PASSEPARTOUT
Software e servizi gestionali

WebShaker

Gianluca Suzzi

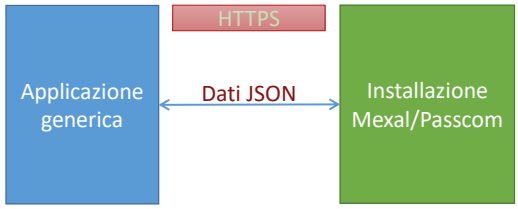
WebShaker

Cos'è?

WebShaker è un nuovo tipo di servizio Passepartout basato su protocollo https disponibile solo in presenza del modulo MDS (Shaker)
Tramite un protocollo di autenticazione con il server Mexal/Passcom, **qualsiasi applicazione esterna** è in grado di interagire con la base dati del gestionale mediante lo scambio di dati nel formato JSON (*JavaScript Object Notation*)

Il punto di forza di WebShaker è **l'assoluta indipendenza dal linguaggio e dalla piattaforma** utilizzati. L'unico prerequisito è che vengano utilizzati strumenti che permettano di fare richieste https verso i servizi Passepartout.

Esempi:
- Sito web su base JavaScript o back-end in PHP
- Applicazioni Mobile per Android o iOS
- Applicazioni desktop (C#, Java, Python, etc.)
- Smart Speakers (es. Amazon Alexa, Google Home, etc.)



In generale qualsiasi applicazione su qualsiasi piattaforma in grado di effettuare richieste https

WebShaker

Come si utilizza

WebShaker è disponibile sia su installazioni Live che locali con alcune differenze solo per quanto riguarda il metodo di autenticazione. Lo scambio dei dati JSON invece avviene nello stesso modo.

Installazioni Locali
L'accesso alle risorse (l'autenticazione) avviene inserendo le credenziali nell'header della richiesta https

Installazioni Live
Solo per le installazioni live ci sono due modalità di accesso alle risorse dell'installazione:

1. Come per le locali, l'accesso alle risorse avviene inserendo le credenziali nell'header della richiesta https
2. L'accesso alle risorse avviene tramite autenticazione secondo protocollo standard OAuth 2.0, nello specifico utilizzando lo strumento OpenID (al momento disponibile solo per Amazon Alexa)

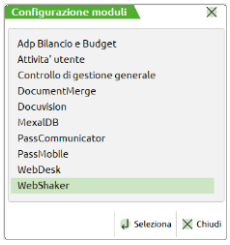
3

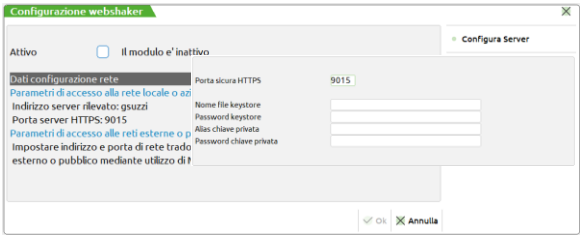
PASSEPARTOUT
Software e servizi gestionali

WebShaker

Installazione locale

E' stato inserito un nuovo servizio in ascolto sulla porta utilizzata per le connessioni sicure in https ed attivabile dal menu *Servizi-Configurazione-Configurazioni moduli-WebShaker*





Dati necessari per la connessione

- INDIRIZZO: indirizzo del server (IP pubblico, nome nella forma p1xxxxxx000.bp.pasccom.it)
- PORTA: porta https (tipicamente <porta-installazione>+4: es. 9004)
- USER: utente di Mexal/Passcom
- PWD: password utente Mexal/Passcom

Per le installazioni locali saranno utilizzati dei certificati autoprodotti (self-signed). Per utilizzare certificati propri è necessario inserirli tramite il pannello Configura Server

