



UNIVERSITÀ
DEGLI STUDI DI BARI
ALDO MORO

Implementazione di un traduttore
utilizzando le

‘Microsoft Translator API’

(Microsoft Translator)

di SALVATORE DE ASTIS

Sommario

Introduzione	2
Implementazione	3
Creazione dell'account Microsoft	3
Creazione dell'account sul Marketplace di Windows Azure	3
Implementazione dell'applicazione 'Microsoft Traduttore'	11
Modulo 'Credentials.php'	11
Modulo 'AccessTokenAuthentication.php'	12
Modulo 'HttpTranslator.php'	13
Modulo 'index.php'	14
Modulo 'functions_js.js'	19
Modulo 'css.css'	20
Installazione dell'applicazione in XAMPP su piattaforma Windows	23
Installazione dell'applicazione in XAMPP su piattaforma Linux	25

Introduzione

Mentre il mondo diventa sempre più piccolo grazie all'utilizzo di molti strumenti di comunicazione come smartphone, tablet, personal computer, ecc., l'unico ostacolo che persiste tra persone di culture differenti è la lingua, e proprio per superare quest'ultima barriera di comunicazione che, in ambito applicativo, si stanno diffondendo i traduttori automatici.

Infatti per rendere le pagine web o le applicazioni utilizzabili anche da utenti con nazionalità e cultura differente da quella dello sviluppatore, vengono utilizzati traduttori automatici, i quali rimangono uno dei modi più efficaci per colmare il divario imposto dalla lingua. Ma non sempre la miglior tecnologia di traduzione automatica fornisce traduzioni accurate o adatte per un sito o per l'utente, proprio come accade con un traduttore umano. Inoltre, molte pagine web non sono totalmente tradotte, ma presentano solamente delle sezioni che sono disponibili in varie lingue; questo accade per colpa dell'elevato costo della traduzione e del costo che la traduzione ha nel tempo, dato dall'eventuali aggiornamenti del sito. Lo stesso vale per le applicazioni; è impossibile pensare che uno sviluppatore si affidi a dei traduttori umani così da rendere i contenuti dell'applicazione disponibili in varie lingue. Proprio in questo contesto che entra in gioco Microsoft Translator.

Nello specifico, Microsoft Translator è un servizio host accessibile via API (Application Program Interface), che permette di tradurre le applicazioni senza che sia necessario un traduttore umano. Può essere utilizzato per utilizzato in qualsiasi scenario in cui c'è bisogno di una traduzione, dalla reazione di una applicazione per smartphone o tablet, alla creazione di applicativi desktop senza dimenticarci di pagine web ecc. La traduzione può essere fatta utilizzando il widget di Microsoft Translator o un API Ajax, che possono essere integrati perfettamente nel sito senza il bisogno di ulteriori sforzi. In più il widget ha la capacità di migliorare la l'accuratezza grammaticale e contestuale delle traduzioni. Tutto il lavoro di traduzione viene effettuato grazie ai metodi *AddTranslation()* e *GetTranslations()* che formano, insieme anche al metodo *Translate()*, il **Collaborative Translation Framework**.

La traduzione non viene fatta solo attraverso un widget, ma anche attraverso varie interfacce come quella SOAP, http e Ajax. Il progetto per il corso di Sistemi per la collaborazione in rete si focalizza nel costruire un traduttore che utilizzi le API utilizzando l'interfaccia HTTP.

La seguente documentazione andrà a spiegare il funzionamento delle API attraverso il protocollo HTTP, andando a mostrare tutte le fasi necessarie per implementare il traduttore, senza tralasciare le fasi preliminari che servono per ottenere le credenziali token, necessarie per poter utilizzare la API.

Implementazione

Di seguito verrà mostrato lo sviluppo del traduttore. Iniziamo col mostrare come ottenere le credenziali token per utilizzare le API.

Creazione dell'account Microsoft

Per poter ottenere il token bisogna prima essere in possesso di un account Microsoft. L'account Microsoft è una combinazione di indirizzo di posta elettronica e password usata per accedere a servizi quali Outlook.com, Skype, SkyDrive, oltre che per dispositivi come i PC e tablet Windows 8, telefoni Windows Phone e console Xbox. Se si utilizza già un indirizzo di posta elettronica e una password per accedere a questi servizi, si possiede già un account Microsoft.

Nel caso in cui non si abbia un account Microsoft, è possibile crearlo al seguente indirizzo <http://login.live.com/> cliccando su *Iscriviti ora*.

Creazione dell'account sul Marketplace di Windows Azure

Per ottenere l'accesso alle API dobbiamo accedere al Marketplace Microsoft Azure, accessibile al seguente indirizzo <https://datamarket.azure.com/>. Anche qui c'è bisogno di un account, ma questa volta non dobbiamo creare un nuovo account, ma basta collegare il nostro account Microsoft a quello del Marketplace seguendo i seguenti passi:

- 1) Registrare un account sul Marketplace Windows Azure (nel caso in cui si abbia già un account, si può utilizzare quello).
- 2) Iscrivere all'API utilizzando l'account con cui ci si è registrati.
- 3) Iscrivere la propria applicazione sul Marketplace;
- 4) Infine si ottengono le credenziali Client ID e Client Secret dell'applicazione che è stata registrata.

Vediamo le varie schermate che appariranno durante questi passi.

Quando accediamo all'indirizzo <https://datamarket.azure.com/> verrà mostrata la seguente pagina:

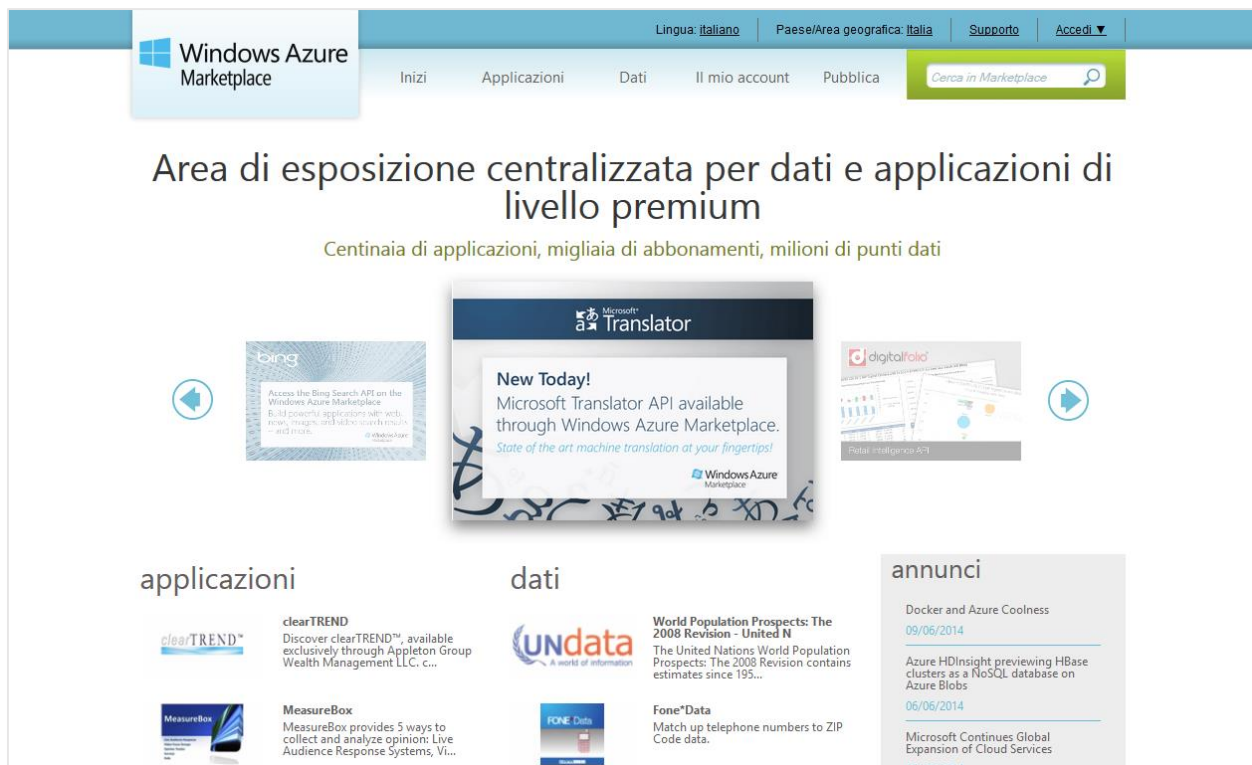


FIGURA 1 - SCHERMATA PRINCIPALE MARKETPLACE

Una volta arrivati a questa pagina bisogna accedere al nostro account cliccando su *Accedi*, posto in alto a destra. Cliccandoci verremo mandati alla schermata di login del Marketplace, ossia:

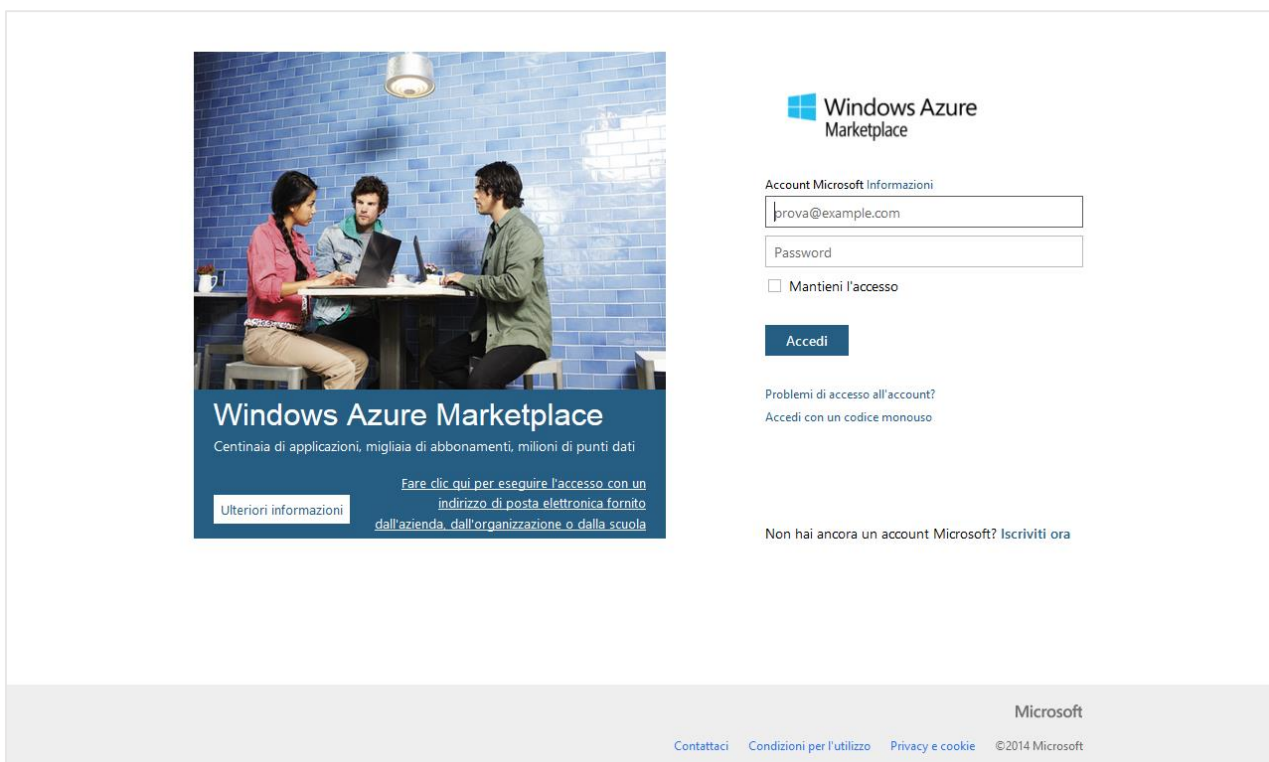


FIGURA 2 - SCHERMATA DI LOGIN DEL MARKETPLACE

Effettuando il login, verrà mostrata la schermata di registrazione dell'account:

Windows Azure Marketplace

italiano Italia Supporto Disconnetti

Inizi Applicazioni Dati Il mio account Pubblica

Cerca in Marketplace

Registrazione

Immetti le informazioni personali per creare un account Windows Azure Marketplace.
Il diritto alla protezione dei dati personali è importante per Microsoft. Per ulteriori informazioni, estrarre l'[informativa sulla privacy](#).

DETTAGLI ACCOUNT

* Nome

* Cognome

Organizzazione

Indirizzo di posta elettronica

Paese/Area geografica ?

Lingua ?

Partita IVA ?

☐ Autorizzo Microsoft a utilizzare il mio indirizzo di posta elettronica per inviare informazioni e offerte relative a Windows Azure Marketplace.

CONTINUA

HOME White paper Case study Video Documentazione	SFOGLIA Tutto Dati Applicazioni	ACCOUNT Informazioni sull'account Applicazioni personali Dati personali Chiavi dell'account	PUBBLICA Portale di pubblicazione Risorse online Video Kit pubblicazione dati Kit pubblicazione applicazioni	SVILUPPO Procedure Esempi di codice Registra l'applicazione creata Utilizzo dell'API di Microsoft Translator Balista per sviluppatori	SUPPORTO Forum/Blog Supporto fatturazione Supporto tecnico Modulo violazioni proprietà intellettuale Migliori risposte con voto più alto
---	---	--	--	--	--

FIGURA 3 - SCHERMATA DI REGISTRAZIONE

Inserite le credenziali, diventa cliccabile il tasto *Continua*, e accettando le condizioni per l'utilizzo del servizio, possiamo terminare la registrazione.

Windows Azure Marketplace

Italiano Italia Supporto Disconnetti

Inizi Applicazioni Dati Il mio account Pubblica Cerca in Marketplace

Registrazione

È necessario accettare le condizioni per l'utilizzo prima della creazione dell'account in Windows Azure Marketplace. Se non si accettano tali condizioni, la registrazione dell'account non verrà completata. È possibile stampare le condizioni per l'utilizzo facendo clic sul pulsante Stampa visualizzato di seguito. Tali condizioni sono inoltre disponibili in [Condizioni per l'utilizzo](#).

Stampa

MICROSOFT WINDOWS AZURE MARKETPLACE

CONTRATTO

DATA: Febbraio 2013

GRAZIE PER AVER SCELTO MICROSOFT WINDOWS AZURE MARKETPLACE. L'utente deve leggere il presente contratto con attenzione.

Il presente Contratto Microsoft Windows Azure Marketplace ("Contratto") intercorre tra l'utente, la società o un'altra persona giuridica ("utente") e Microsoft Corporation, One Microsoft Way, Redmond, Washington, USA o, in base alla sede principale dell'utente, una delle consociate di Microsoft Corporation identificate all'Articolo 15 ("Microsoft"). Il presente Contratto si applica all'accesso da parte dell'utente al marketplace online Microsoft "Microsoft Windows Azure Marketplace" e all'utilizzo di esso, inclusi i servizi, i cataloghi, i dati, le applicazioni, il software, i siti, le informazioni, i materiali, il contenuto e gli aggiornamenti di Windows Azure Marketplace resi disponibili nell'ambito di tale marketplace ("Servizio Marketplace").

Qualora l'utente intenda utilizzare i Servizi Marketplace unitamente all'account Windows Azure (ad esempio, eseguendo software ricevuto da Windows Azure Marketplace nel proprio account Windows Azure), tale utilizzo verrà disciplinato anche dal Contratto dei Microsoft Online Services o dal Contratto Enterprise, a seconda dei casi, e dalle condizioni ivi contenute.

