immagini di input. In Tabella 5, le cui colonne riportano le seguenti informazioni:
 - Servizio: il nome del servizio che la primitiva può utilizzare;
 - Parametro: il parametro da passare in input;
 - Disponibile ad elaborazione backend: se il servizio è disponibile negli ambienti o meno,

sono elencati i servizi utilizzabili per la primitiva in oggetto.

Servizio	Parametro	Disponibile ad elaborazione backend
Default	default	SI
Document Intelligence	internal	SI

Tabella 5 - Servizi disponibili per la Primitiva OCR e la Primitiva ICR

- *typology*, obbligatoria, che indica la tipologia della risorsa (e.g, monografia o periodico);
- *language*, obbligatoria, che indica la lingua della risorsa/bitstream. I valori ammissibili sono in lowercase e in inglese (e.g., 'italian', 'english', 'french', 'german', etc.).

2.3. Speech To Text (STT)

Il servizio di *Speech-to-Text (STT)* consente di convertire file audio in testo digitale effettuando la trascrizione del contenuto audio. Consente inoltre di estrarre il contenuto testuale anche da file video.

2.3.1. Api da richiamare

Per eseguire l'API di *Speech To Text* sarà necessario richiamare la seguente API:
</api/v1/batch/tasks/creazione-cpa-speech-to-text>

È di seguito riportato un esempio di chiamata all'API:

```
{
  "codiceOwnerSpecifico": "",
  "descrizione": "SPEECH TO TEXT",
  "publicTask": false,
  "uuidRisorseDigitali": {
    "7FA6E37AEC3941B4B5A00CAA3ACB07": {}
  },
  "crossInfo": {
    "typology": "video",
    "service": "whisper_gpu",
    "language": "italian",
    "diarization": false,
    "size": "large"
  }
}
```

Di seguito sono elencati i parametri che caratterizzano questa API, come possono essere popolati e la loro eventuale obbligatorietà. Quando si utilizza la primitiva è necessario considerare alcuni parametri fondamentali.:

- *service*, obbligatoria, è il servizio STT da utilizzare per estrarre file testuale da file audio/video di input.

Nella tabella sottostante, le cui colonne riportano le seguenti informazioni:

- Servizio: il nome del servizio che la primitiva può utilizzare;
- Parametro: il parametro da passare in input;
- Disponibile ad elaborazione backend: se il servizio è disponibile negli ambienti o meno,

sono elencati i servizi utilizzabili per la primitiva in oggetto.

Servizio	Parametro	Disponibile ad elaborazione backend
----------	-----------	-------------------------------------

Whisper	whisper	SI
---------	---------	----

Tabella 6 - Servizi disponibili per la Primitiva Speech To Text

- *size*, obbligatoria, è la dimensione del modello whisper da utilizzare. Attualmente il modello Whisper esegue direttamente sul Pod Openshift su cui è deployata la primitiva. Di conseguenza, la taglia del modello da eseguire deve tenere in considerazione la capacità del Pod in ciascun ambiente di elaborazione;
- *diarization*, facoltativa, è un flag booleano che identifica la volontà o meno di estrarre la diarizzazione del file di input. La diarizzazione corrisponde ad identificare i parlanti ed etichettare il testo estratto da whisper in base al parlante;
- *typology*, obbligatoria, che indica la tipologia della risorsa;
- *language*, obbligatoria, che indica la lingua della risorsa/bitstream.

2.4. Ricostruzione articoli

La primitiva *Ricostruzione Articoli* ha lo scopo di ricostruire la struttura degli articoli partendo dall'output di un modulo OCR/ICR. A tal fine, utilizza le coordinate del testo estratto per mappare ogni titolo sul suo corpo testuale corrispondente.

2.4.1. API da richiamare

Per eseguire l'API di *Ricostruzione articoli* sarà necessario richiamare la seguente API:
</api/v1/batch/tasks/creazione-cpa-ricostruzione-articoli>

È di seguito riportato un esempio di chiamata all'API:

```
{
  "codiceOwnerSpecifico": "",
  "descrizione": "RICOSTRUZIONE ARTICOLI",
  "publicTask": false,
  "uuidRisorseDigitali": {
    "E2F3AC2732A842D986968C8C8BEDAEB6": {}
  },
  "crossInfo": {
    "typology": "TESTO",
    "service": "llama_3_3_70B",
    "language": "italian",
    "heuristics": false,
    "titlePosition": "mix"
  }
}
```

}

Di seguito sono elencati i parametri che caratterizzano questa API, come possono essere popolati e la loro eventuale obbligatorietà. Quando si utilizza la primitiva è necessario considerare alcuni parametri fondamentali.:

- *typology*, di tipo stringa e obbligatorio, indica la tipologia della risorsa (e.g., immagine, testo);
- *service*, obbligatorio. Identificativo del servizio backend che viene istanziato e supportato dalla primitiva. Questo influenza la qualità dell'operazione ed è stato selezionato dal gruppo AI di riferimento. Come modello si consiglia l'utilizzo di "llama_3_3_70B";
- *language*, di tipo stringa e obbligatorio, indica la lingua della risorsa, da passare in lowercase in inglese (e.g., italian, english...);
- *heuristics*, facoltativo. Parametro per l'utilizzo delle euristiche, che permettono in alcuni casi di estrarre maggiori informazioni anche a fronte di input con bassi valori di accuratezza. (output del modello di ocr) Il parametro viene impostato di default a False;
- *titlePosition*, facoltativo. Parametro utilizzato dalle euristiche come indicazione per l'allineamento dei titoli, può essere "left", "center", "right", "mix". Di default viene impostato a "mix".

2.5. NER per Classificazione Automatica

Per *NER* (Named Entity Recognition) si intende una tecnica di elaborazione del linguaggio naturale (NLP) che consiste nel riconoscere, identificare e classificare specifiche entità all'interno di un testo non strutturato, come nomi di persone, luoghi, organizzazioni, date, valute e altro ancora.

L'obiettivo principale della primitiva *NER per Classificazione Automatica* è estrarre informazioni rilevanti dal testo e assegnare a ciascuna entità una categoria specifica.

Il NER utilizza algoritmi di apprendimento automatico, modelli statistici e regole linguistiche per identificare le entità e assegnare loro un'etichetta. Può coinvolgere diverse fasi, come la tokenizzazione del testo, l'individuazione delle parti del discorso e l'applicazione di modelli di classificazione per l'etichettatura delle entità.

2.5.1. API da richiamare

Per eseguire la *NER per Classificazione Automatica* sarà necessario richiamare la seguente API:
</api/v1/batch/tasks/creazione-ner>

È di seguito riportato un esempio di chiamata all'API:

```
{
  "codiceOwnerSpecifico": "",
  "descrizione": "NER",
  "publicTask": false,
  "uuidRisorseDigitali": {
    "179E0250D6754751B98AF766D57723C4": {}
  },
  "crossInfo": {
    "service": null,
    "language": "italian",
    "typology": "monografia"
  }
}
```

Di seguito sono elencati i parametri che caratterizzano questa API, come possono essere popolati e la loro eventuale obbligatorietà. Quando si utilizza la primitiva è necessario considerare alcuni parametri fondamentali:

- *service*, obbligatoria, è il modello LLM da utilizzare per estrarre le entità nominali dal testo di input. In Tabella 7, le cui colonne riportano le seguenti informazioni:
 - Servizio: il nome del servizio che la primitiva può utilizzare;
 - Parametro: il parametro da passare in input;
 - Disponibile ad elaborazione backend: se il servizio è disponibile negli ambienti o meno;
 - Parameter Tuning: i parametri di tuning consentiti dal servizio,
 sono elencati i servizi utilizzabili per la primitiva in oggetto.