4

PASSEPARTOUT
Software e servizi gestionali

WebShaker

Installazione locale

Credenziali nell'header https

- URL = https://<indirizzo_server>:PORTA**/webshaker/spx/ppt** *(es. https://mioserver:9004/webshaker/spx/ppt)*
- Concatenare USER e PWD formando la seguente stringa: CREDENZIALI=<USER:PWD> es. “MARIO:ADMIN”
- Fare un encoding in base64 delle credenziali (es. JavaScript: btoa("MARIO:ADMIN"))

```
{
  'Authorization': 'Passepartout QURNSA46QQ==',
  'Content-type': 'application/json; charset=utf-8',
  'Cookie': 'JSESSIONID=tom02~04589776CBEE000155FE7ADBACD40395;path=/webshaker;Secure;HttpOnly'
}
```

La chiave *Cookie* contiene l'id di sessione inviato dal servizio di WebShaker che deve essere salvato dal client ed inserito in ogni richiesta https.
Viene gestita automaticamente dai Web Browser, mentre in altri contesti **DEVE** essere gestita dal programmatore.

5

PASSEPARTOUT

WebShaker

Installazione live con autenticazione in header https

Il servizio è sempre attivo sul cloud Passepartout

Dati necessari per la connessione

- URL = **https://webshaker.passepartout.cloud/webshaker/spx/ppt**
- DOMINIO = <dominio_della_installazione_live>
- USER = <utente_di_mexal_passcom>
- PWD = <password_utente_mexal_passcom>

Questi dati devono essere inseriti nell'header della richiesta https in questo modo:

- Concatenare USER e PWD formando la seguente stringa: CREDENZIALI=<USER:PWD> es. “MARIO:ADMIN”
- Fare un encoding in base64 delle credenziali (es. JavaScript: btoa("MARIO:ADMIN"))

6

PASSEPARTOUT

WebShaker

Installazione live con autenticazione in header https

Esempio di header https

L'header è analogo al caso di installazione locale con la differenza che la chiave Authorization deve contenere anche l'informazione sul dominio: *Dominio=MIODOMINIO*

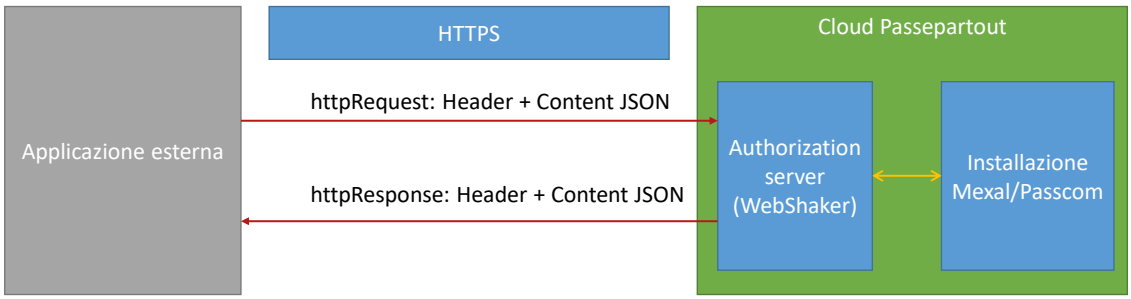
```
{
  'Authorization': 'Passepartout QURNSA46QQ== Dominio=MIODOMINIO',
  'Content-type': 'application/json; charset=utf-8',
  'Cookie': 'JSESSIONID=tom02~04589776CBEE000155FE7ADBACD40395;path=/webshaker;Secure;HttpOnly'
}
```

Per la chiave *Cookie* valgono le stesse raccomandazioni fatte per le installazioni locali

7

WebShaker

Installazione live con autenticazione in header https



Il body o content della richiesta https sarà un JSON formattato secondo regole che vedremo in seguito.