Selezionando la casella di controllo sottostante, l'utente dà atto di aver letto il presente Contratto e di accettarne le condizioni, incluse quelle relative al recapito elettronico delle comunicazioni e alle operazioni di divulgazione in relazione al Servizio Marketplace, di essere autorizzato a stipulare il presente Contratto e di utilizzare i Servizi Marketplace ESCLUSIVAMENTE PER FINI COMMERCIALI. Qualora l'utente non accetti o non sia in grado di accettare le presenti condizioni oppure non desideri stipulare il presente Contratto con Microsoft, è tenuto a chiudere la pagina Web e uscire dal sito.

- UTILIZZO DEL SERVIZIO MARKETPLACE.

◀ Precedente

☒ Accetto le condizioni per l'utilizzo

REGISTRA

FIGURA 4 - SCHERMATA DI ACCETTAZIONE DELLE CONDIZIONI

Dopo la registrazione, il sistema ci autenticcherà automaticamente al Marketplace facendo comparire il nostro nome nel menu in alto a destra.

Windows Azure Marketplace

Ciao Salvatore De Astis Italiano Italia Supporto Disconnetti

Inizi Applicazioni Dati Il mio account Pubblica Cerca in Marketplace

FIGURA 5 - MENÙ UTENTE AUTENTICATO

Adesso dobbiamo trovare le Microsoft Translator API; quindi scriviamo nella Search box 'Translator' e premiamo invio. Quando il sistema troverà le API le selezioniamo.

4 risultati in: CERCA: TRANSLATOR

Ordina per: Data di aggiunta Nome Editore 1

Microsoft® Translator

Microsoft Translator

pubblicato da: Microsoft Translator

Microsoft Translator delivers automatic translation (Machine Translation) of a text into a specified language. It is a state-of-the-art statistical machine translation system translating between any of the supported languages, and powering millions of translations every day.

dati

FIGURA 6 - RISULTATO DELLA RICERCA

Cliccando sul Microsoft Translator API (mostrato in figura 6), verremo portati ad una pagina contenente varie iscrizioni per utilizzare le API. Ci sono vari tipi di iscrizioni mensili; ciò che differenzia le offerte è nel numero di caratteri traducibili al mese. È disponibile anche un'iscrizione gratuita che permette di tradurre fino a 2.000.000 di caratteri/mese.

The screenshot shows the Windows Azure Marketplace interface. At the top, there's a navigation bar with the user's name 'Ciao Salvatore De Astis', language 'italiano', country 'Italia', and links for 'Supporto' and 'Disconnetti'. Below this, a secondary navigation bar includes 'Inizi', 'Applicazioni', 'Dati', 'Il mio account', and 'Pubblica'. A search bar on the right says 'Cerca in Marketplace'. The main content area is titled 'Microsoft Translator' and 'Dati'. It lists the publisher as 'Microsoft Translator', categories as 'Comunicazioni, Riferimenti, Servizi per sviluppatori, Elettronica e tecnologia', and the date added as '31/08/2011'. A description of the service is provided, along with a 'Like' button showing 659 likes. On the right, a table lists subscription options:

Caratteri al mese	Prezzo al mese	Botone
2.000.000	€ 0,00	ISCRIVITI
4.000.000	€ 29,89	ACQUISTA
6.000.000	€ 44,84	ACQUISTA
8.000.000	€ 59,78	ACQUISTA
16.000.000	€ 119,57	ACQUISTA
32.000.000	€ 229,59	ACQUISTA
64.000.000	€ 440,81	

At the bottom of the main content area, there are tabs for 'Immagini di esempio', 'Dettagli', and 'Condizioni offerta dell'editore'.

FIGURA 7 - SCHERMATA DI SCELTA DEL TIPO DI ISCRIZIONE AL SERVIZIO

Cliccando su bottone 'Iscriviti', del tipo di iscrizione scelto, verremo mandati alla pagina di riepilogo dell'iscrizione scelta e la data d'inizio della decorrenza del servizio. Una volta accettate le condizioni, il tasto 'Iscriviti' diventerà verde e cliccandoci su saremo iscritti al servizio.

The screenshot shows the 'Iscriviti' (Sign Up) page for the Microsoft Translator API. The header is identical to Figure 7. The main heading is 'Iscriviti'. Below it, the publisher is 'Microsoft Translator'. The selected offer is 'Sottoscrizione con 2.000.000 Caratteri' at a price of 'Gratuita, con decorrenza dal 10/06/2014'. A link to 'Visualizza condizioni offerta editore' is provided. A disclaimer states: 'Le offerte vengono concesse in licenza dall'editore in base alle specifiche condizioni definite dall'editore stesso. Microsoft non concede diritti né licenze per offerte di editori di terze parti.' A checkbox for 'Ho letto e accetto le condizioni dell'offerta e l'Informativa sulla privacy dell'editore precedentemente indicato.' is present. At the bottom left is an 'Annulla' button, and at the bottom right is a large green 'ISCRIVITI' button.

FIGURA 8 - ISCRIZIONE AL SERVIZIO

Una volta finita l'iscrizione, dobbiamo andare a registrare la nostra applicazione così da ottenere le nostre credenziali. Quindi andiamo alla pagina iniziale del Marketplace (quella di figura 1 accessibile

al seguente link <https://datamarket.azure.com/>), accediamo al nostro account, sempre che non lo siamo già, ed andiamo alla fine della pagina così da poter vedere dei link. Sotto la voce 'Sviluppo' troveremo la voce 'Registra l'applicazione creata'.

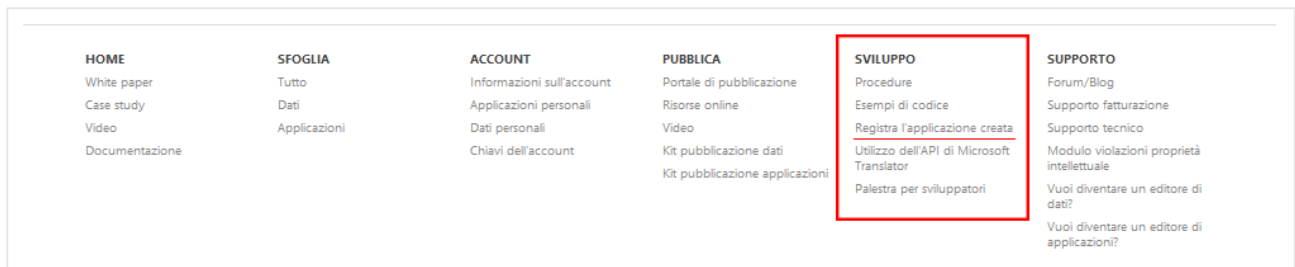


FIGURA 9 - MENÙ A FONDO PAGINA

Verremo portati alla pagina di registrazione della nostra applicazione; questa pagina consentirà di ottenere i valori del Client ID e il Client Secret necessari per autenticare il servizio quando si genera l'applicazione. La pagina di registrazione dell'applicazione sarà simile a questa:

Registra l'applicazione creata

ID cliente

ID dell'applicazione. Può essere composto da numeri, lettere, trattini e caratteri di sottolineatura.

Nome

Nome dell'applicazione visualizzato all'utente quando viene chiesta l'autorizzazione dell'utente per accedere all'account.

Segreto cliente

Password dell'applicazione. Tenerla segreta. È possibile fornire una password personale o utilizzare la password fornita sopra.

URI di

reindirizzamento

URL da reindirizzare alla fine del **flusso di consenso**. Per i client che non utilizzano il flusso di consenso, ad esempio Microsoft Translator, è necessario immettere un URL valido.

☐ **Abilita accesso sottodominio**

Consente di specificare i sottodomini per l'URI di reindirizzamento quando viene richiamato il flusso di consenso. Nella maggior parte degli scenari è consigliabile lasciare l'opzione deselezionata.

Descrizione

Descrizione facoltativa dell'applicazione, visibile solo all'utente.