Servizio	Parametro	Disponibile ad elaborazione backend	Parameter Tuning
Llama-3-1-7B-Instruct	llama-3-1-70b	NO	Temperature = 0, max_tokens = 800
Llama-3-3-7B-Instruct	llama-3-3-70b	SI	Temperature = 0, max_tokens = 800
Gpt-4o	gpt-4o	NO	Temperature = 0, max_tokens = 800

Gpt-4o-mini	gpt-4o-mini	NO	Temperature = 0, max_tokens = 800
-------------	-------------	----	--------------------------------------

Tabella 7 - Servizi disponibili per la Primitiva NER per Classificazione Automatica

- *language*, obbligatoria, che indica la lingua della risorsa/bitstream. I valori ammissibili sono in lowercase e in inglese (e.g., 'italian', 'english', 'french', 'german', ...);
- *typology*, obbligatoria, che indica la tipologia della risorsa (e.g, monografia_stampa o periodico_stampa).

2.6. Tagging Automatico dei Testi

Nel contesto I.PaC, la primitiva di *Tagging Automatico dei Testi* esegue l'estrazione del tema o dei temi principali trattati nel testo (che possono essere termini astratti, provenienti dal Nuovo Soggettario di Firenze, o entità nominate, come agenti, luoghi, beni culturali, etc).

La primitiva *Tagging Automatico dei Testi* dipende dall'output del servizio *NER per Classificazione Automatica* e della primitiva *OCR*.

2.6.1. API da richiamare

Per eseguire il *Tagging automatico dei Testi* sarà necessario richiamare la seguente API:

</api/v1/batch/tasks/creazione-tagging>

È di seguito riportato un esempio di chiamata all'API:

```
{
  "codiceOwnerSpecifico": "",
  "descrizione": "TAGGING",
  "publicTask": false,
  "uuidRisorseDigitali": {
    "5D529DF5CAAD48C290EFC31942C5E9F4": {}
  },
  "crossInfo": {
    "service": null,
    "language": "italian",
    "typology": "monografia"
  }
}
```

Di seguito sono elencati i parametri che caratterizzano questa API, come possono essere popolati e la loro eventuale obbligatorietà. Quando si utilizza la primitiva è necessario considerare alcuni parametri fondamentali:

- *service*, obbligatoria, indica il modello LLM da utilizzare per estrarre i temi astratti dal testo di input. In Tabella 8 le cui colonne riportano le seguenti informazioni:
 - Servizio: il nome del servizio che la primitiva può utilizzare;
 - Parametro: il parametro da passare in input;
 - Disponibile ad elaborazione backend: se il servizio è disponibile negli ambienti o meno;
 - Parameter Tuning: i parametri di tuning consentiti dal servizio,
 sono elencati i servizi utilizzabili per la primitiva in oggetto.

Servizio	Parametro	Disponibile ad elaborazione backend	Parameter Tuning
Llama-3-1-7B-Instruct	llama-3-1-70b	NO	Temperature = 0, max_tokens = 800
Llama-3-3-7B-Instruct	llama-3-3-70b	SI	Temperature = 0, max_tokens = 800
Gpt-4o	gpt-4o	NO	Temperature = 0, max_tokens = 800
Gpt-4o-mini	gpt-4o-mini	NO	Temperature = 0, max_tokens = 800

Tabella 8 - Servizi disponibili per la Primitiva Tagging Automatico dei Testi

- *language*, obbligatoria, che indica la lingua della risorsa/bitstream. I valori ammissibili sono in lowercase in inglese (e.g., 'italian', 'english', 'french', 'german', ...);
- *typology*, obbligatoria, che indica la tipologia della risorsa (e.g, monografia_stampa o periodico_stampa).

2.7. Estrazione Entità Immagini

La primitiva *Estrazione Entità Immagini* è il processo che permette di riconoscere e isolare delle entità presenti all'interno di un'immagine. Con entità si fa riferimento a degli elementi che possono variare in tipologia (es. strutture architettoniche, elementi decorativi, utensili, ecc.) che vengono definiti di valore per il caso d'uso di riferimento.

Le entità ottenute da questa primitiva possono essere utilizzate per avere una descrizione strutturata della scena rappresentata dall'immagine.

2.7.1. API da richiamare

Per eseguire l'API di *Estrazione Entità Immagini* sarà necessario richiamare la seguente API:
</api/v1/batch/tasks/creazione-estrazione-entita>

È di seguito riportato un esempio di chiamata all'API:

```
{
  "codiceOwnerSpecifico": "",
  "descrizione": "ESTRAZIONE ENTITA",
  "publicTask": false,
  "uuidBitstream": {
    "35191B0CE8A7E258E0630204FE0A0702": {}
  },
  "crossInfo": {
    "service": "MISTRAL_MEDIUM_2505",
    "language": "italian",
    "typology": "monografia"
  }
}
```

Di seguito sono elencati i parametri che caratterizzano questa API, come possono essere popolati e la loro eventuale obbligatorietà. Quando si utilizza la primitiva è necessario considerare alcuni parametri fondamentali:

- *language*, di tipo stringa e obbligatorio, indica la lingua della risorsa, da passare in lowercase in inglese (e.g., italian, english...);
- *typology*, di tipo stringa e obbligatorio, indica la tipologia della risorsa (e.g., immagine, testo);
- *service*, di tipo stringa e obbligatorio, indica il servizio di estrazione da utilizzare (e.g. "MISTRAL_MEDIUM_2505");
- *temperature*, di tipo float e obbligatorio, indica il parametro di temperatura da fornire al modello (e.g. 0.1).

2.8. Anagrafiche

Lo scopo della primitiva *Anagrafiche*, all'interno dell'ecosistema I.PaC, è arricchire il grafo della conoscenza estraendo informazioni anagrafiche dalle entità nominali presenti in un testo.

La primitiva *Anagrafiche* può essere un'estensione del modulo NER, in tal caso l'output viene arricchito includendo, oltre alle entità e categorie standard della NER, anche le informazioni anagrafiche estratte per ciascuna entità.

2.8.1. API da richiamare

Per eseguire l'API di *Anagrafiche* sarà necessario richiamare la seguente API:
</api/v1/batch/tasks/creazione-anagrafiche>

È di seguito riportato un esempio di chiamata all'API:

```
{
  "codiceOwnerSpecifico": "",
  "descrizione": "ANAGRAFICHE",
  "publicTask": false,
  "uuidRisorseDigitali": {
    "3BA33A9AC6434956BAA08026D06442B1": {}
  },
  "crossInfo": {
    "service": "llama-3-3-70b",
    "language": "italian",
    "typology": "monografia"
  }
}
```

Di seguito sono elencati i parametri che caratterizzano questa API, come possono essere popolati e la loro eventuale obbligatorietà. Quando si utilizza la primitiva è necessario considerare alcuni parametri fondamentali:

- *service*, obbligatoria, è il modello LLM da utilizzare per estrarre le entità nominali dal testo di input. Nella tabella sottostante, le cui colonne riportano le seguenti informazioni:
 - Servizio: il nome del servizio che la primitiva può utilizzare;
 - Parametro: il parametro da passare in input;
 - Disponibile ad elaborazione backend: se il servizio è disponibile negli ambienti o meno;
 - Parameter Tuning: i parametri di tuning consentiti dal servizio,
 sono elencati i servizi utilizzabili per la primitiva in oggetto.

