Progetto ws NAUSICAA-PCPLANET

In tutti i casi qui sotto descritti si tratta di servizi REST; verranno implementati i soli metodi HTTP necessari al funzionamento.
Tutti i vari metodi descritti sono accessibili previa autenticazione (user name, password → token).

Token.
A autenticazione avvenuta, un client riceve un identificativo utente numerico e un token, ad esempio un codice GUID, che identifica la sessione in corso; il client utilizzerà identificativo utente e il token a ogni richiesta successiva all'autenticazione.
Il token è valido:

- fino a quando il client indica il termine della sessione (end session);
- sempreché il client contatti il server entro un tempo predefinito dopo autenticazione (20 minuti predefinito); a ogni contatto avvenuto con successo tra client e server, l'intervallo di controllo viene ripristinato.

Eccettuato per la richiesta di autenticazione, a ogni richiesta il server deve verificare che il token sia ancora valido e che corrisponda con l'identificativo utente indicato (vedi nei dettagli).

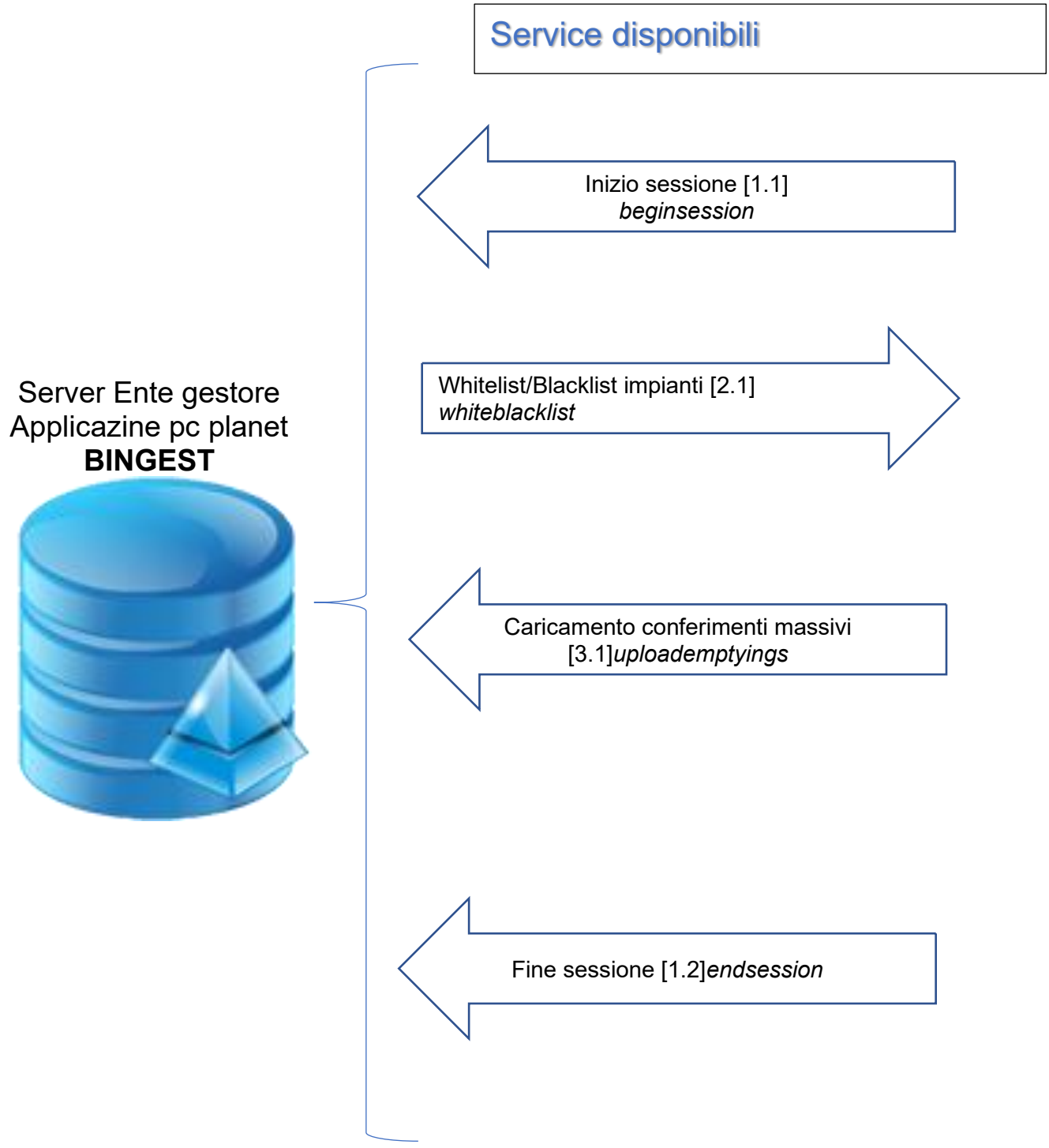
Colloquio.
Tutto il colloquio tra client e server, in entrambe le direzioni, avviene attraverso oggetti JSON.