◀ Annulla

CREA

FIGURA 10 - REGISTRAZIONE DELL'APPLICAZIONE

Vediamo cosa indicano nello specifico i campi presenti nella registrazione:

- ID Cliente: indica l'ID dell'applicazione.
- Nome: indica il nome dell'applicazione che viene visualizzato all'utente quando viene chiesta l'autorizzazione dell'utente per accedere all'account.
- Segreto cliente: è la password dell'applicazione; bisogna tenerla segreta. Questo campo viene riempito automaticamente e non deve essere cambiato.
- URI di re-indirizzamento: Indica URL da reindirizzare alla fine del flusso di consenso. Affinché un'applicazione possa accedere al Marketplace per conto di un utente, l'utente deve fornire il consenso per l'accesso dell'applicazione ai relativi account e sottoscrizioni. Se l'utente fornisce il consenso, l'applicazione ottiene un token di accesso che verrà utilizzato per eseguire l'autenticazione con i servizi di Marketplace per conto dell'utente. Microsoft Translator non utilizza il flusso di consenso, è necessario mettere un URL valido.
- Descrizione: descrizione facoltativa dell'applicazione che è visibile solo all'utente.

Una volta riempiti i vari campi, clicchiamo su '*Crea*' così che la nostra applicazione venga registrata. Saremo indirizzati alla pagina Sviluppatori, dove ci saranno tutte le applicazioni che abbiamo registrato. Per ogni applicazione verranno indicati ID cliente, il nome, la descrizione e lo stato. In più saranno presenti due link:

- Modifica: che serve per ritornare alla schermata di figura 10, così da poter modificare i campi Nome, Segreto cliente, URI di re-indirizzamento e Descrizione.
- Elimina: per eliminare l'applicazione registrata.

Il tasto registra serve per poter registrare una nuova applicazione.

Sviluppatori

Windows Azure™ Marketplace consente agli sviluppatori di individuare, acquistare e gestire con facilità sottoscrizioni a dati di domini pubblici attendibili e dati commerciali premium attraverso un modello comune di sicurezza, fatturazione, controllo e autenticazione. È possibile utilizzare facilmente le informazioni tramite un'unica API REST coerente per tutti i set di dati e/o sfruttare appieno i vantaggi offerti della vendita tramite Microsoft per migliorare il ROI e acquisire nuovi clienti vendendo l'applicazione creata in Marketplace.

Ulteriori informazioni

- Utilizzo dei dati di Marketplace nell'applicazione creata
- Sviluppo di un'applicazione da vendere in Marketplace
- Sviluppo di un'applicazione che utilizza il servizio Microsoft Translator
- Estrazione degli ultimi esempi di codice di Marketplace

Applicazioni registrate

ID cliente	Nome	Descrizione	Stato	
DeAstisMicrosoftTraduttore	Microsoft Traduttore	Traduttore che utilizza Microsoft Translator API	Attivo	Modifica Elimina

REGISTRA

FIGURA 11 - SCHERMATA DELLE APPLICAZIONI REGistrate

Adesso possiamo usare queste credenziali per far sì che la nostra applicazione possa utilizzare le API di Microsoft Translator.

Implementazione dell'applicazione 'Microsoft Traduttore'

Per semplificare la progettazione del traduttore, si è deciso di separare le varie funzionalità in moduli, così da poterli richiamare nella pagina principale dell'applicazione. In tutto sono stati creati sei moduli, ovvero:

- Credentials.php;
- AccessTokenAuthentication.php;
- HttpTranslator.php;
- functions.js;
- css.css;
- index.php.

Vediamo nel dettaglio ogni modulo.

Modulo 'Credentials.php'

Il modulo *Credentials.php* è il modulo del progetto in cui sono contenute le credenziali di accesso per l'utilizzo delle API. Ricordo che le credenziali ottenute durante la fase di registrazione sono: Client ID, Client Secret, OAuth URL, Application scope URL e l'application grant.

Il valore di ognuna di queste credenziali è stato salvato in una variabile PHP; si devono avvalorare queste variabili con i valori presenti nella schermata di registrazione dell'applicazione, vista in figura 10.

Vediamo come si presenta il modulo:

```
1. <?php
2.
3.     $clientID = "DeAstisMicrosoftTraduttore2014";
4.     $clientSecret = "Vfx6Ihf8PoK8OEFx0ZFPgChvTR+N5pD2BLqWSe+Fp7k=";
5.     $authUrl = "https://datamarket.accesscontrol.windows.net/v2/OAuth2-13/";
6.     $scopeUrl = "http://api.microsofttranslator.com";
7.     $grantType = "client_credentials";
8.
9. ?>
```

MODULO - CREDENTIALS.PHP

Come si vede dal codice, questo modulo è molto semplice, infatti è presente solo una dichiarazione di variabili a cui sono state associati i relativi valori. Si è deciso di mettere queste variabili in un modulo a parte così da semplificare un'eventuale modifica di questi valori, ma al contempo, permette di non inserire queste credenziali all'interno della pagina principale dell'applicazione. In particolar modo i valori delle variabili '\$clientID' e '\$clientSecret' sono molto importanti perché sono dati in modo confidenziale all'autore dell'applicazione e devono rimanere segreti rispetto all'utilizzatore dell'applicazione.

Modulo 'AccessTokenAuthentication.php'

L'API di Microsoft Translator per essere utilizzato, richiede l'autenticazione al Windows Azure Marketplace. In questo modulo si va a creare la classe `AccessTokenAuthentication` in cui si va ad implementare la funzione di generazione del token di accesso, che sarà mandato al servizio Microsoft Translator così da autorizzare la traduzione del testo o della parola. Vediamo il codice di questo modulo:

```
1. <?php
2. class AccessTokenAuthentication
3. {
4.
5.     function getTokens($grantType, $scopeUrl, $clientId, $clientSecret, $authUrl)
6.     {
7.         try
8.         {
9.             $ch = curl_init();
10.            $paramArr = array ('grant type'=>$grantType, 'scope'=>$scopeUrl,
'client id'=>$clientId, 'client secret'=>$clientSecret);
11.
12.            $paramArr = http_build_query($paramArr);
13.            curl_setopt($ch, CURLOPT_URL, $authUrl);
14.            curl_setopt($ch, CURLOPT_POST, TRUE);
15.            curl_setopt($ch, CURLOPT_POSTFIELDS, $paramArr);
16.            curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);
17.            curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
18.
19.            $strResponse = curl_exec($ch);
20.
21.            $curlErrno = curl_errno($ch);
22.            if($curlErrno)
23.            {
24.                $curlError = curl_error($ch);
25.                throw new Exception($curlError);
26.            }
27.
28.            curl_close($ch);
29.
30.            $objResponse = json_decode($strResponse);
31.            if ($objResponse -> error)
32.            {
33.                throw new Exception($objResponse -> error description);
34.            }
35.            return $objResponse -> access_token;
36.
37.        } catch (Exception $e)
38.        {
39.            echo "Exception-".$e -> getMessage();
40.
41.        }
42.    }
43. }
```

MODULO - ACCESSTOKENAUTHORIZATION.PHP

Dopo aver definito il nome della classe, si va a dichiarare la funzione *getTokens* che prende in input il valore delle credenziali inseriti nel modulo '*Credentials.php*'.

Nella funzione viene, prima di tutto, inizializzata una sessione *cURL* utilizzando il comando *curl_init()*. *cURL* è un comando che appartiene alla libreria **libcurl**, libreria free e facile da usare da lato client, che permette il trasferimento dell'URL. Supporta famosi servizi come http, https, file, gopher, pop3,

ecc. Inoltre supporta i certificati SSL, http POST, http PUT, proxy, autenticazione con nome utente e password (Digest, Kerberos, ecc.).

Successivamente si va ad inizializzare un array contenente le credenziali di accesso passate come input alla funzione. Il contenuto dell'array verrà utilizzato dalla funzione *http_build_query()* che consentirà di generare una query http. Dopo aver impostato le opzioni per il trasferimento del cURL, utilizzando la funzione *curl_setopt*, il cURL viene eseguito. Dopo aver controllato che l'esecuzione sia avvenuta senza errore, si va a chiudere la sessione cURL (chiamando la funzione *curl_close()*) ed infine si va a decodificare la stringa JSON ricevuta. Viene effettuato un'ulteriore controllo, questa volta sulla stringa JSON, dove in caso di errore viene sollevata un'eccezione altrimenti verrà restituito il token di accesso.