Servizio	Parametro	Disponibile ad elaborazione backend	Parameter Tuning
----------	-----------	-------------------------------------	------------------

Llama-3-1-7B-Instruct	llama-3-1-70b	NO	Temperatura = 0
Llama-3-3-7B-Instruct	llama-3-3-70b	SI	Temperatura = 0
Gpt-4o	gpt-4o	NO	Temperatura = 0
Gpt-4o-mini	gpt-4o-mini	NO	Temperatura = 0

Tabella 9 - Servizi disponibili per la Primitiva Anagrafiche

- *language*, obbligatoria, che indica la lingua della risorsa/bitstream. I valori ammissibili sono in lowercase e in inglese (e.g., 'italian', 'english', 'french', 'german', ...);
- *typology*, obbligatoria, che indica la tipologia della risorsa (e.g, monografia o periodico). Attualmente, l'unico valore accettato da vocabolario è 'monografia';

2.9. Ricerca/Associazione Entità a Soggettario di Firenze

La primitiva *Ricerca/Associazione Entità a Soggettario di Firenze* offre la possibilità di ricavare i termini all'interno del Nuovo Soggettario di Firenze che meglio corrispondono ai termini forniti. In particolare, permette di identificare le entità del Soggettario che corrispondono alle entità estratte dalla primitiva *Estrazione Entità Immagini*.

2.9.1. API da richiamare

Per eseguire l'API di *Associazione Entità a Soggettario di Firenze* sarà necessario richiamare la seguente API: [/api/v1/batch/tasks/creazione-cpa-soggettario](#)

È di seguito riportato un esempio di chiamata all'API:

```
{
  "codiceOwnerSpecifico": "",
  "descrizione": "CPA SOGGETTARIO",
  "uuidBitstream": {
    "386641BB1F0239ABE0634A0DFE0AA530": {}
  },
  "crossInfo": {
    "similarityThreshold": 70,
    "service": "LLAMA_3_3_70B",
    "temperature": 0
  }
}
```

Di seguito sono elencati i parametri che caratterizzano questa API, come possono essere popolati e la loro eventuale obbligatorietà. Quando si utilizza la primitiva è necessario considerare alcuni parametri fondamentali:

- *similarityThreshold*, obbligatorio, indica il valore limite di similarità sotto al quale non prendere in considerazione l'associazione a termini all'interno del Nuovo Soggettario di Firenze (al fine di evitare sostituzioni con termini esageratamente distanti);
- *service*, di tipo stringa e obbligatorio, indica il servizio di filtro da utilizzare (e.g. "llama_3_3_70B");
- *temperature*, di tipo float e obbligatorio, indica il parametro di temperatura da fornire al modello (e.g. 0.1).

2.10. Associazione di topic con dei tag per un'immagine

Con tagging delle immagini si fa riferimento all'operazione di trovare le coordinate delle entità all'interno dell'immagine, entrambe passate come input alla primitiva.

Le coordinate nell'immagine ottenute da questa primitiva possono essere utilizzate per visualizzare le entità all'interno dell'immagine, o per estrarne i relativi box per processamenti futuri.

2.10.1. API da richiamare

Per eseguire l'API di *Associazione di topic con dei tag per un'immagine* sarà necessario richiamare la seguente API:

</api/v1/batch/tasks/creazione-cpa-tagging-entita-immagini>

È di seguito riportato un esempio di chiamata all'API:

```
{
  "codiceOwnerSpecifico": "",
  "descrizione": "TAGGING ENTITA IMMAGINI",
  "publicTask": false,
  "uuidRisorseDigitali": {
    "E8489C6C8A5E40CCB9295BAB661DDCA9": {}
  },
  "crossInfo": {
    "typology": "immagine",
    "service": "llama_3_3_70B",
    "language": "italiano"
  }
}
```

Di seguito sono elencati i parametri che caratterizzano questa API, come possono essere popolati e la loro eventuale obbligatorietà. Quando si utilizza la primitiva è necessario considerare alcuni parametri fondamentali:

- *language*, di tipo stringa e obbligatorio, indica la lingua della risorsa, da passare in lowercase in inglese (e.g., italian, english, ...);
- *typology*, di tipo stringa e obbligatorio, indica la tipologia della risorsa (e.g., immagine, testo);
- *service*, di tipo stringa e obbligatorio, indica il servizio di formattazione da utilizzare (e.g. "llama_3_2_70B").

2.11. Analisi e/o Creazione struttura fisica risorsa

La primitiva *Analisi e/o Creazione Struttura Fisica* consiste nell'estrazione della sequenza numerica delle pagine della risorsa digitale in input, tramite l'utilizzo dell'output del servizio *OCR* (o primitiva *ICR*) con la finalità di estrarre i numeri di pagina dai media che compongono una risorsa.

2.11.1. API da richiamare

Per eseguire l'API di *Analisi e/o Creazione Struttura Fisica* sarà necessario richiamare la seguente API:
</api/v1/batch/tasks/creazione-cpa-struttura-fisica>

È di seguito riportato un esempio di chiamata all'API:

```
{
  "codiceOwnerSpecifico": "",
  "descrizione": "STRUTTURA FISICA",
  "publicTask": false,
  "uuidRisorseDigitali": {
    "179E0250D6754751B98AF766D57723C4": {
      "uuidStrutturaFisica": "30151CA2316DE471E0634A0DFE0A545D"
    }
  },
  "crossInfo": {
    "service": "llama-3-1-70b",
    "language": "italian",
    "typology": "monografia"
  }
}
```

Di seguito sono elencati i parametri che caratterizzano questa API, come possono essere popolati e la loro eventuale obbligatorietà. Quando si utilizza la primitiva è necessario considerare alcuni parametri fondamentali:

- *uuid_strutturafisica*: la struttura fisica da processare e che permette di recuperare la lista di *uuid_oggettigitali* che la primitiva dovrà analizzare ed ordinare, la loro *posizioneGerarchica* presente su GPA ed infine bitstream derivati dalla primitiva OCR di ciascun oggetto digitale;
- *service*, obbligatoria, indica il modello LLM da utilizzare per eseguire la traduzione del testo in input. Nella tabella sottostante, le cui colonne riportano le seguenti informazioni:
 - Servizio: il nome del servizio che la primitiva può utilizzare;
 - Parametro: il parametro da passare in input;
 - Disponibile ad elaborazione backend: se il servizio è disponibile negli ambienti o meno;
 - Parameter Tuning: i parametri di tuning consentiti dal servizio,

sono riportati i servizi utilizzabili dalla primitiva in oggetto.

Servizio	Parametro	Disponibile ad elaborazione backend	Parameter Tuning
Llama-3-1-7B-Instruct	llama-3-1-70b	NO	Temperature = 0
Llama-3-3-7B-Instruct	llama-3-3-70b	SI	Temperature = 0
Gpt-4o-mini	gpt-4o-mini	NO	Temperature = 0

Tabella 10 - Servizi disponibili per la Primitiva Analisi e/o Creazione Struttura Fisica

Il parametro della temperatura è quello che definisce quanto è deterministico un risultato di un modello LLM. Il valore della temperatura è compreso tra 0 e 1. Più la temperatura è bassa, più il risultato sarà deterministico;

- *language*, obbligatoria, indica la lingua della risorsa. I valori ammissibili sono in lowercase in inglese (e.g., 'italian', 'english', 'french', 'german', ...);
- *typology*, obbligatoria, indica la tipologia della risorsa (e.g, monografia o periodico). Attualmente, l'unico valore accettato da vocabolario è 'monografia'.