8

WebShaker

Installazione live con autenticazione OAuth 2.0

Il servizio è sempre attivo sul cloud Passepartout

Cos'è OAuth 2.0

OAuth 2.0, o più brevemente OAuth2, è un protocollo di rete aperto e standard, progettato specificamente per lavorare su http/https, che consente l'emissione di un token di accesso da parte di un server autorizzativo (identity manager) ad un client di terze parti, previa approvazione dell'utente proprietario della risorsa cui si intende accedere. Nel nostro contesto, il proprietario della risorsa è l'utente di Mexal/Passcom, il server autorizzativo è il servizio di WebShaker su OpenID ed i client di terze parti sono le applicazioni esterne.

È comunemente utilizzato come modalità per gli utenti in Internet per garantire alle applicazioni ed ai siti web l'accesso alle proprie informazioni senza fornire loro alcuna password (ad esempio login tramite Facebook o Account Linking per le Alexa Skills).

OAuth2 garantisce ai service provider l'accesso da parte di terzi ai dati degli utenti proteggendo contemporaneamente le loro credenziali.

WebShaker

Installazione live con autenticazione OAuth 2.0

Come si usa OAuth 2.0 con WebShaker

Sul cloud Passepartout è presente un servizio di autenticazione OAuth 2.0 basato su OpenID.

L'accesso a risorse private (nel nostro caso alle risorse presenti sulle installazioni Mexal o Passcom) avviene utilizzando il *Grant Type* di tipo *Authorization Code* il quale prevede le seguenti tre fasi:

1. una prima fase in cui si richiede l'autorizzazione ad eccedere alle risorse: ***Authorization Request.*** In questa fase viene rilasciato un codice di autorizzazione per la richiesta dei token: ***Authorization Code***
2. una seconda fase in cui per mezzo del codice ottenuto si richiede il token di accesso alle risorse (***Access Token***): ***Authorization Grant***
3. la terza fase è quella in cui, utilizzando l'*Access Token,* si possono richiedere direttamente le risorse al server

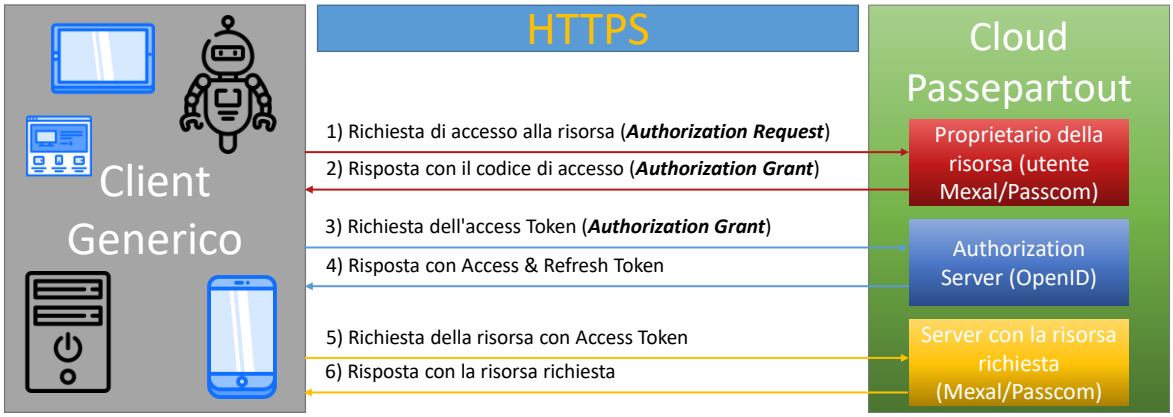
L'autenticazione viene fatta solo al primo accesso. I successivi accessi alle risorse verranno effettuati esclusivamente mediante lo scambio di Access Token all'interno degli header https

Attualmente OAuth2 è disponibile solo per effettuare *Account Linking* con Amazon Alexa