Modulo 'HttpTranslator.php'

Nel modulo è presente l'altra classe helper che consente di gestire la comunicazione con le API al nostro posto. Viene utilizzata la funzionalità cURL di PHP così può comunicare in maniera sincrona con un servizio web di tipo HTTP.

Il codice di questo modulo è il seguente:

```
1. <?php
2.
3. Class HTTPTranslator
4. {
5.     function curlRequest($url, $authHeader)
6.     {
7.         $ch = curl_init();
8.
9.         curl_setopt ($ch, CURLOPT_URL, $url);
10.        curl_setopt ($ch, CURLOPT_HTTPHEADER, array($authHeader,"Content-Type:
text/xml"));
11.        curl_setopt ($ch, CURLOPT_RETURNTRANSFER, TRUE);
12.        curl_setopt ($ch, CURLOPT_SSL_VERIFYPEER, False);
13.        $curlResponse = curl_exec($ch);
14.
15.        $curlErrno = curl_errno($ch);
16.
17.        if ($curlErrno)
18.        {
19.            $curlError = curl_error($ch);
20.            throw new Exception($curlError);
21.        }
22.
23.        curl_close($ch);
24.        return $curlResponse;
25.    }
26. }
27.
28. ?>
```

MODULO - HTTPTRANSLATOR.PHP

In generale la funzione prende come parametri di input il request URL, il request header e i dati mandati attraverso il metodo POST dal form e restituisce il cURL di risposta o un errore.

Una volta implementata anche questa funzione abbiamo tutte le funzioni necessarie per accedere al servizio di traduzione.

Modulo 'index.php'

Possiamo considerare questo modulo come la pagina principale della nostra applicazione, in quanto contiene non solo gli elementi che consentono di inserire il testo da tradurre e di scegliere la lingua di origine del testo e la lingua in cui vogliamo che il nostro testo sia tradotto, ma contiene anche il comportamento dei due bottoni principali: 'Traduci' e 'Ascolta'. Quest'ultimo bottone è presente nella pagina con il simbolo di un megafono vicino al bottone 'Traduci'.

Nella modulo sono presenti due sezioni principali:

- 1) la prima è la sezione **PHP**, indicata con i tag `<?php ?>`, in cui è presente tutto il codice necessario per effettuare la traduzione e la generazione dello stream audio per il text-to-speech.
- 2) la seconda sezione comprende tutto il codice **HTML**, indicato con i tag `<html> </html>`, necessario per creare gli oggetti presenti nella pagina, come le textarea, select box e i vari bottoni.

Vediamo nel dettaglio le due sezioni.

Il codice della parte PHP del modulo è il seguente:

```
1. <?php
2.
3. session_start();

4. if (glob('./speech file/*.mp3'))
5. {
6.     $file = glob('speech_file/'.'*.mp3', GLOB_MARK);
7.
8.     foreach ($file as $file)
9.     {
10.         if (is_dir($file))
11.         {
12.             self::deleteDir($file);
13.         }
14.         else
15.         {
16.             unlink($file);
17.         }
18.     }
19. }

20. include 'HttpTranslator.php';
21. include 'AccessTokenAuthentication.php';
22. require 'credential.php';

23. if (isset($_POST['traduci_sub']))
24. {
25.     if (isset($_POST['source_lang']))
26.     {
27.         $_SESSION['source_lang'] = $_POST['source_lang'];
28.     }

29.     if (isset($_POST['dest_lang']))
30.     {
31.         $_SESSION['dest_lang'] = $_POST['dest_lang'];
32.     }
33.
34.     try
35.     {
36.         $authObj = new AccessTokenAuthentication();
```

```

37.     $accessToken = $authObj -> getTokens($grantType, $scopeUrl, $clientId,
        $clientSecret, $authUrl);
        a. $authHeader = "Authorization: Bearer ". $accessToken;

        b. $fromLanguage = $_POST["source_lang"];
        c. $toLanguage = $_POST["dest_lang"];
        d. $inputStr = $_POST["source_text"];

        e. $contentType = 'text/plain';
        f. $category = 'general';

        g. $paramst = "text=".urlencode($inputStr)."&to=".$toLanguage."&from=".$fromLanguage;
        h. $translateUrl = "http://api.microsofttranslator.com/v2/Http.svc/Translate?$paramst";

        i. $translatorObj = new HTTPTranslator();

        j. $curlResponse = $translatorObj -> curlRequest($translateUrl, $authHeader);

        k. $xmlObj = simplexml_load_string($curlResponse);
        l. foreach((array)$xmlObj[0] as $val)
        m. {
            i. $translatedStr = $val;
        n. }

        o. $translatedText = urlencode($translatedStr);
        p. $out = 'audio/mp3';
        q. $params = "text=$translatedText&language=$toLanguage&format=$out";

        r. $url = "http://api.microsofttranslator.com/V2/Http.svc/Speak?$params";

        s. $translatorObj = new HTTPTranslator();

        t. $strResponse = $translatorObj -> curlRequest($url, $authHeader);

        u. if (!is_dir('speech_file'))
        v. {
            i. mkdir('speech file');
        w. }

        x. $var = uniqid('SPC_').".mp3";

        y. file_put_contents('./speech file/'.$var, $strResponse);

38. } catch (Exception $e)
39. {
        a. echo "Exception: " . $e->getMessage() . PHP_EOL;
40. }
41. }
42.
43. ?>

```

MODULO - INDEX.PHP (PARTE PHP)

Per poter utilizzare le API per effettuare la traduzione e il text-to-speech, la richiesta avviene tramite il passaggio del testo e della lingua di origine e di destinazione. Inizialmente viene effettuato un controllo sulla presenza di precedenti file audio. Nel caso in cui sia presente già un file, esso viene eliminato, altrimenti si prosegue andando ad includere i moduli creati precedentemente. Successivamente vengono salvati in delle variabili di sessione le due lingue selezionate; questo viene fatto per permettere all'applicazione di mantenere la scelta delle lingue effettuata dall'utente anche dopo il refresh della pagina.

Arrivati a questo punto inizia la procedura per poter tradurre il testo: il tutto viene eseguito in un *try-catch* così che nel caso in cui ci sia un errore, esso venga riportato a schermo. Si inizia con l'ottenere il permesso per utilizzare le API ottenendo un token. Successivamente attraverso il

passaggio dei valori del testo e delle lingue si vanno a settare i parametri che saranno aggiunti all'URL per poter eseguire la traduzione. La traduzione avviene utilizzando il metodo *'Translate'*, invocato attraverso l'URL creato precedentemente. Una volta ottenuta la traduzione, in formato XML, esso viene interpretato in modo che diventi un oggetto che possa essere usato nell'applicazione. Il risultato della traduzione viene salvato nella variabile *\$translatedText*.

Dopo aver ottenuto il testo tradotto, viene anche generato il text-to-speech. Anche qui si vanno a settare delle variabili così che si possa costruire l'URL di richiesta. Il metodo invocato per ottenere il text-to-speech è lo *'Speak'* e che richiede il settaggio dei seguenti parametri:

- il testo tradotto;
- la lingua in cui il testo è stato tradotto;
- il formato in cui restituire lo stream audio.

Per quanto riguarda il formato dello stream audio, nell'implementazione è stato scelto *'mp3'*, ma API mette a disposizione anche il formato *'wav'*; nel caso in cui il formato non venisse specificato, l'API utilizza il formato *'wav'*. Un altro parametro che si può settare riguarda la qualità dell'audio, non utilizzato nella implementazione di questo traduttore.

Una volta che si è ottenuto lo stream audio, esso viene memorizzato nella cartella *'speech_file'*, la quale viene creata nel caso in cui non fosse già presente. Al file viene assegnato ogni volta un nome univoco, ottenuto grazie all'istruzione *'uniqid('SPC_')."mp3"'*, e viene salvato nella directory utilizzando la funzione *'file_put_contents('./speech_file/.\$var, \$strResponse)'* dove il primo argomento indica la cartella dove salvare il file, mentre il secondo argomento indica il file.

