TECHNOMILE

TechnoMile Magento Salesforce Connect Installation Guide

Version 0.1.0

Table of Contents

1. Magento Salesforce Connect

Magento is a Enterprise level ecommerce system which has several ecommerce features like Customers management, Orders management, Inventory management, Catalog management, Coupons and promotions, Newsletters and Reports etc. whereas Salesforce is CRM and has the capabilities to manage the customer interaction, call logs, marketing campaigns etc.

TechnoMile Magento Salesforce Connect is an extension that's developed under the TechnoMile Development which helps to unite the Salesforce and Magento with generic feature which are needed for any business who works on CRM and Ecommerce independently.

TechnoMile Magento Salesforce Connect sync the data from Magento to SalesForce and vice versa.

2. Use Cases

- Customer has Magento site & Salesforce Org (Follow steps 3 - 8)
- Customer has Magento site but has no Salesforce Org (Follow steps 3 - 8)
- Customer has no Magento site but has Salesforce Org (Follow steps 3 - 9)
- Customer has no Magento site & Salesforce Org (Follow steps 3 - 9)

3. Magento Package Code Installation

1. Unzip the downloaded file (TechnoMile_Salesforce-0.1.0.tgz)
2. Read the installation document present in the Docs folder.
3. Once the required zip file is downloaded and unzipped, all related files would be available in your local machine.
4. Now it is required to follow instructions 4.1-4.5 mentioned below to copy the necessary files from your local machine to your Magento instance in the server box.

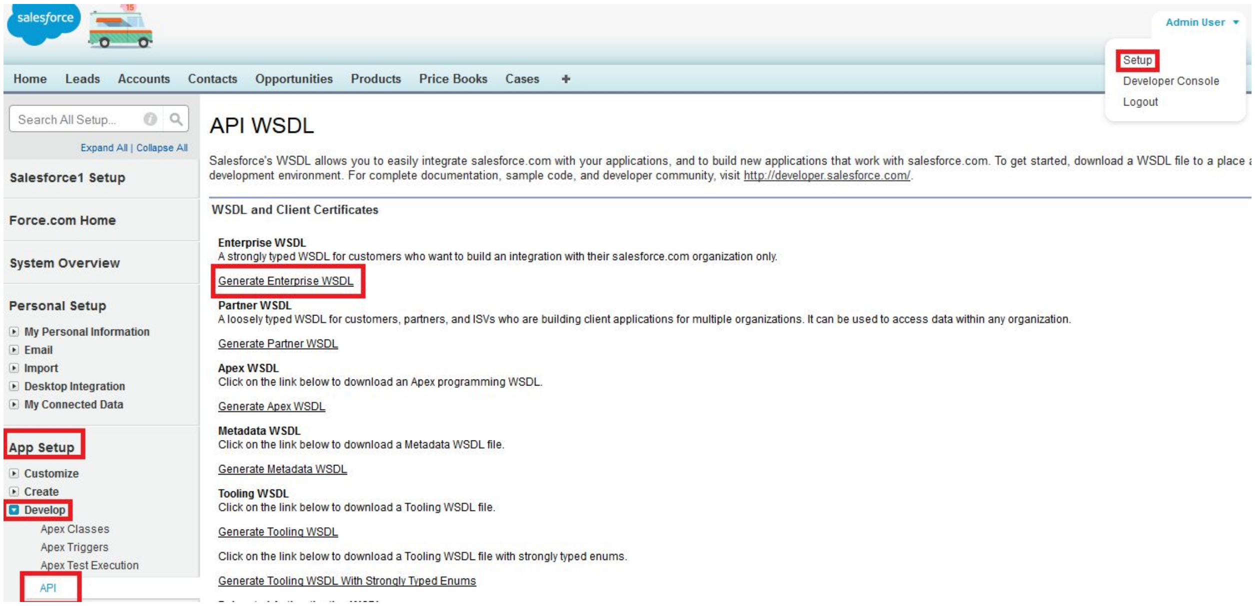
1. Navigate to **app\code\community** and Place **TechnoMile** folder.
2. Navigate to **app\design\frontend** and Place **salesforce** folder.
3. Navigate to **app\etc\modules** and Place **TechnoMile_Salesforce.xml** file.
4. Navigate to **lib** and Place **soapclient** folder.
5. Navigate to **skin\frontend** and Place **salesforce** folder.

