

Capitolo 1

Protocollo messaggi

Per ogni azione del client è definito un protocollo per lo scambio dei messaggi, sia per comunicazioni **N2N** (*node-to-node*) che **N2S** (*node-to-server*).

1.1 Node To Server

Questa tipologia di comunicazione vede come entità coinvolte il client del giocatore e il server REST deputato alla gestione delle partite.

Il server REST espone le proprie risorse all'URL **http://server_root:8080/server_war_exploded/match/**

Tabella 1.1: Risorse REST

Risorsa	Percorso	Metodo REST	Input	Output
Dettagli match	/ (matchId)/info	GET	String matchId	ok(Match) noContent()
Lista match	/	GET		ok(List<Match>)
Crea match	/	POST	Match match	notAcceptable() created() notModified()
Entra in match	/ (matchId)/join	POST	String matchId String player.Json	ok(Match) badRequest()
Lascia match	/ (matchId)/leave	POST	String matchId String player.Json	noContent() ok()
Rimuovi match	/ (matchId)/delete	DELETE	String matchId	ok() notFound()

1.2 Node To Node

Le comunicazioni N2N avvengono all'interno di una rete **Token Ring** sfruttando il protocollo di trasporto **TCP**.

1.2.1 Modello messaggi

Per modellare i messaggi scambiati tra i vari nodi della rete è stata definita la classe **RingMessage**. Ogni messaggio è caratterizzato dalle seguenti proprietà:

- *MessageType* **type**;
- *String* **sourceAddress**;
- *String* **id**;
- *String* **content**;

Nota: il campo **sourceAddress** segue la convenzione *ip:port*. La tipologia di messaggio viene definita nell'enumerazione **MessageType** che comprende le seguenti tipologie:

- **ACK** - per messaggi di conferma;
- **GAME** - per messaggi riguardanti azioni relative al funzionamento del gioco;
- **TOKEN** - per messaggi inerenti la gestione del token;

1.2.2 Messaggi ACK

I messaggi **ACK** (*Acknowledgement*) vengono utilizzati da un nodo ricevente per confermare la ricezione di un messaggio al relativo mittente: il messaggio ACK ha lo stesso **id** del messaggio da confermare.

I messaggi ACK hanno tutti il formato definito nella seguente tabella.

Tabella 1.2: Formato messaggi ACK

Azione	Type	Source	Id	Content
Conferma ricezione	ACK	M_{in}	M_{in}	null

M_{in} si riferisce al messaggio ricevuto

1.2.3 Messaggi TOKEN

I messaggi **TOKEN** vengono utilizzati dai nodi della rete per gestire il movimento del token all'interno della rete: mediante un messaggio di questo tipo viene data la possibilità ad un nodo di cedere il token.

I messaggi TOKEN hanno tutti il formato definito nella seguente tabella.

Tabella 1.3: Formato messaggi TOKEN

Azione	Type	Source	Id	Content
Conferma ricezione	TOKEN	M	M	null

1.2.4 Messaggi GAME

I messaggi **GAME** vengono utilizzati per la trasmissione di informazioni relative ad eventi del gioco. I messaggi GAME hanno i formati definiti nella seguente tabella.

Tabella 1.4: Formato messaggi GAME

Azione	Type	Content
Entrata in partita	GAME	ENTER-PLAYER# <i>Player_JSON</i>
Uscita dalla partita	GAME	EXIT-PLAYER# <i>Player_JSON</i>
Verifica posizione giocatore	GAME	CHECK-POSITION
Notifica posizione giocatore	GAME	CHECK-POSITION# <i>Position_JSON</i>
Notifica movimento	GAME	MOVE# <i>Player_JSON</i> # <i>Position_JSON</i>
Notifica giocatore ucciso	GAME	KILLED# <i>Player_JSON</i>
Rilascio bomba	GAME	BOMB-RELEASE# <i>Bomb_JSON</i>
Esplosione bomba	GAME	BOMB-EXPLOSION# <i>Bomb_JSON</i>
Uccisione bomba	GAME	BOMB-KILL# <i>Player_JSON</i>
Fine partita	GAME	GAME-END