Salvato il file audio, viene chiuso il try-catch, ponendo fine allo script PHP.

Per quanto riguarda la sezione **HTML** non ha bisogno di particolari spiegazioni, in quanto contiene il codice necessario per la creazione degli oggetti del form come textarea, select box e i vari bottoni. L'unica parte del codice che potrebbe essere necessario spiegare è quello riguardante le select box.

Come già specificato nelle select box sono presenti le lingue disponibili per la traduzione (quelle presenti nell'applicazione non sono tutte quelle disponibili, l'elenco con tutte le lingue messe a disposizione dell'API è disponibile al seguente indirizzo <http://msdn.microsoft.com/en-us/library/hh456380.aspx>). Per non creare delle select box con un numero elevato di tag *'<option>'* e per non duplicare il codice HTML per riempire le select box, si è deciso di dichiarare un array contenente le lingue.

L'array è stato settato in modo tale che l'indice di ogni elemento corrispondesse al codice della lingua utilizzato dall'API. Così facendo il valore che viene passato allo script PHP, spiegato in precedenza, può essere direttamente inserito nell'URL senza che sia richiesto il ritrovamento del codice della lingua attraverso una richiesta all'API.

Dichiarato l'array, nelle select è stato necessario inserire del codice PHP sia per poter accedere all'array e visualizzarne i valori sia per far in modo che, se l'utente ha già fatto una traduzione, vengano ricordate le scelte fatte in precedenza. Inoltre è possibile anche scegliere le altre lingue per traduzioni successive.

Questo comportamento è stato implementato tramite condizioni in PHP e attraverso l'utilizzo delle variabili di sessione: all'inizio viene controllato se la variabile di sessione della lingua sia stata settata o meno (viene usato il metodo *'isset'* messo a disposizione del PHP). Nel caso in cui non sia stata settata, allora si devono mostrare all'utente tutte le lingue disponibili nell'array delle lingue, altrimenti viene fatta una ricerca nel vettore finché non venga trovata la lingua che abbia il codice uguale a quello presente nella variabile di sessione. Siccome l'utente potrebbe decidere di voler cambiare lingua, è stato aggiunto il codice in modo da mostrare anche le altre lingue una volta che apre il menu a tendina della select box.

```

1. <!DOCTYPE html>
2. <html lang = "en">
3.
4. <head>
5.     <meta charset="utf-8" />
6.     <title>Microsoft Translator</title>
7.     <link rel = "stylesheet" href = "css.css" />
8.     <script type = "text/javascript" src = "functions_js.js"></script>
9. </head>
10.
11.
12. <body>
13.
14.     <?php
15.         $languages = array("it"=>"Italian", "en"=>"English", "ja"=>"Japanese",
16.         "pt"=>"Portuguese", "es"=>"Spanish", "tr"=>"Turkish",
17.         "zh-CHS"=>"Simplified Chinese", "zh-CHT"=>"Traditional Chinese",
18.         "de"=>"German", "fr"=>"French",);
19.
20.         asort($languages);
21.     <?>
22.     <div class = "body">
23.
24.         <div class = "header">
25.             <img class = "logo" src = "image/logo.png" width = "300" height = "47" alt
26.             = "logo"/>
27.         </div>
28.
29.         <form id = "trad" name = 'trad' action = "index.php" method = "POST">
30.
31.             <select class = "source lang" name = "source lang">
32.
33.                 <?php if (isset($_SESSION['source_lang']) == false)
34.                 {?>
35.                     <option value =
36.                     "Select lang">Select language</option>
37.                     <?php foreach($languages as $code => $lang)
38.                     { ?>
39.                         <option value = "<?php echo $code; ?>" <?php echo
40.                         $lang; ?> </option>
41.                     }
42.                     <?php }
43.                     }
44.                     else
45.                     {
46.                         foreach($languages as $code => $lang)
47.                         { if ($code == $_SESSION['source_lang'])
48.                         {?>
49.                             <option value = "<?php echo $code; ?>" <?php echo
50.                             $lang; ?> </option>
51.                         }
52.                         <?php }
53.                         }
54.                         foreach($languages as $code => $lang)
55.                         {?>
56.                             <option value = "<?php echo $code; ?>" <?php echo
57.                             $lang; ?> </option>
58.                         }
59.                         <?php }
60.                     }
61.                 }
62.             }
63.         }
64.     }
65. </div>
66. </body>
67. </html>

```

```

54.             } ?>
55.
56.         </select>
57.
58.
59.         <textarea id = "source_text" class = "source_text" name = "source_text"
onclick = "javascript:select(source_text)" onkeypress =
"javascript:send('translate sub', event);"><?php if (isset($inputStr)== true){echo
$inputStr;}else{echo ' ';} ?></textarea>
60.
61.         <select id = "dest_lang" class = "dest_lang" name = "dest_lang">
62.
63.
64.             <?php if (isset($_SESSION['source_lang']) == false)
65.                 {?>
66.                                     <option value =
"Select_lang">Select language</option>
67.                 <?php foreach($languages as $code => $lang)
68.                     { ?>
69.                         <option value = "<?php echo $code; ?>"> <?php echo
$lang; ?> </option>
70.                 <?php }
71.                     }
72.                 else
73.                 {
74.                     foreach($languages as $code => $lang)
75.                     { if ($code == $_SESSION['dest_lang'])
76.                         { ?>
77.                             <option value = "<?php echo $code; ?>"> <?php echo
$lang; ?> </option>
78.                 <?php }
79.                     }
80.                 }
81.                 foreach($languages as $code => $lang)
82.                 { ?>
83.                     <option value = "<?php echo $code; ?>"> <?php echo
$lang; ?> </option>
84.                 <?php }
85.                 }?>
86.
87.         </select>
88.
89.
90.         <input id = "translate_sub" class = "translate_sub" name ="translate_sub"
type = "submit" value = "Translate">
91.
92.     </form>
93.
94.
95.     <textarea id = "transl_text" class = "transl_text" name = "transl_text"
onclick = "javascript:select(transl_text)"><?php if (isset($inputStr)== true){echo
$translatedStr;}else{echo ' ';} ?></textarea>
96.
97.
98.
99.
100.    <button class = "reset" onclick = "javascript:trad.source_text.value = ' ';
javascript:transl_text.value = ' '">Reset</button>
101.
102.
103.
104.    <?php foreach(glob('./speech_file/*.*) as $filename)
105.        {
106.            $speech = $filename;
107.        }
108.    ?>
109.
110.    <audio id = "play_speech">
111.        <source src = "<?php echo $speech; ?>" type = "audio/mpeg" />
112.    </audio>
113.
114.
115.    <button id = "t2s" class = "t2s" type = "button" onclick =
"speech_play()"></button>

```

```
116.
117.         <div class = "pagef">
118.             <img class = "credit" src = "image/credit.png" width = "200" height =
119.             "25" alt = "credit"/>
120.         </div>
121.     </div>
122.
123. </body>
124. </html>
```

MODULO - INDEX.PHP (PARTE HTML)

Una volta premuto il tasto *Traduci* sotto la textarea dove compare il testo tradotto, il tasto *Ascolta* (che ha il simbolo di un megafono) può essere premuto in modo tale che l'utente possa ascoltare il testo tradotto.

Il tasto *Reset* ha il compito di pulire le text area da ciò che contiene; questo comportamento è stato ottenuto utilizzando una funzione Javascript che si attiva ogni volta che il tasto viene premuto. La funzione è molto semplice in quanto imposta il valore della textarea come vuoto; il valore vuoto nella funzione viene impostato tramite '' (apici senza nessun contenuto all'interno di esse).

Modulo 'functions_js.js'

Il modulo functions_js.js è un modulo contenente due funzioni Javascript che vengono utilizzate nel modulo *index.php*, in cui viene incluso. Le funzioni che sono presenti nel modulo sono:

- select: questa funzione permette di selezionare il testo presente nella textarea quando si clicca su quest'ultima.
- send: questa funzione permette di inviare il testo da tradurre, quindi di spingere il pulsante *Traduci* premendo il tasto *Invio* (*Enter* nelle tastiere inglesi/americane) della tastiera.