Varie.
1) Ove previste, le password vanno passate codificate in MD5.
2) Le date vanno sempre indicate in formato ISO `'yyyy-MM-ddThh:mm:ss.zzz'`; la parte frazione di secondo può essere MOESSA.
3) I valori negli oggetti JSON descritti da qui in avanti adottano le seguenti convenzioni tipografiche:

- le parentesi angolari (< e >) delimitano una descrizione del contenuto;
- tutti i valori delimitati da doppi apici (") sono stringhe di testo;
- tutti gli altri valori, ove non diversamente specificato, sono numerici.

4) L'indirizzo base degli oggetti è: `[http | https]://<indirizzo del server:porta>/rest/`.
5) in caso di errore, oltre a opportuno oggetto JSON, viene restituito un codice errore HTTP adeguato.

Specifica servizio BINGEST



1. Inizio/fine sessione

1.1. Inizio sessione

Metodo: POST

Oggetto: beginsession (da richiamare sempre prima di qualsiasi operazione se trascorsi più di 20 min, termine validità token)

```
Endpoint TEST: https:// <indirizzo del server:porta>//rest/beginsession
Endpoint PRODUZIONE: https:// <indirizzo del server:porta>//rest/beginsession
```

Il client formulerà una richiesta strutturata come segue:

```
{
    "username": "<nome dell'utente>",
    "password": "<password associata all'utente>"
}
```

Risposta se autenticazione avvenuta:

```
{
    "result": "OK",
    "username": "<nome dell'utente>",
    "idutente": <identificativo dell'utente>,
    "token": "<identificativo sessione restituito dal server per le richieste successive>"
}
```

Risposta se autenticazione fallita o in caso di errore:

```
{
    "result": "KO",
    "username": "<nome dell'utente>",
    "message": "<motivazione del fallimento>"
}
```

1.2. Termine della sessione.

Metodo: POST

Oggetto: endsession

```
Endpoint TEST: https:// <indirizzo del server:porta>//rest/endsession
Endpoint PRODUZIONE: https:// <indirizzo del server:porta>//rest/endsession
```

Il client invierà una notifica strutturata come segue:

```
{
    "idutente": <identificativo dell'utente>,
    "token": "<identificativo sessione>"
}
```

Risposta se terminazione avvenuta (da questo momento il token non sarà più valido):

```
{
    "result": "OK",
    "idutente": <identificativo dell'utente>,
    "token": "<identificativo sessione non più valido>"
}
```

Risposta se terminazione fallita o in caso di errore:

```
{
    "result": "KO",
    "idutente": <identificativo dell'utente>,
    "token": "<identificativo sessione>",
    "message": "<motivazione del fallimento>"
}
```

2. WHITELIST E BLACKLIST

2.1. Scarico dati whitelist e blacklist

Metodo: GET

Oggetto: getwhiteblacklist

Consente l'accesso a whitelist e blacklist, a partire dalla data indicata fino alla data corrente.

```
Endpoint TEST: https:// <indirizzo del server:porta>//rest/whiteblacklist
Endpoint PRODUZIONE: https:// <indirizzo del server:porta>/rest/ whiteblacklist
```