WebShaker

Installazione live con autenticazione OAuth 2.0

Flusso operativo in sintesi



11

PASSEPARTOUT

WebShaker

Contenuto della richiesta HTTPS

A prescindere dal tipo di sistema utilizzato per l'autenticazione (Locale, Live, Live con OAuth 2.0), il corpo della richiesta HTTPS consiste in un JSON formattato secondo opportune regole.

```
{
    'Client':{
      'Tipo' : 'WebShaker',
      'Versione' : 1000
    },
    'cmd' : 'EseguiEticCollageRemoto',
    'SiglaAzienda' : 'IMP',
    'SottoAzienda' : '',
    'Data' : '01/01/2020',
    'CodiceApp' : '123456CODICEAPP',
    'CodiceAppEst' : '',
    'NomeCollage' : 'nomeCollage',
    'NomeEtichetta' : 'NOME_ETIC',
    'DatiInput' : datiInput
}
```

```
{
  'str':{
    'Tipo': 'A',
    'Livello':0,
    'Valore':'Stringa'
  },
  'num':{
    'Tipo': 'N',
    'Livello':0,
    'Valore':'3'
  }
}
```

12

PASSEPARTOUT

WebShaker

E lato server?

Lato Mexal/Passcom abbiamo 2 possibilità:

1. Demandare tutto a WebShaker utilizzando funzioni native Passepartout: non è richiesta conoscenza SPRIX ma le funzioni disponibili saranno solo quelle rilasciate da Passepartout ***WORK IN PROGRESS***
2. Gestire lo scambio dei dati tramite Collage Server Remoto: è richiesta conoscenza SPRIX ma i limiti sono solo quelli imposti dal Collage Server Remoto

Nel caso di utilizzo del Collage Server Remoto il JSON di input può essere formato utilizzando 2 modalità:

1. JSON libero: lato client il JSON può essere definito liberamente dal programmatore. Lato Collage Server Remoto è però necessario scorrere il JSON con le opportune istruzioni SPRIX
2. JSON formattato: lato client il JSON deve seguire un pattern predefinito. Lato Collage Server Remoto può essere letto utilizzando le istruzioni SPRIX GETREM_STR, GETREM_NUM, GETREM_ARRAY

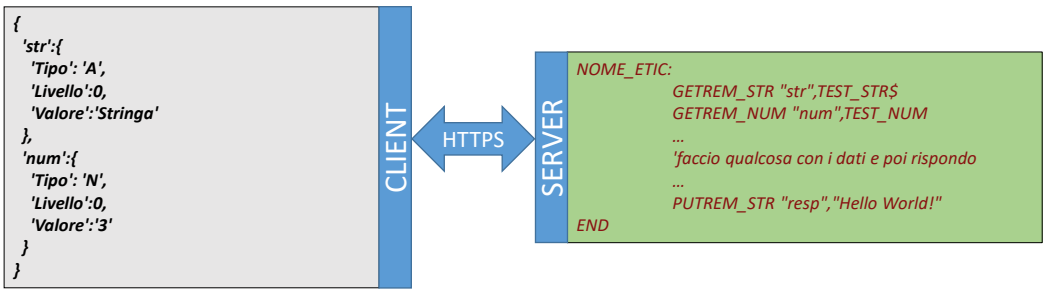
13

PASSEPARTOUT

WebShaker

E lato server?

Il seguente esempio mostra come leggere i dati in un Collage Server Remoto utilizzando un JSON formattato secondo il pattern Passepartout e gestibile lato server con le istruzioni GETREM



14

PASSEPARTOUT

WebShaker

Esempio applicazione web

Un semplice esempio di pagina web in HTML e JavaScript con JSON libero.
JavaScripit in questo caso è essenziale per poter fare chiamate https verso l'installazione (*XMLHttpRequest()*)

Lato server i dati verranno processati da un Collage Server Remoto utilizzando le istruzioni JSON di SPRIX

Per maggiori dettagli sulla configurazione e l'utilizzo di WebShaker si rimanda al manuale presente nell'Area Sviluppatori: ***Manuale WebShaker***

PASSEPARTOUT
Software e servizi gestionali

PASSEPARTOUT®
Software e servizi gestionali

WebShaker

Gianluca Suzzi