Il modulo si presenta in questa maniera:

```
1.  function select(text)
2.  {
3.      var t = document.getElementById(text).innerHTML;
4.      alert(t);
5.  }
6.
7.
8.  function send(t, e)
9.  {
10.
11.     if ((e.which && e.which == 13) || (ev.keyCode && e.keyCode == 13))
12.     {
13.         document.getElementById(t).click();
14.         return false;
15.     }
16.     else
17.     {
18.         return true;
19.     }
20. }
21.
22. function speech_play()
23. {
24.     var play_speech = document.getElementById("play_speech");
25.     if (play_speech.paused)
26.     {
27.         play_speech.play();
```

```
28.     }
29.     else
30.     {
31.         play speech.pause();
32.     }
33. }
```

MODULO – FUNCTIONS_JS.JS

Modulo 'css.css'

Questo è il foglio di stile in cui sono presenti tutte le direttive per la formattazione dei componenti dell'applicazione. Questo file viene incluso nel modulo 'index.php'.

```
1. @charset "utf-8";
2. /* CSS Document */
3.
4. @font-face
5. {
6.     font-family: 'Segoe UI Light';
7.     src: url('font/segoe ui light.eot');
8.     src: url('font/segoe ui light.eot?#iefix') format('embedded-opentype'),
9.     url('font/segoe ui light.svg#Segoe UI Light') format('svg'),
10.    url('font/segoe ui light.woff') format('woff'),
11.    url('font/segoe ui light.ttf') format('truetype');
12.    font-weight: normal;
13.    font-style: normal;
14. }
15.
16. div.header
17. {
18.    position: absolute;
19.    width: 100%;
20.    height: 10%;
21.    top: 0%;
22.    left: 0%;
23.    background-color: #eeeeec;
24. }
25.
26. /* Setting for logo */
27. img.logo
28. {
29.    position: absolute;
30.    top: 6px;
31. }
32.
33.
34. select.source_lang
35. {
36.    position: absolute;
37.    top: 23%;
38.    width: 146px;
39. }
40.
41. select.dest_lang
42. {
43.    position: absolute;
44.    top: 23%;
45.    left: 50%;
46.    width: 146px;
47. }
48.
49.
50. select
51. {
52.    font-size: 14px;
53.    color: #333;
54.    background: #CCC;
55.    font-family: Segoe UI;
56.    border: solid 2px;
```

```

57.   border-color:#999;
58. }
59.
60.
61. button.reset
62. {
63.     position: absolute;
64.     top: 60%;
65.     left: 8px;
66.     font-size: 14px;
67.     color: #333;
68.     background: #CCC;
69.     font-family: Segoe UI;
70.     border: solid 2px;
71.     border-color:#999;
72. }
73.
74. input.translate_sub
75. {
76.     position: absolute;
77.     top: 23%;
78.     left: 65%;
79.     width: 100px;
80.     font-size: 14px;
81.     color: #fff;
82.     background: #0094ff;
83.     font-family: Segoe UI;
84.     border: solid 2px;
85.     border-color:#0082ff;
86. }
87.
88.
89. /* Setting position of the text-to-speech player
90. audio.t2s
91. {
92.     position: absolute;
93.     top: 57%;
94.     left: 50.5%;
95. }*/
96.
97. button.t2s
98. {
99.     width: 26px;
100.    height: 26px;
101.    display: inline-block;
102.    top: 23%;
103.    left: 75%;
104.    position: absolute;
105.    cursor: pointer;
106.    border: none;
107.    background: transparent;
108.    background-image: url("../image/speech.png");
109.    box-shadow: initial ;
110. }
111.
112.
113. button.t2s:active
114. {
115.     top: 22.5%;
116.     box-shadow: initial;
117. }
118.
119.
120. textarea.source text
121. {
122.     position:absolute;
123.     top: 30%;
124.     font-family: 'Segoe UI';
125. }
126.
127. textarea.transl text
128. {
129.     position:absolute;
130.     top: 30%;

```

```
131.     left: 50%;
132.     font-family: 'Segoe UI';
133. }
134.
135.
136. textarea
137. {
138.     font-size: 14px;
139.     width: 440px;
140.     height: 200px;
141.     border:solid 2px;
142.     border-color: #999;
143.     resize: none;
144.
145. }
146.
147.
148.
149. /* Setting for page footer */
150. div.pagef
151. {
152.     position: absolute;
153.     width: 100%;
154.     height: 10%;
155.     bottom: 0;
156.     left: 0;
157.     background-color: #eeecec;
158. }
159.
160. /* Setting for credit */
161. img.credit
162. {
163.     position: absolute;
164.     bottom: 16px;
165.     left: 80%;
166. }
```

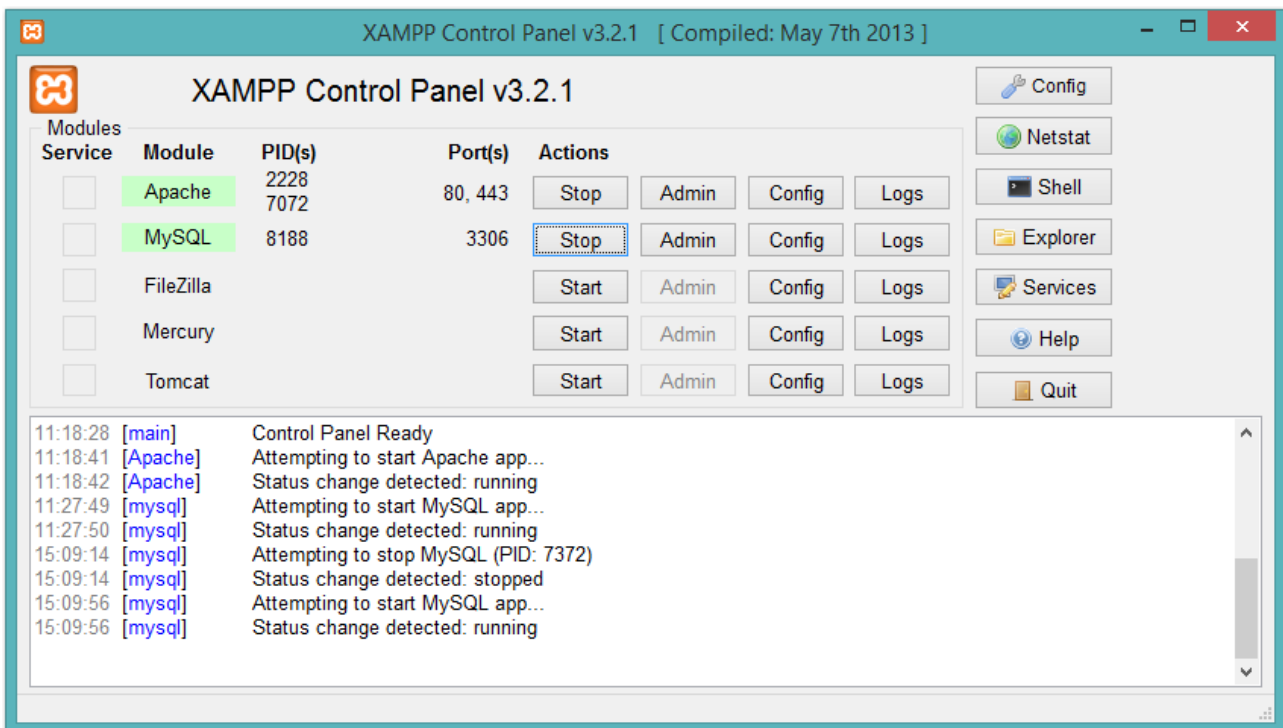
MODULO – CSS.CSS

Installazione dell'applicazione in XAMPP su piattaforma Windows

È possibile utilizzare l'applicazione utilizzando il server web XAMPP. XAMPP è una piattaforma software gratuita costituita da Apache http Server, MySQL e da tutti gli altri strumenti necessari per utilizzare linguaggi di programmazione come PHP e Perl.