4. Generate Token Key in Salesforce

Note: To generate enterprise WSDL file follow the below Process

Generate Enterprise WSDL file

- Go to Setup **>>** Develop **>>** API click on the link Generate Enterprise WSDL



API WSDL

Salesforce's WSDL allows you to easily integrate salesforce.com with your applications, and to build new applications that work with salesforce.com. To get started, download a WSDL file to a place a development environment. For complete documentation, sample code, and developer community, visit http://developer.salesforce.com/.

WSDL and Client Certificates

Enterprise WSDL
A strongly typed WSDL for customers who want to build an integration with their salesforce.com organization only.
Generate Enterprise WSDL

Partner WSDL
A loosely typed WSDL for customers, partners, and ISVs who are building client applications for multiple organizations. It can be used to access data within any organization.
Generate Partner WSDL

Apex WSDL
Click on the link below to download an Apex programming WSDL.
Generate Apex WSDL

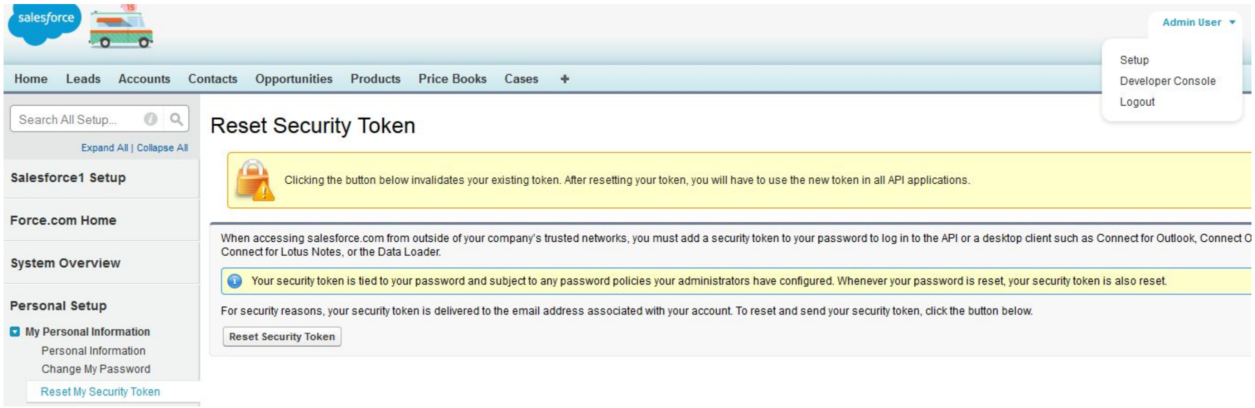
Metadata WSDL
Click on the link below to download a Metadata WSDL file.
Generate Metadata WSDL

Tooling WSDL
Click on the link below to download a Tooling WSDL file.
Generate Tooling WSDL

Click on the link below to download a Tooling WSDL file with strongly typed enums.
Generate Tooling WSDL With Strongly Typed Enums

Fig: Generating Enterprise WSDL file

- Generate Token Key.
  Go to My Settings **>>** Personal **>>** Reset My Security Token **>>** click on the Reset Security
- Token button is shown below:



Reset Security Token

Clicking the button below invalidates your existing token. After resetting your token, you will have to use the new token in all API applications.

When accessing salesforce.com from outside of your company's trusted networks, you must add a security token to your password to log in to the API or a desktop client such as Connect for Outlook, Connect O
Connect for Lotus Notes, or the Data Loader.

Your security token is tied to your password and subject to any password policies your administrators have configured. Whenever your password is reset, your security token is also reset.

For security reasons, your security token is delivered to the email address associated with your account. To reset and send your security token, click the button below.