Il client formula una richiesta come segue nella url:

```
{
    "idutente": <identificativo dell'utente>,
    "token": "<identificativo sessione>",
    "codicecomune": <codice ISTAT del comune>,
    "datainizioperiodo": "<data formato ISO 8601 senza fuso orario>"
}
```

Risposta se la richiesta è stata soddisfatta correttamente:

```
{
    "result": "OK",
    "idutente": <identificativo dell'utente>,
    "token": "<identificativo sessione>",
    "count": <numero degli oggetti passati, per controllo>,
    "rows": [
  <elementi in whitelist/blacklist, vedi dettaglio>,
        …
    ]
}
```

Risposta in caso di errore:

```
{
    "result": "KO",
    "idutente": <identificativo dell'utente>,
    "token": "<identificativo sessione>",
    "message": "<motivazione del fallimento>"
}
```

2.2. Specifica dei singoli oggetti whitelist e blacklist

NOME CAMPO	TIPO	LUNGHEZZA MASSIMA	OBBLIGATORIETA	VALOTRI POSSIBILI
CODICECOMUNE	Alfanumerico	10	Si	
CODICEDISPOSITIVO	Alfanumerico	50	Si	
ISBLACKLIST	Intero		SI	1 se BLACKLIST, 0 altrimenti
CODICERIFIUTO	Alfanumerico	250	SI	Tipo di rifiuti abilitati, suddivisi da ;. Esempio: CRT;VRD;RSU
DATADECORRENZA	Data/Time	18	SI	FORMATO ISO 8601 SENZA FUSO ORARIO [YYYY]-[MM]-[DD]T[hh]:[mm]:[ss] Esempio: 2019-04-05T14:30:30
DATASCADENZA	Data/Time	18	NO	
TIPODISPOSITIVO	Alfanumerico	20	SI	Può assumere i valori 'TESSERA VIRTUALE', 'TESSERA FISICA', 'CONTENITORE CON TAG'

3. CARICO CONFERIMENTI.

3.1. Carico conferimenti

Metodo: POST

Oggetto: uploademptyings, carica l'elenco dei conferimenti massivi/singoli

```
Endpoint TEST: https:// <indirizzo del server:porta>//rest/ uploademptyings
Endpoint PRODUZIONE: https:// <indirizzo del server:porta>/rest/ uploademptyings
```

Il client invia una richiesta strutturata come segue:

```
{ "token": "<identificativo sessione>"
  "idutente": <identificativo dell'utente>,
  "rows": [
       <elenco degli oggetti conferimenti da caricare>,
        …
    ]
}
```

Risposta normale:

```
{
    "result": "OK",
    "idutente": <identificativo dell'utente>,
    "token": "<identificativo sessione>",
    "count": <numero degli oggetti caricati, per controllo>
}
```

Risposta in caso di errore:

```
{
    "result": "KO",
    "idutente": <identificativo dell'utente>,
    "token": "<identificativo sessione>",
    "message": "<motivazione del fallimento>"
}
```

3.2. Specifiche singoli oggetti carico conferimenti

NOME CAMPO	TIPO	LUNGHEZZA MASSIMA	OBBLIGATORIETA	VALORI POSSIBILI
CODICECOMUNE	Alfanumerico	10	SI	
CODANTENNA	Alfanumerico	50	NO	
DATACONFERIMENTO	Data/Time	18	SI	FORMATO ISO 8601 SENZA FUSO ORARIO [YYYY]-[MM]-[DD]T[hh]:[mm]:[ss] Esempio: 2019-04-05T14:30:30
CODMEZZO	Alfanumerico	50	SI se <TARGA> è vuoto	
TARGA	Alfanumerico	20	SI se <CODMEZZO> è vuoto	
CODIMPIANTO	Alfanumerico	50	NO	
LAT	Alfanumerico	50	NO	
LONG	Alfanumerico	50	NO	
CODTRANSPONDER	Alfanumerico	50	SI	
PESO	Numerico	Decimal(8,2)	NO	
PERCENTUALERIEMPIMENTO	Numerico	Decimal(5,2)	NO	
CODERRORE	Alfanumerico	50	NO	