2.12. Creazione Sommario (ToC) per Monografie

La *Primitiva Creazione Sommario (ToC) per Monografie* consiste nell'identificazione della Tavola dei Contenuti (ToC), cioè delle sezioni logiche di una monografia (capitoli, prefazione, note, ...) ed i relativi media che le compongono.

La primitiva analizza file testuali relativi ad una monografia (ottenuti, ad esempio, da un servizio di OCR) e, utilizzando un modello LLM, identifica i nomi delle sezioni. Inoltre, vengono identificati quali sono i media che compongono una sezione, in modo tale da avere l'associazione capitolo-pagine del capitolo.

La primitiva di *Creazione Sommario per Monografie* prevede l'integrazione con il servizio *Analisi e/o Creazione Struttura Fisica*, per incorporare anche le informazioni relative al numero di pagina dei media.

2.12.1. API da richiamare

Per eseguire l'API di *Creazione Sommario (ToC) per Monografie* sarà necessario richiamare la seguente API: </api/v1/batch/tasks/creazione-cpa-struttura-logica>

È di seguito riportato un esempio di chiamata all'API:

```
{
  "codiceOwnerSpecifico": "",
  "descrizione": "STRUTTURA LOGICA",
  "publicTask": false,
  "uuidRisorseDigitali": {
    "179E0250D6754751B98AF766D57723C4": {
      "uuidStrutturaFisica": "30151CA2316DE471E0634A0DFE0A545D"
    }
  },
  "crossInfo": {
    "service": null,
    "language": "italian",
    "typology": "monografia"
  }
}
```

Di seguito sono elencati i parametri che caratterizzano questa API, come possono essere popolati e la loro eventuale obbligatorietà. Quando si utilizza la primitiva è necessario considerare alcuni parametri fondamentali:

- *uuid_strutturafisica*, obbligatoria, che serve per leggere il bitstream associato alla risorsa digitale che ha come campo *tipoDerivazione* = 'CPA_STRUTTURA_FISICA'. In questo modo la

primitiva elabora i bitstream nell'ordine della struttura fisica in input, derivante dalla primitiva di struttura fisica.

- *service*, obbligatoria, è il modello LLM da utilizzare per estrarre le entità nominali dal testo di input. Nella tabella sottostante, le cui colonne riportano le seguenti informazioni:
 - Servizio: il nome del servizio che la primitiva può utilizzare;
 - Parametro: il parametro da passare in input;
 - Disponibile ad elaborazione backend: se il servizio è disponibile negli ambienti o meno;
 - Parameter Tuning: i parametri di tuning consentiti dal servizio, sono riportati i servizi utilizzabili dalla primitiva in oggetto.

Servizio	Parametro	Disponibile ad elaborazione backend	Parameter Tuning
Llama-3-1-7B-Instruct	llama-3-1-70b	NO	Temperature = 0
Llama-3-3-7B-Instruct	llama-3-3-70b	SI	Temperature = 0
Gpt-4o-mini	gpt-4o-mini	NO	Temperature = 0

Tabella 11 - Servizi disponibili per la Primitiva Creazione Sommario (ToC) per Monografie

- *language*, obbligatoria, che indica la lingua della risorsa/bitstream. I valori ammissibili sono in lowercase e in inglese (e.g., 'italian', 'english', 'french', 'german', ...);
- *typology*, obbligatoria, che indica la tipologia della risorsa (e.g, monografia o periodico). Attualmente, l'unico valore accettato da vocabolario è 'monografia'.

2.13. Creazione abstract da testo

La primitiva *Creazione abstract da testo* consente di generare un abstract contenente le informazioni più rilevanti e significative dal testo originale e presentarle in modo sintetico, tramite l'utilizzo di modelli LLM.

Nello specifico, questa primitiva consente di generare abstract per le seguenti tipologie di risorse:

- monografie;
- periodici;
- transcript di audio.

Nel caso delle monografie e dei transcript delle risorse audio, verrà generato un abstract complessivo dell'intera risorsa, nel caso dei periodici viene invece restituito un abstract per ciascun articolo presente nel periodico. Viene richiesto che ciascun abstract inizi con una descrizione/inquadratura iniziale della risorsa o articolo a partire dalle informazioni derivate dai metadati associata risorsa o articolo, seguita dalla sintesi del testo. La lingua in cui viene generato l'abstract è la lingua originale della risorsa, che viene indicata al servizio in fase di chiamata.

La lunghezza dell'abstract atteso varia a seconda della tipologia di risorsa.

Nel caso delle **monografie**, la lunghezza dell'abstract viene definita sulla base del numero di pagine (approssimato con il numero di bitstream) della risorsa digitale in input.

Per gli articoli di **periodici**, è previsto che venga prodotto un abstract per ciascun articolo presente, ammesso che questo non risulti troppo corto.

Per gli **audio**, la lunghezza viene definita sulla base del numero di parole del transcript in input.

Le soglie identificate per le varie tipologie di risorsa sono indicate in Tabella 12.

Tipo risorsa	Lunghezza opera	Lunghezza abstract
Monografia	<150 bitstream	100-150 parole
	150-500 bitstream	200-250 parole
	>500 bitstream	250-300 parole
Periodico	Articolo corto (<100 parole)	No abstract
	Articolo di media lunghezza (100-500 parole)	0.3*numero parole input
	Articolo lungo (>500 parole)	150 parole
Audio	<200 parole	50-70 parole
	200-700 parole	70-100 parole
	> 700 parole	150-200 parole

Tabella 12 - Soglie utili a definire la lunghezza dell'abstract

2.13.1. API da richiamare

Per eseguire l'API di *Creazione Abstract da testo* sarà necessario richiamare la seguente API:

</api/v1/batch/tasks/creazione-abstract>

È di seguito riportato un esempio di chiamata all'API:

```
{
  "codiceOwnerSpecifico": "",
  "descrizione": "ABSTRACT",
  "publicTask": false,
  "uuidRisorseDigitali": {
    "179E0250D6754751B98AF766D57723C4": {}
  }
}
```

```

},
"crossInfo": {
  "language": "italian",
  "typology": "monografia"
}
}

```

Di seguito sono elencati i parametri che caratterizzano questa API, come possono essere popolati e la loro eventuale obbligatorietà. Quando si utilizza la primitiva è necessario considerare alcuni parametri fondamentali:

- *service*, obbligatoria, indica il modello LLM da utilizzare per eseguire la traduzione del testo in input. Nella tabella sottostante, le cui colonne riportano le seguenti informazioni:
 - Servizio: il nome del servizio che la primitiva può utilizzare;
 - Parametro: il parametro da passare in input;
 - Disponibile ad elaborazione backend: se il servizio è disponibile negli ambienti o meno;
 - Parameter Tuning: i parametri di tuning consentiti dal servizio,
 sono elencati i servizi utilizzabili dalla primitiva in oggetto.

Servizio	Parametro	Disponibile ad elaborazione backend	Parameter Tuning
Llama-3-1-7B-Instruct	llama-3-1-70b	NO	Temperatura=0
Llama-3-3-7B-Instruct	llama-3-3-70b	SI	Temperatura=0
Gpt-4o-mini	gpt-4o-mini	NO	Temperatura=0

Tabella 13 - Servizi disponibili per la Primitiva Abstract da Testo

Il parametro della temperatura è quello che definisce quanto è deterministico un risultato di un modello LLM. Il valore della temperatura è compreso tra 0 e 1. Più la temperatura è bassa, più il risultato sarà deterministico;

- *language*, obbligatoria, indica la lingua della risorsa/bitstream. I valori ammissibili sono in lowercase in inglese (e.g., 'italian', 'english', 'french', 'german', ...);
- *typology*, obbligatoria, indica la tipologia della risorsa (e.g, monografia, periodico o audio).

