**Data Dictionary**

**Premessa:** i termini “String”, “Numbers”, “Structured”, “Array” sono da intendersi come definiti nella specifica JSON (<http://www.json.org)>. I simboli “-->” e “<--“ sono da intendersi, rispettivamente, come messaggio emesso dal *Client* (e ricevuto dal *Server*) e messaggio emesso dal *Server* (e ricevuto dal *Client*). I termini “*Server*” e “*Client*” sono da intendersi contestualizzati nella parte di progetto (Applicazione o Libreria) di cui si sta facendo riferimento.

**Available Services**

|  |  |
| --- | --- |
| **Definizione** | *Services* messi a disposizione dal *Broker* a tutti i *Clients* del sistema*.* |
| **Tipo di dato** | Lista di istanze di *Service* |
| **Derivato da** | *Service* |

**Broker**

|  |  |
| --- | --- |
| **Definizione** | Un componente del sistema che si occupa di fornire ai *Clients* i servizi disponibili all’interno del sistema. |

**Client – Applicazione[[1]](#footnote-1)**

|  |  |
| --- | --- |
| **Definizione** | Un *Client* è un componente del sistema che attraverso il *Broker* effettua richieste ai *Server* che forniscono il servizio richiesto. |
| **Derivato da** | *Node* |

**Client - Libreria[[2]](#footnote-2)**

|  |  |
| --- | --- |
| **Definizione** | Un programma finalizzato ad inviare *JSON-RPC Requests* e ottenere *JSON-RPC Responses* attraverso lo standard JSON-RPC. |

**Error**

|  |  |
| --- | --- |
| **Definizione** | Messaggio inviato dal *Server* al *Client* quando una richiesta non viene eseguita correttamente. Per i codici di errore con i messaggi corrispondenti fare riferimento alla sezione “Tipo di dato” di questa voce. |
| **Tipo di dato** | Structured**:**   * **code:** valore numerico che specifica il tipo di errore. Questo campo deve essere specificato per ogni tipo di errore.  |  |  |  | | --- | --- | --- | | **Codice (code)** | **Messaggio (message)** | **Significato** | | -32700 | Parse error | Errore di traduzione dell’oggetto JSON ricevuto dal Server. | | -32600 | Invalid Request | *JSON-RPC Request* non valida. | | -32601 | Method not found | Il metodo JSON-RPC non esiste o non è disponibile. | | -32602 | Invalid params | Parametri/o non validi/o per il metodo richiesto. | | -32603 | Internal error | Errore interno JSON-RPC. | | Da -32000 a -32099 | Server error | Riservati per errori definiti al di fuori della libreria. |  * **message:** una String che descrive brevemente l’errore riscontrato. * **data:** un valore di tipo Primitive o Structured che contiene informazioni aggiuntive riguardanti l’errore riscontrato. Questo campo può essere omesso. |
| **Esempi di utilizzo** | --> {"jsonrpc": "2.0", "method": "foobar, "params": "bar", "baz]  <-- {"jsonrpc": "2.0", "error": {"code": -32700, "message": "Parse error"}, "id": null} |
| **Derivato da** | *JSON-RPC Response* |

**JSON-RPC Request**

|  |  |
| --- | --- |
| **Definizione** | Una risorsa che rappresenta una chiamata a procedura remota (rpc-call), cioè una richiesta inviata da un *Client* ad un *Server* al fine di usare un *Service*. |
| **Tipo di dato** | Structured:   * **jsonrpc:** String rappresentante la versione di json-rpc. * **method:** String contenente il nome del metodo che deve essere invocato. * **params:** valore Structured rappresentante l’insieme dei parametri utilizzati nell’invocazione. * **id:** valore String o Numer stabilito dal *Client* per identificare la richiesta. |
| **Esempi di utilizzo** | {"jsonrpc": "2.0", "method": "sum", "params": [12, 6], "id": 1} |