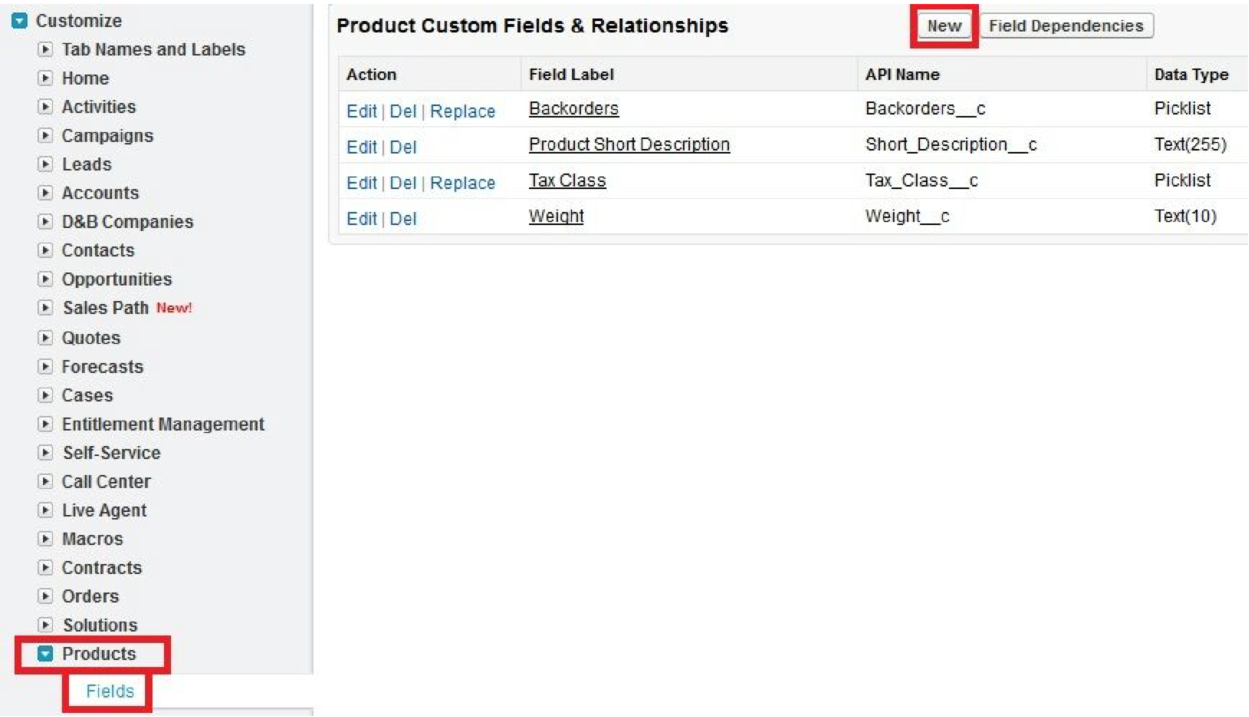
Fig: Reset Security Token

5. Create Custom fields in Salesforce

Note: To create custom fields in Salesforce follow the below Process

Custom fields for Products object

- Goto Setup **>>** App Setup **>>** Customize **>>** Products **>>** Fields **>>** and click **'New'** button to create custom fields.



Product Custom Fields & Relationships

Action	Field Label	API Name	Data Type
Edit | Del | Replace	Backorders	Backorders__c	Picklist
Edit | Del	Product Short Description	Short_Description__c	Text(255)
Edit | Del | Replace	Tax Class	Tax_Class__c	Picklist
Edit | Del	Weight	Weight__c	Text(10)

Fig: Creation of new custom field in Products.