2.14. Creazione abstract da immagine

La primitiva *Creazione abstract da immagine* consente di generare un abstract a partire da un'immagine, attraverso l'utilizzo di modelli generativi multimodali. In particolare, è inoltre possibile generare un abstract che sia comprensivo di tutte le entità ottenute dalla primitiva *Estrazione Entità Immagini* o senza l'utilizzo di queste entità, su richiesta dell'utente.

2.14.1. API da richiamare

Per eseguire l'API di *Creazione abstract da immagine* sarà necessario richiamare la seguente API:

</api/v1/batch/tasks/creazione-abstract-immagini>

È di seguito riportato un esempio di chiamata all'API:

```
{
  "codiceOwnerSpecifico": "",
  "descrizione": "ABSTRACT IMMAGINE",
  "publicTask": false,
  "uuidBitstream": {
    "3AF9D5DBEC899E25E0630204FE0A3A9F": {}
  },
  "crossInfo": {
    "service": "mistral-medium-2505",
    "language": "italian",
    "typology": "Immagine"
  }
}
```

Di seguito sono elencati i parametri che caratterizzano questa API, come possono essere popolati e la loro eventuale obbligatorietà. Quando si utilizza la primitiva è necessario considerare alcuni parametri fondamentali:

- *service*, obbligatoria, indica il modello multimodale da utilizzare per eseguire la traduzione del testo in input. Nella tabella sottostante, le cui colonne riportano le seguenti informazioni:
 - Servizio: il nome del servizio che la primitiva può utilizzare;
 - Parametro: il parametro da passare in input;
 - Disponibile ad elaborazione backend: se il servizio è disponibile negli ambienti o meno;
 - Parameter Tuning: i parametri di tuning consentiti dal servizio,

sono elencati i servizi utilizzabili dalla primitiva in oggetto.

Servizio	Parametro	Disponibile ad elaborazione backend	Parameter Tuning
Llama-3-2-90B	llama-3-2-90b	NO	Temperatura=0

mistral-medium-2505	mistral-medium-2505	SI	Temperatura=0
---------------------	---------------------	----	---------------

Tabella 14 - Servizi disponibili per la Primitiva Abstract da Immagini

Il parametro della temperatura è quello che definisce quanto è deterministico un risultato di un modello LLM. Il valore della temperatura è compreso tra 0 e 1. Più la temperatura è bassa, più il risultato sarà deterministico.

- *language*, obbligatoria, indica la lingua della risorsa/bitstream. I valori ammissibili sono in lowercase in inglese (e.g., 'italian', 'english', 'french', 'german', ...);
- *typology*, obbligatoria, indica la tipologia della risorsa (e.g, immagine).

2.15. Language Translation

Per *Language Translation* si intende il processo di conversione di un testo scritto da una lingua in un'altra lingua, utilizzando algoritmi e software specializzati. La primitiva di *Language Translation* traduce il testo in input da lingua straniera in lingua italiana utilizzando modelli di LLM.

2.15.1. API da richiamare

Per eseguire una *Language Translation* in italiano sarà necessario richiamare la seguente API: </api/v1/batch/tasks/creazione-translation>

È di seguito riportato un esempio di chiamata all'API:

```
{
  "codiceOwnerSpecifico": "",
  "descrizione": "Translation",
  "publicTask": false,
  "nomeZip": "",
  "uuidRisorseDigitali": {
    "179E0250D6754751B98AF766D57723C4": {
      }
    },
  "crossInfo": {
    "typology": "Monografia",
    "service": null,
    "language": "italian"
  }
}
```

Di seguito sono elencati i parametri che caratterizzano questa API, come possono essere popolati e la loro eventuale obbligatorietà. Quando si utilizza la primitiva è necessario considerare alcuni parametri fondamentali:

- **service**, obbligatoria, indica il modello LLM da utilizzare per eseguire la traduzione del testo in input. Nella tabella sottostante, le cui colonne riportano le seguenti informazioni:
 - Servizio: il nome del servizio che la primitiva può utilizzare;
 - Parametro: il parametro da passare in input;
 - Disponibile ad elaborazione backend: se il servizio è disponibile negli ambienti o meno;
 - Parameter Tuning: i parametri di tuning consentiti dal servizio,
 sono elencati i servizi utilizzabili dalla primitiva in oggetto.

Servizio	Parametro	Disponibile ad elaborazione backend	Parameter Tuning
Llama-3-1-7B-Instruct	llama-3-1-70b	SI	Temperatura=0
Llama-3-3-7B-Instruct	llama-3-3-70b	NO	Temperatura=0
Gpt-4o-mini	gpt-4o-mini	NO	Temperatura=0

Tabella 15 - Servizi disponibili per la Primitiva Language Translation

- **language**, obbligatoria, lingua della risorsa/bitstream. I valori ammissibili sono in lowercase e in inglese (e.g., 'italian', 'english', 'french', 'german', ...) i valori supportati sono english, german, french, portoghese, hindi, spanish e thai;
- **typology**, obbligatoria, che indica la tipologia della risorsa (e.g, monografia o periodico). Attualmente, l'unico valore accettato dal vocabolario è 'monografia'.

2.16. Text To Speech (TTS)

Nel contesto I.PaC, con primitiva *Text to Speech (TTS)* si intende un sistema di conversione del testo in audio, che consente di trasformare il testo digitale sfruttando le informazioni sulla lingua contenuti nei metadati.

La primitiva *Text to Speech (TTS)*, di cui si avvale la primitiva in oggetto, utilizza algoritmi di sintesi vocale per generare un suono che imita la voce umana, leggendo il testo digitale.

2.16.1. API da richiamare

Per eseguire il *Text to Speech (TTS)* sarà necessario richiamare la seguente API:

</api/v1/batch/tasks/creazione-tts>

È di seguito riportato un esempio di chiamata all'API:

```
{
  "codiceOwnerSpecifico": "",
  "descrizione": "TTS",
  "publicTask": false,
  "uuidRisorseDigitali": {
    "5D529DF5CAAD48C290EFC31942C5E9F4": {}
  },
  "crossInfo": {
    "typology": "Monografia",
    "service": null,
    "language": "italian"
  }
}
```

Di seguito sono elencati i parametri che caratterizzano questa API, come possono essere popolati e la loro eventuale obbligatorietà. Quando si utilizza la primitiva è necessario considerare alcuni parametri fondamentali:

- *service*, obbligatorio, indica il modello LLM da utilizzare per l'elaborazione. In Tabella 16, le cui colonne riportano le seguenti informazioni:
 - Servizio: il nome del servizio che la primitiva può utilizzare;
 - Parametro: il parametro da passare in input;
 - Disponibile ad elaborazione backend: se il servizio è disponibile negli ambienti o meno;
 - Open Source: se il servizio è open source o meno,

sono elencati i servizi utilizzabili dalla primitiva in oggetto.

Servizio	Parametro	Disponibile ad elaborazione backend	Open source
Azure Neural TTS	azure-tts	NO	NO
Kokoro TTS	kokoro-tts	SI	SI

Tabella 16 - Servizi disponibili per la Primitiva Text To Speech

- *language*, obbligatorio, indica il linguaggio del testo in questione in formato lowercase minuscolo in inglese ('english', 'italian', ...). Le lingue disponibili sono riportate in Tabella 17;

Servizio	Lingue disponibili
Azure Neural TTS	Italiano, Francese, Tedesco, Inglese
Kokoro TTS	Italiano, Francese, Tedesco, Inglese

Tabella 17 - Lista delle lingue disponibili per la Primitiva Text To Speech

- *typology*, obbligatorio, indica la tipologia della risorsa.