**JSON-RPC Response**

|  |  |
| --- | --- |
| **Definizione** | Una risorsa che rappresenta la risposta conseguita dal Server a seguito di una JSON-RPC Request ricevuta. |
| **Tipo di dato** | Structured:   * **jsonrpc:** String rappresentante la versione di json-rpc. * **result:** attributo specificato solo in caso di successo dell’operazione. Il valore di questo attributo è determinato dal metodo invocato nel *Server*. * **error**: attributo presente solo se si verifica un errore durante l’invocazione del metodo. Se presente deve contenere un oggetto di tipo *JSON-RPC Error*. * **id:** valore String o Numer identico all’id della *JSON-RPC Request* associata. Se si verifica un errore durante l’identificazione dell’id relativo alla *JSON-RPC Request* associata questo campo deve essere nullo. |
| **Esempi di utilizzo** | {"jsonrpc": "2.0", "result": 18, "id": 1} |

**Notification**

|  |  |
| --- | --- |
| **Definizione** | Un messaggio inviato dal *Client* al *Server* che ha lo scopo di fornire informazioni senza aspettarsi un messaggio di risposta. |
| **Tipo di dato** | *JSON-RPC Request* con l’id non specificato. |
| **Esempi di utilizzo** | --> {"jsonrpc": "2.0", "method": "update", "params": [1,2,3,4,5]}  --> {"jsonrpc": "2.0", "method": "foobar"} |
| **Derivato da** | *JSON-RPC Request* |

**Metadata**

|  |  |
| --- | --- |
| **Definizione** | L’insieme delle informazioni scambiate tra gli attori del sistema che descrivono un *Service*. |
| **Tipo di dato** | Structured:   * **title**: String rappresentante il nome del *Service* associato. * **description**: String rappresentante una breve descrizione del *Service* associato. * **keywords**: Array of String rappresentante le parole chiave del *Service* associato. * **applicationField**: String rappresentante il campo di applicazione del *Service* associato. * **owner**: String rappresentante l’identificativo del fornitore del *Service* associato. |

**Method**

|  |  |
| --- | --- |
| **Definizione** | Un identificativo utilizzato per invocare una procedura remota (RPC) disponibile nel sistema. |
| **Tipo di dato** | String |
| **Esempi di utilizzo** | {"jsonrpc": "2.0", "method": "subtract", "params": [42, 23], "id": 1}  “subtract” è il nome del metodo che si intende invocare attraverso la soprastante *JSON-RPC Request*. |

**Node**

|  |  |
| --- | --- |
| **Definizione** | Un componente del sistema che può utilizzare e/o fornire servizi. Ovvero, un *Node* potrebbe assumere la funzione di *Client* e/o di *Server*. |

**Publish (a *Service*)**

|  |  |
| --- | --- |
| **Definizione** | L’azione effettuata dal *Server* finalizzata a rendere disponibile il proprio *Service* all’interno del sistema. |

**Server – Applicazione**

|  |  |
| --- | --- |
| **Definizione** | Un *Server* è un componente del sistema che attraverso il *Broker* pubblica e fornisce ai *Client* rispettivamente il servizio erogato e la risposta associata a una richiesta ricevuta da un *Client*. |
| **Derivato da** | *Node* |

**Server - Libreria**

|  |  |
| --- | --- |
| **Definizione** | Un programma finalizzato a ricevere dai *Clients* *JSON-RPC Requests* ed elaborare *JSON-RPC Responses* da inviare in risposta alle *JSON-RPC Requests* associate. |

**Service**

|  |  |
| --- | --- |
| **Definizione** | Una risorsa scambiata nel sistema rappresentante una funzionalità fornita da un *Server*. |
| **Tipo di dato** | Structured   * **method:** String rappresentante il nome (univoco) del metodo associato alla risorsa. * ***Metadata:*** metadata relativi alla risorsa. Vedasi voce “*Metadata*” per i dettagli sulla composizione di questo attributo. |

**Transparent communication**

|  |  |
| --- | --- |
| **Definizione** | Il *Broker* permette ai *Clients* di invocare *Services* prendendosi carico della gestione della comunicazione tra i *Nodes. JSON-RPC Responses* e *JSON-RPC Requests* vengono inoltrate al *Broker* il quale si occupa di consegnarle ai rispettivi destinatari senza che il *Client* conosca direttamente il *Server* e viceversa. |

1. S’intende il significato che assume il termine nel contesto della parte di progetto relativo all’applicazione B. [↑](#footnote-ref-1)
2. S’intende il significato che assume il termine nel contesto della parte di progetto relativo alla libreria JSON-RPC. [↑](#footnote-ref-2)