Create a custom field Weight in Salesforce under object Products

1. Choose the field type i.e. Data Type as Text and click on Next Button.
2. Enter the mandatory field details which were highlighted with red color bar and click on Next button.
3. Again Click the Next button
4. Click on Save button

Create a custom field Backorders in Salesforce under object Products

1. Choose the field type i.e. Data Type as Picklist and click on Next Button.
2. Enter the mandatory field details which were highlighted with red color bar and click on Next button.
3. Enter values for the picklist, with each value separated by a new line. No back orders, Allow Quantity above 0, Allow Quantity below 0
4. Again Click the Next button
5. Click on Save button

Create a custom field Short Description in Salesforce under object Products

1. Choose the field type i.e. Data Type as Text and click on Next Button.
2. Enter the mandatory field details which were highlighted with red color bar and click on Next button.
3. Again Click the Next button
4. Click on Save button

Create a custom field Tax Class in Salesforce under object Products

1. Choose the field type i.e. Data Type as Picklist and click on Next Button.
2. Enter the mandatory field details which were highlighted with red color bar and click on Next button.
3. Enter values for the picklist, with each value separated by a new line. None, Taxable Goods, Shipping, Tax Exempt
4. Again Click the Next button
5. Click on Save button

Custom fields for Pricebook2 object

- Goto Setup **>>** App Setup **>>** Customize **>>** Pricebook2 **>>** Fields **>>** and click **'New'** button to create custom fields.

Create a custom field Quantity in Salesforce under object Pricebook2

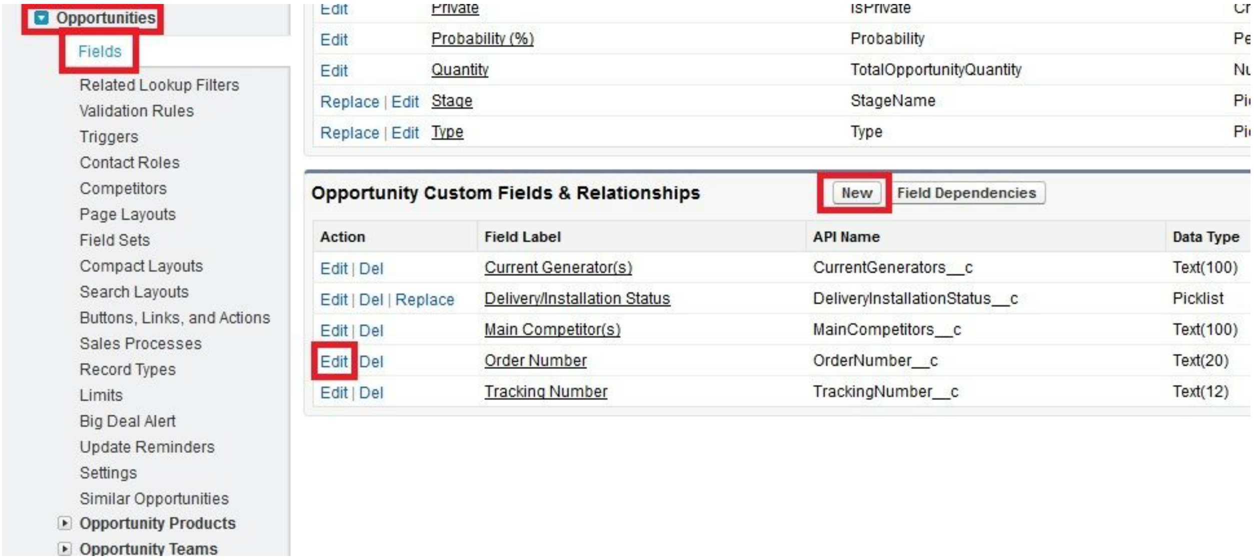
1. Choose the field type i.e. Data Type as Number and click on Next Button.
2. Enter the mandatory field details which were highlighted with red color bar and click on Next button.
3. Again Click the Next button
4. Click on Save button

Create a custom field Stock Availability in Salesforce under object Pricebook2

1. Choose the field type i.e. Data Type as Picklist and click on Next Button.
2. Enter the mandatory field details which were highlighted with red color bar and click on Next button.
3. Enter values for the picklist, with each value separated by a new line. In Stock, Out of Stock
4. Again Click the Next button
5. Click on Save button

Custom fields for Opportunities object

- Goto Setup **>>** App Setup **>>** Customize **>>** Opportunities **>>** Fields **>>** click **'New'** button to create custom fields.