2.17. Riconoscimento delle immagini simili

I database vettoriali offrono la possibilità di archiviare e recuperare vettori come punti ad alta dimensione. Vengono generalmente utilizzati dove sono necessarie operazioni ad alte prestazioni sui dati, infatti, questa primitiva ha l'obiettivo di rappresentare le immagini all'interno di un database vettoriale per ottimizzare le operazioni sulle immagini.

Il database vettoriale ottenuto da questa primitiva può essere utilizzato per operazioni di confronto e di recupero di immagini.

2.17.1. API da richiamare

Per eseguire l'API di *Riconoscimento delle immagini simili* sarà necessario richiamare la seguente API:

</api/v1/batch/tasks/creazione-dbv>

È di seguito riportato un esempio di chiamata all'API:

```
{
  "codiceOwnerSpecifico": "",
  "descrizione": "RICONOSCIMENTO IMMAGINI SIMILI",
  "publicTask": false,
  "uuidRisorseDigitali": {
    "57D8BCF349F74A0683669F400620D806": {}
  },
  "crossInfo": {
    "service": "dinov2-small",
    "language": "italian",
    "typology": "TESTO"
  }
}
```

Di seguito sono elencati i parametri che caratterizzano questa API, come possono essere popolati e la loro eventuale obbligatorietà. Quando si utilizza la primitiva è necessario considerare alcuni parametri fondamentali:

- *language*, obbligatorio. Identificativo della lingua da utilizzare per le operazioni. Nella primitiva attualmente non viene utilizzato. Si può usare il valore di default "italian";
- *typology*, obbligatorio. Identificativo della tipologia di dato che si sta utilizzando. Nella primitiva attualmente non viene utilizzato. Si può usare il valore di default "TESTO";
- *service*, obbligatorio. Identificativo del servizio backend che viene istanziato e supportato dalla primitiva. Questo influenza la qualità dell'operazione ed è stato selezionato dal gruppo AI di riferimento.

2.18. Immagini simili

La ricerca vettoriale avviene tramite determinati algoritmi che operano su di un database vettoriale. La primitiva prevede di ricevere in input un'immagine, che servirà per cercare le immagini più simili alla stessa. La primitiva prevede diverse politiche di similarità, definite esplicitamente come parametro in input.

L'output della primitiva in oggetto sarà la lista delle immagini più simili a quella passata in input.

2.18.1. API da richiamare

Per eseguire l'API di *Immagini simili* sarà necessario richiamare la seguente API: </api/v1/gpa/bitstreams/{uuidBitstream}/simili>

È di seguito riportato un esempio di chiamata all'API:

```
{  
  "numberOfResults": 5,  
  "searchBy": "entity",  
  "logicalOperator": "AND",  
  "listOfEntities": ["Aereo"],  
  "listOfSDFEntities": []  
}
```

Di seguito sono elencati i parametri che caratterizzano questa API, come possono essere popolati e la loro eventuale obbligatorietà. Quando si utilizza la primitiva è necessario considerare alcuni parametri fondamentali:

- *numberOfResults*, opzionale. Numero di immagini simili da recuperare. Default 5;
- *searchBy*, opzionale. Parametro di politica di similarità. Può essere 'similarity', 'entity' o 'both';
- *logicalOperator*, opzionale. Operatore logico. Può essere 'AND', 'OR' o 'NOT';
- *listOfEntities*, opzionale. Lista di entità da utilizzare per la ricerca;
- *listOfSDFEntities*, opzionale. Lista di entità del Soggettario di Firenze da utilizzare per la ricerca.

2.19. OMR

Nel contesto di I.PaC, la primitiva *OMR* (Optical Music Recognition) ha l'obiettivo di convertire spartiti musicali stampati in formato "MUSIC XML" e di generarne un'immagine PDF. Per valutare la qualità dell'elaborazione del servizio, vengono impiegate due metriche, una basata sulla coerenza della Time Signature individuata (4/4, 3/4, etc.) con l'effettiva durata delle battute; l'altra utilizza un modello ResNet per calcolare la somiglianza tra l'immagine originale ed il pdf prodotto dal servizio.

2.19.1. API da richiamare

Per eseguire l'API di *OMR* sarà necessario richiamare la seguente API: </api/v1/batch/tasks/creazione-cpa-omr>

È di seguito riportato un esempio di chiamata all'API:

```
{
  "codiceOwnerSpecifico": "",
  "descrizione": "OMR",
  "publicTask": false,
  "uuidRisorseDigitali": {
    "57D8BCF349F74A0683669F400620D806": {}
  },
  "crossInfo": {
    "service": "audiveris",
    "typology": "immagine"
  }
}
```

Di seguito sono elencati i parametri che caratterizzano questa API, come possono essere popolati e la loro eventuale obbligatorietà. Quando si utilizza la primitiva è necessario considerare alcuni parametri fondamentali:

- *service*, obbligatorio, indica il programma da utilizzare per l'elaborazione. Nella tabella sottostante, le cui colonne riportano le seguenti informazioni:
 - Servizio: il nome del servizio che la primitiva può utilizzare;
 - Parametro: il parametro da passare in input;
 - Disponibile ad elaborazione backend: se il servizio è disponibile negli ambienti o meno;
 - Open Source: se il servizio è open source o meno, sono riportati i servizi utilizzabili dalla primitiva in oggetto.

Servizio	Parametro	Disponibile ad elaborazione backend	Open source
Audiveris	audiveris	SI	SI

Tabella 18 - Servizi disponibili per la Primitiva OMR

- *typology*, obbligatorio, indica la tipologia della risorsa.

2.20. Generazione Audio da XML Musicali

La primitiva di *Generazione Audio da XML Musicali* ha lo scopo di creare un file audio in formato standard (WAV) partendo da una partitura digitale in formato MUSICXML, tipicamente generata da un servizio di Optical Music Recognition (OMR).

Per massimizzare la qualità del risultato finale, la primitiva adotta un approccio di elaborazione differenziato, che si adatta dinamicamente alla complessità della partitura in input.

2.20.1. API da richiamare

Per eseguire l'API di *Generazione Audio da XML Musicali* sarà necessario richiamare la seguente API:

</api/v1/batch/tasks/creazione-cpa-xmlmidi>

È di seguito riportato un esempio di chiamata all'API:

```
{
  "codiceOwnerSpecifico": "",
  "descrizione": "CPA XML MIDI",
  "publicTask": false,
  "uuidBitstream": {
```

```
"3909901897B9AD18E0634A0DFE0A0F82": { }
},
"crossInfo": {
  "service": "partitura"
}
}
```

Di seguito sono elencati i parametri che caratterizzano questa API, come possono essere popolati e la loro eventuale obbligatorietà. Quando si utilizza la primitiva è necessario considerare alcuni parametri fondamentali:

- *service*, obbligatoria, è la libreria da utilizzare per elaborare la conversione

Nella tabella sottostante, le cui colonne riportano le seguenti informazioni:

- Servizio: il nome del servizio che la primitiva può utilizzare;
- Parametro: il parametro da passare in input;
- Disponibile ad elaborazione backend: se il servizio è disponibile negli ambienti o meno;
- Open Source: se il servizio è open source o meno,

sono riportati i servizi utilizzabili dalla primitiva in oggetto.