Per poter utilizzare l'applicazione, basta installare XAMPP, disponibile al seguente indirizzo <https://www.apachefriends.org/index.html>, una volta installata, basta inserire la cartella dell'applicazione nella cartella 'htdocs' di XAMPP raggiungibile al seguente percorso 'C:/xampp/'. Si noti che il percorso potrebbe cambiare se durante la fase di installazione del web server si è scelto un percorso differente.

Per poter avviare l'applicazione dobbiamo attivare il servizio Apache e MySQL, andando ad utilizzare l'interfaccia di XAMPP. Il servizio sarà attivo se la scritta del servizio sarà evidenziata di verde come mostrato in figura.



ATTIVAZIONE SERVIZIO APACHE IN XAMPP

Una volta attivati i servizi, dobbiamo verificare XAMPP sia realmente attivo; per far questo dobbiamo aprire il browser e digitare l'indirizzo 'http://localhost/'. Se il tutto funziona dovrebbe apparire la schermata di benvenuto come la seguente (si noti che la schermata potrebbe cambiare in base alla versione).



XAMPP for Windows

English / Deutsch / Français / Nederlands / Polski / **Italiano** / Norwegian / Español / 中文 / Português (Brasil) / 日本語

XAMPP

1.8.3

[PHP: 5.5.11]

Benvenuto

Stato

Sicurezza

Documentazione

Componenti

Applications

Php

phpinfo()

Collezione CD

Bioritmo

Grafica istantanea

Rubrica

Perl

perlinfo()

Guest Book

J2ee

Info

Tomcat examples

Tools

phpMyAdmin

FileZilla FTP

Webalizer

Mail

©2002-2014

Benvenuto a XAMPP per Windows!

Congratulazioni:

Hai installato XAMPP con successo!

Ora puoi iniziare ad utilizzare Apache & Co. Innanzitutto dovresti andare su »Stato« sul menu di navigazione a sinistra per essere certo che tutto funzioni correttamente.

Buon lavoro, Kay Vogelgesang + Kai 'Oswald' Seidler

Install applications on XAMPP using BitNami

Apache Friends and BitNami are cooperating to make dozens of open source applications available on XAMPP, for free. BitNami-packaged applications include Wordpress, Drupal, Joomla! and dozens of others and can be deployed with one-click installers. Visit the [BitNami XAMPP page](#) for details on the currently available apps.



SCHERMATA INIZIALE DI XAMPP

Ora per accedere all'applicazione basta inserire il percorso nella barra dell'indirizzo; quindi basta aggiungere all'URL il nome della cartella contenente il nome del progetto e successivamente il nome della pagina principale del progetto, che è *'index.php'*. Quindi l'URL diventa: *'http://localhost/speech-translation-tools/index.php'*. Vediamo come appare la schermata iniziale dell'applicazione:

Microsoft Traduttore

Select language ▼

Select language ▼

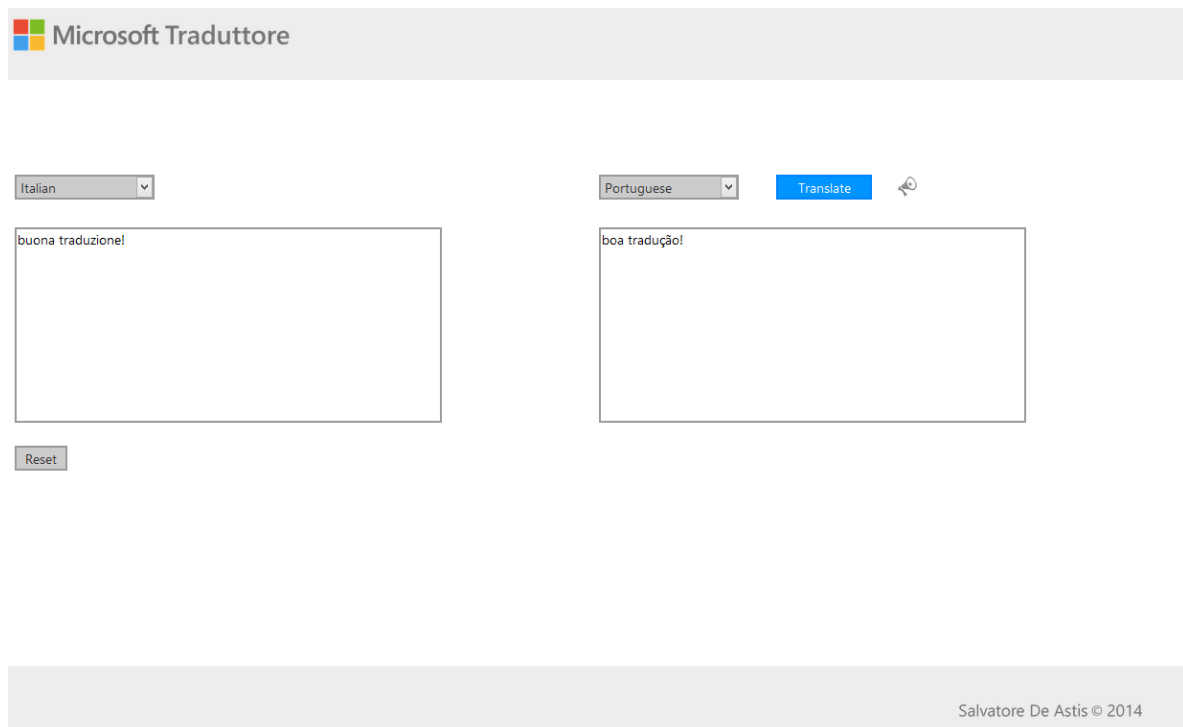
Translate

Reset

Salvatore De Astis © 2014

PAGINA INIZIALE DEL TRADUTTORE

Adesso l'utente può inserire il testo che vuole tradurre nella textarea presente a sinistra dello schermo, scegliere la lingua di origine e quella di destinazione e premere sul pulsante *'Traduci'*. Alla fine verrà mostrato nella textarea a destra il testo tradotto e comparirà il player per sentire il text-to-speech del testo tradotto.



ESEMPIO DI TRADUZIONE

Installazione dell'applicazione in XAMPP su piattaforma Linux

Per quanto riguarda l'installazione dell'applicazione su piattaforma linux (si è testato l'applicazione su Ubuntu 14.04 a 64 bit) il procedimento non cambia di molto. Innanzitutto bisogna scaricare XAMPP; una volta scaricato dobbiamo cambiare i permessi al file utilizzando il seguente comando:

```
sudo chmod 755 xampp-linux-*-installer.run
```

e avviare l'installazione utilizzando questo comando:

```
sudo ./xampp-linux-*-installer.run
```

Una volta finita la procedura, XAMPP sarà disponibile nella seguente directory *'/opt/lampp'*. Dopo bisogna attivare i servizi Apache e MySQL; questo viene fatto utilizzando il comando:

```
sudo /opt/lampp/lampp start
```

Quando eseguiremo il comando, nel terminale comparirà un messaggio simile al seguente:

```
Starting XAMPP for Linux 1.8.1...
XAMPP: Starting Apache with SSL (and PHP5)...
XAMPP: Starting MySQL...
XAMPP: Starting ProFTPD...
```

XAMPP for Linux started.

Adesso per verificare che i servizi siano effettivamente avviati, basta aprire il browser e digitare nella barra degli indirizzi `'http://localhost/'`. Se tutto è andato a buon fine apparirà la schermata principale di XAMPP. Adesso inserendo la cartella dell'applicazione nel percorso `'/opt/lampp'` e digitando nella barra degli indirizzi `'http://localhost/speech-translation-tools/index.php'` si potrà utilizzare l'applicazione.

Se si volessero stoppare i servizi di XAMPP, basterà digitare nel terminale il comando:

```
sudo /opt/lampp/lampp stop
```

Buona traduzione!