Edit	Private	IsPrivate	Cr
Edit	Probability (%)	Probability	Pe
Edit	Quantity	TotalOpportunityQuantity	Nu
Replace | Edit	Stage	StageName	Pi
Replace | Edit	Type	Type	Pi

Opportunity Custom Fields & Relationships

Action	Field Label	API Name	Data Type
Edit | Del	Current Generator(s)	CurrentGenerators__c	Text(100)
Edit | Del | Replace	Delivery/Installation Status	DeliveryInstallationStatus__c	Picklist
Edit | Del	Main Competitor(s)	MainCompetitors__c	Text(100)
Edit | Del	Order Number	OrderNumber__c	Text(20)
Edit | Del	Tracking Number	TrackingNumber__c	Text(12)

Fig: Creation of new custom field in Opportunities

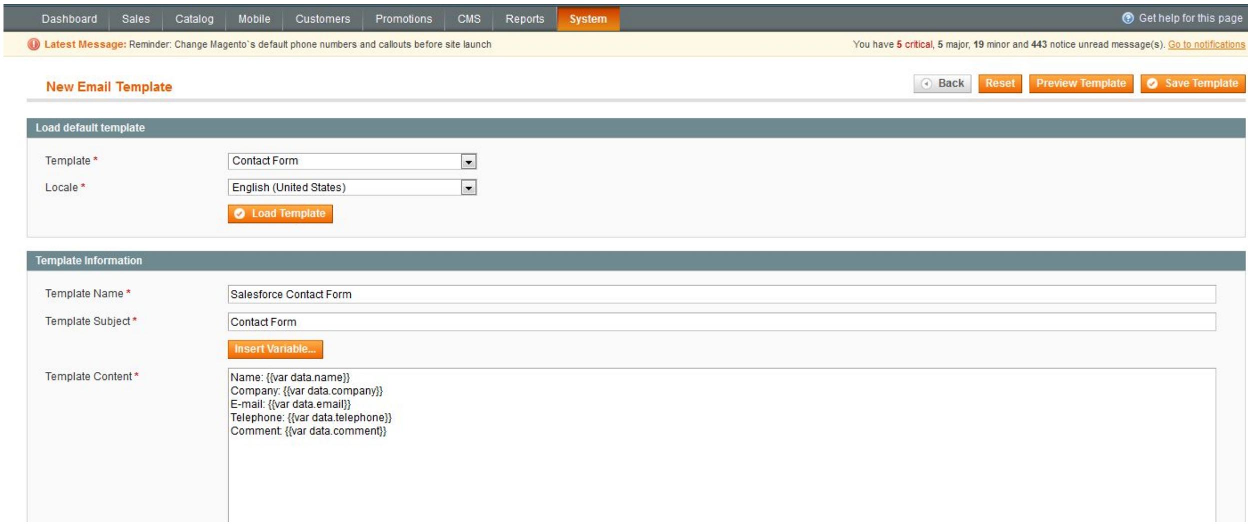
Create a custom field "Order number" in Salesforce under object Opportunities

1. If the “Ordernumber_c” custom field already exists, then increase the length to 12 by editing.
2. If not present, choose the field type i.e. Data Type as Text and click on Next Button.
3. Enter the mandatory field details which were highlighted with red color bar and click on Next button.
4. Again Click the Next button
5. Click on Save button

6. Contact Us form configuration setting in Magento

Create Custom Field for Contact Us form in Magento

1. Login to Magento Admin panel.
2. Go to System >>Transactional Emails >> Add New Template
3. Select ‘Contact Form’ from Template drop down
4. Click on ‘Load Template’.