Servizio	Parametro	Disponibile ad elaborazione backend
Partitura	partitura	SI

Tabella 19 - Servizi disponibili per la Primitiva Generazione Audio da XML Musicali

2.21. Formattazione testo

La primitiva *Formattazione testo* è un abilitatore che si occupa della formattazione del testo ottenuto tramite OCR. Il suo scopo è organizzare e strutturare il contenuto, effettuando la mappatura dei campi in base alle necessità del caso d'uso.

2.21.1. API da richiamare

Per eseguire l'API di *Formattazione testo* sarà necessario richiamare la seguente API:
</api/v1/batch/tasks/creazione-format-text>

È di seguito riportato un esempio di chiamata all'API:

```
{
  "codiceOwnerSpecifico": "",
  "descrizione": "FORMAT TEXT",
  "publicTask": false,
  "uuidRisorseDigitali": {
    "179E0250D6754751B98AF766D57723C4": {}
  },
  "crossInfo": {
    "service": null,
    "language": "italian",
    "typology": "monografia",
    "tipoScheda": "A",
    "layout": "layout_1"
  }
}
```

Di seguito sono elencati i parametri che caratterizzano questa API, come possono essere popolati e la loro eventuale obbligatorietà. Quando si utilizza la primitiva è necessario considerare alcuni parametri fondamentali:

- *service*, di tipo stringa e obbligatorio, indica il servizio di formattazione da utilizzare (es. "gpt-4o-mini");
- *language*, di tipo stringa e obbligatorio, indica la lingua della risorsa, da passare in lowercase in inglese (e.g., italian, english...);
- *typology*, di tipo stringa e obbligatorio, indica la tipologia della risorsa (e.g., testo);
- *tipoScheda*, di tipo stringa e obbligatorio, indica il tipo di scheda da elaborare (es. "RA", "A");
- *layout*, di tipo stringa e obbligatorio, indica il layout da applicare al testo (es. "layout_1", "layout_2").

2.22. Adattamento Latino per Epigrafi

La *Primitiva Adattamento Latino per Epigrafi* permette l'analisi di immagini (tipologia epigrafi) per il riconoscimento e l'estrazione di testi latini presenti all'interno di esse.

2.22.1. API da richiamare

Per eseguire l'API di *Adattamento Latino per Epigrafi* sarà necessario richiamare la seguente API: [/api/v1/batch/tasks/creazione-cpa-epigrafiLatine](#)

Di seguito è riportato un esempio della chiamata:

```
{
  "codiceOwnerSpecifico": "",
  "descrizione": "EPIGRAFE LATINE",
  "publicTask": false,
  "uuidRisorseDigitali": {
    "6FDFC65FB44E4EE3B52C0E1C3AFA900D": {}
  },
  "crossInfo": {
    "service": "MISTRAL_MEDIUM_2505",
    "language": "italian",
    "typology": "Immagine",
    "temperature": 0
  }
}
```

Di seguito sono elencati i parametri che caratterizzano questa API, come possono essere popolati e la loro eventuale obbligatorietà. Quando si utilizza la primitiva è necessario considerare alcuni parametri fondamentali:

- *service*, di tipo stringa e obbligatorio, indica il servizio di formattazione da utilizzare (e.g. "llama_3_2_90B");
- *language*, di tipo stringa e obbligatorio, indica la lingua della risorsa, da passare in lowercase in inglese (e.g., italian, english, ...);
- *typology*, di tipo stringa e obbligatorio, indica la tipologia della risorsa (e.g., immagine, testo);
- *temperature*, di tipo float e obbligatorio, indica il parametro di temperatura da fornire al modello (e.g. 0.1).

3. Request e response CPA

Il presente capitolo riporta la generica *request* da utilizzare per l'invocazione dei servizi CPA (servendosi delle informazioni fornite nel capitolo precedente) e la relativa *response*, fornendo, contestualmente, una descrizione dettagliata per ogni singolo campo di queste.

3.1. Request CPA

Di seguito è riportato un esempio di request generica CPA con le relative descrizioni per ciascun tag.

codiceOwnerSpecifico	string	
	example: dlherew3f	
	Contiene un codice aggiuntivo generato dal chiamante che può essere usato come filtro in una eventuale ricerca di task usando il servizio apposito	
descrizione	string	
	Contiene un'eventuale descrizione specifica del task che viene sottomesso	
uuidRisorseDigitali	{	
	description:	Contiene un elenco di risorse digitali da processare. Se valorizzato non viene considerata la lista in 'uuidBitstream'. Da considerare che i parametri uuidRisorseDigitali e uuidBitstream sono mutualmente esclusi vi, ma almeno uno dei due deve essere presente. ATTENZIONE: si può tranquillamente passare un oggetto vuoto a questa chiave, verrà preso in considerazione solo il parametro crossInfo
	< * >:	CreaContentProcessCpaInfo{
	description:	Contiene le informazioni e proprietà che servono al servizio realative alla risorsa. Questo parametro è obbligatorio, e i valori dipendono dal servizio o stesso che si sta chiamando.
	service*	string
		Modello di AI che si desidera utilizzare per l'operazione richiesta. Per ogni servizio esiste un valore di default, ma comunque deve essere valorizzato, anche con null.
	language	string
		Linguaggio associato alle risorse che si stanno caricando. In alcuni servizi potrebbe essere obbligatorio.
	typology	string
		Tipologia associato alle risorse che si stanno caricando. Es Monografia oppure Immagine Enum: [Monografia, Immagine]
	temperature	number(\$float)
		Indica quanto il servizio AI deve essere sensibile alla lavorazione.
	tipoScheda	string

		Indica la tipologia della scheda da passare alla I A.
layout	string	
		Indica il layout della scheda da passare alla IA.
scale	integer(\$int32)	
		Indica il valore di upscaling da apportare all'immagine.
tile	integer(\$int32)	
		Indica i valori con cui il servizio lavora per il rendering.
tile_pad	integer(\$int32)	
		Indica i valori con cui il servizio lavora per il rendering.
noise_level	integer(\$int32)	
		Valore per soppressione del rumore.
uuidStrutturaFisica	string	
		UUID della struttura fisica da essere processata.
heuristics	boolean	
		Indica se usare o meno le euristiche così da estrarre maggiori informazioni.
titlePosition	string	
		Indica l'allineamento dei titoli.
diarization	boolean	
		Valore per estrarre la diarizzazione, ovvero associa il testo estratto allo speaker.
size	string	
		Dimensione del modello.
similarityThreshold	number(\$float)	
		Soglia di similarità per il confronto semantico.
	}	
uuidBitstream	{	
	description:	Contiene un elenco di Bitstream da processare. Se presente non viene usato il campo contenente l'elenco delle risorse digitali. Da considerare che i parametri uuidRisorseDigitali e uuidBitstream sono mutualmente esclusivi, ma almeno uno dei due deve essere presente.
		ATTENZIONE:
		si può tranquillamente passare un oggetto vuoto a questa chiave, verrà preso in considerazione solo il parametro crossInfo
< * >:	CreaContentProcessCpaInfo{	
	description:	Contiene le informazioni e proprietà che servono al servizio relative alla risorsa. Questo parametro è obbligatorio, e i valori dipendono dai servizi o stesso che si sta chiamando.
	service*	string
		Modello di AI che si desidera utilizzare per l'operazione richiesta. Per ogni servizio esiste un valore di default, ma comunque deve essere valorizzato, anche con null.
	language	string
		Linguaggio associato alle risorse che si stanno caricando.

typology	string
	Tipologia associato alle risorse che si stanno caricando. Es Monografia oppure Immagine
	Enum:
	[Monografia, Immagine]
temperature	number(\$float)
	Indica quanto il servizio AI deve essere sensibile alla lavorazione.
tipoScheda	string
	Indica la tipologia della scheda da passare alla IA.
layout	string
	Indica il layout della scheda da passare alla IA.
scale	integer(\$int32)
	Indica il valore di upscaling da apportare alla risorsa.
tile	integer(\$int32)
	Indica i valori con cui il servizio lavora per il rendering.
tile_pad	integer(\$int32)
	Indica i valori con cui il servizio lavora per il rendering.
noise_level	integer(\$int32)
	Valore per soppressione del rumore.
uuidStrutturaFisica	string
	UUID della struttura fisica da essere processata.
heuristics	boolean
	Indica se usare o meno le euristiche così da estrarre maggiori informazioni.
titlePosition	string
	Indica l'allineamento dei titoli.
diarization	boolean
	Valore per estrarre la diarizzazione, ovvero associa il testo estratto allo speaker.
size	string
	Dimensione del modello utilizzato.
similarityThreshold	number(\$float)
	Soglia di similarità per il confronto semantico.
	}
	}
crossInfo*	CreaContentProcessCpaInfo{
description:	Contiene le informazioni e proprietà che servono al servizio relativamente alla risorsa. Questo parametro è obbligatorio, e i valori dipendono dal servizio stesso che si sta chiamando.
service*	string

Modello di AI che si desidera utilizzare per l'operazione richiesta. Per ogni servizio esiste un valore di default, ma comunque deve essere valorizzato, anche con null.

language	string Linguaggio associato alle risorse che si stanno caricando.
typology	string Tipologia associato alle risorse che si stanno caricando. Es Monografia oppure Immagine Enum: [Monografia, Immagine]
temperature	number(\$float) Indica quanto il servizio AI deve essere sensibile alla lavorazione.
tipoScheda	string Indica la tipologia della scheda da passare alla IA.
layout	string Indica il layout della scheda da passare alla IA.
scale	integer(\$int32) Indica il valore di upscaling da apportare.
tile	integer(\$int32) Indica i valori con cui il servizio lavora per il rendering.
tile_pad	integer(\$int32) Indica i valori con cui il servizio lavora per il rendering.
noise_level	integer(\$int32) Valore per soppressione del rumore.
uuidStruttura Fisica	string UUID della struttura fisica da essere processata.
heuristics	boolean Indica se usare o meno le euristiche così da estrarre maggiori informazioni.
titlePosition	string Indica l'allineamento dei titoli.
diarization	boolean Valore per estrarre la diarizzazione, ovvero associa il testo estratto allo speaker.
size	string Dimensione del modello utilizzato.
similarityThreshold	number(\$float) Soglia di similarità per il confronto semantico.
}	
}	

3.2. Response CPA

Di seguito è riportato un esempio di response generica CPA con le relative descrizioni per ciascun tag.

description:	Contiene il dettaglio della risposta fornita dal servizio REST
uuid	string Contiene l'identificativo del Task. Può essere usato per successive ricerche
uuidPadre	string Contiene l'eventuale identificativo del Task Padre del Task specifico. Può essere usato per successive ricerche
uuidComplexBatch	string Contiene l'identificativo dell'eventuale Task Complesso che ha generato il Task specifico. Può essere usato per successive ricerche
descrizione	string Contiene la descrizione del Task specifico
descrizioneTenancy	string Contiene la descrizione della Tenancy correlata al Task
codiceTipologia	string Contiene la tipologia del Task. Viene generata in automatico dal sistema durante la creazione del Task e può essere usata per successive ricerche di Task della medesima Tipologia
descrizioneAttivita	string Contiene la descrizione dell'attività associata alla Tipologia. Anche in questo caso viene generate in automatico
entitaGenerata	string <i>example: Risorsa</i> Contiene l'eventuale riferimento all'entità generata. Viene generata in automatico e fa riferimento ad un vocabolario chiuso I.PaC. Può valere: Risorsa,Oggetto,Pacchetto,Bitstream, Dataset
codiceFase	string

example: STARTED

Contiene la fase del Task. Le fasi stazionarie sono ERROR, COMPLETED e STARTED. Un task in esecuzione assume fase STARTED mentre se terminato correttamente COMPLETED

codiceOwner

string

Contiene l'identificativo dell'operatore che ha effettuato la richiesta del Task. Questa informazione è la medesima inviata in fase di autenticazione (Rif. Servizi di Autenticazione)

codiceOwnerSecondario

string

Contiene l'identificativo aggiuntivo che puo' essere dato in fase di creazione del Task dall'utente richiedente. E' utile come filtro ulteriore in una ricerca di Task

dataSchedulazione

integer(\$int64)

example: 1753188234312

Indica la data di sottomissione del Task

dataAvvio

integer(\$int64)

example: 1753188234312

Indica la data in cui il Task e' stato avviato dal Sistema sulla base del proprio carico

dataFineEsecuzione

integer(\$int64)

example: 1753188234312

Indica la data in cui il Task si e' concluso ed è quindi in fase COMPLETE o ERROR

informazioniExtra

InformazioniExtra{

description:

N.B. Gli attributi delle Response sono valorizzati a seconda del tipo di Request che viene inviata per il task specifico, pertanto alcuni di questi attributi potrebbero non essere valorizzati.

idPacchetto

integer(\$int64)

id del Pacchetto che contiene i dati inerenti a Risorse Digitali e/o Beni Culturali oltre ad un potenziale elenco di Pacchetti Contenuti (si veda 'idPacchettoContenuto')

idPacchettoContenuto

integer(\$int64)

id dello specifico Pacchetto che contiene i dati inerenti a Risorse Digitali e/o Beni Culturali

uuidRisorsaDigitale

string

uuid della Risorsa digitale

uuidsRisorseDigitali

[...]

	uuidBitstream	string	uuid del Bitstream
	uuidOggettoDigitale	string	uuid Oggetto digitale
	uuidTaskConfermaRisorsa	string	uuid dell'eventuale Task di Conferma della Risorsa che viene eseguito come figlio
	uuidsTaskConfermaRisorse	[...]	
	creaEliminaInfoResult	CreaEliminaBitstreamInfoResult{...}	
	creaDownloadInfoResult	DetailsCreaDownloadRisorsaTaskRisultato{...}	
	ackContent	string	ACK Content
	externalResult	{...}	
	childTasks	[...]	
	}		
datiMonitoraggio	DatoMonitoraggioTaskAsincrono{		
	description:	Contiene i dati di monitoraggio se previsti per il task. NOTA:	questo parametro non viene valorizzato quando si crea un task.
	valoreAvanzamento	integer(\$int64)	Rappresenta il valore da usare in relazione a quello 'finale' per sapere la percentuale di avanzamento
	valoreFinale	integer(\$int64)	Rappresenta il valore finale che l'avanzamento puo' valere
	fasiMancanti	integer(\$int64)	Rappresenta il numero di fasi mancanti per terminare il task
	descrizioneFase	string	Contiene la descrizione della Fase
	messaggio	string	Contiene un eventuale messaggio utile per il monitoraggio
	}		
streamInfo	[		
		Contiene l'eventuale elenco di stream che possono essere chiesti al Task	
		GenericDLStreamInfoDTO{...}]	
parametro	string		

Contiene l'oggetto serializzato in formato json della request body specifico per task

}