Latest Message: Reminder: Change Magento`s default phone numbers and callouts before site launch

You have 5 critical, 5 major, 19 minor and 443 notice unread message(s). Go to notifications

New Email Template

Load default template

Template * Contact Form

Locale * English (United States)

Template Information

Template Name * Salesforce Contact Form

Template Subject * Contact Form

Template Content *

```
Name: {{var data.name}}
Company: {{var data.company}}
E-mail: {{var data.email}}
Telephone: {{var data.telephone}}
Comment: {{var data.comment}}
```

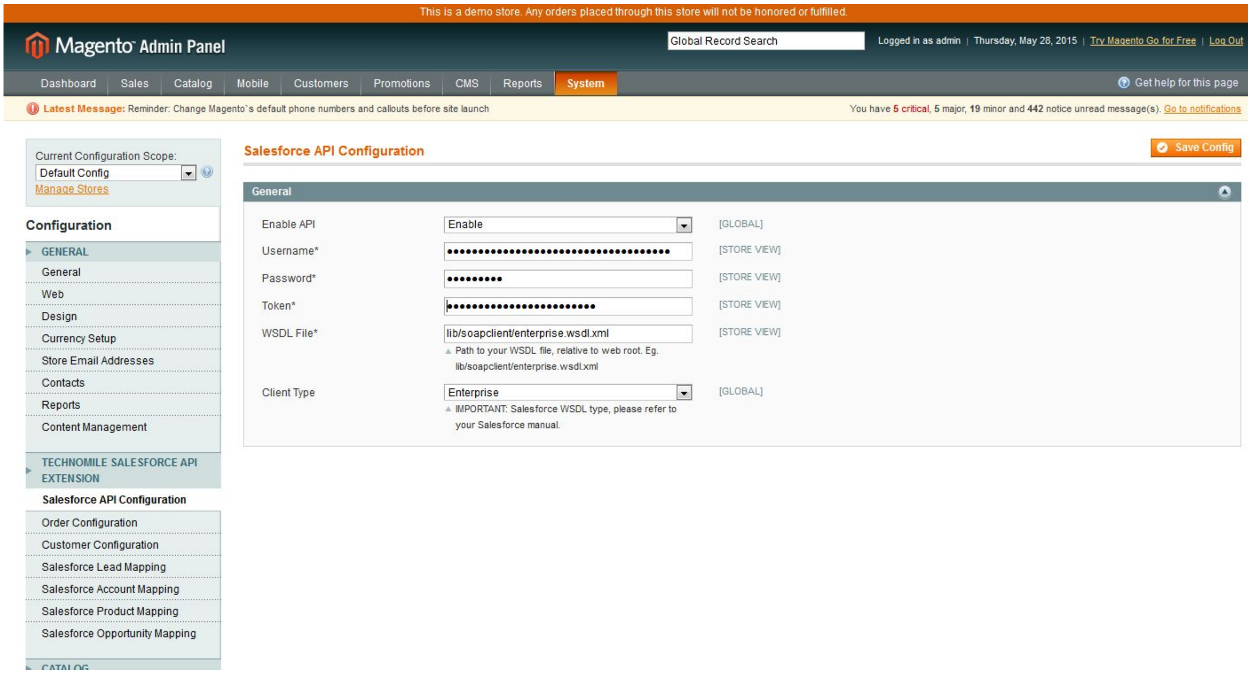
5. Add “Company: {{var data.company}}” in the template content, to add a new Company field in the form.
6. Click on ‘Save Template’.

7. Salesforce Configuration in Magento

Configure Salesforce API to allow data to sync from Magento to Salesforce

Configure Salesforce API Extension

1. Login to Magento Admin panel.
2. Go to System >> Configuration
3. In ‘TechnoMile Salesforce API Extension’ section, click on ‘Salesforce API Configuration’.
4. Enable the API
5. Fill SalesForce username, password, token
6. Specify the path of WSDL file.
7. Select the client type (Enterprise, Partner)


This is a demo store. Any orders placed through this store will not be honored or fulfilled.
Magento Admin Panel
Global Record Search
Logged in as admin | Thursday, May 28, 2015 | Try Magento Go for Free | Log Out
Dashboard
Sales
Catalog
Mobile
Customers
Promotions
CMS
Reports
System
Get help for this page
Latest Message: Reminder: Change Magento`s default phone numbers and callouts before site launch
You have 5 critical, 5 major, 19 minor and 442 notice unread message(s). Go to notifications
Current Configuration Scope:
Default Config
Manage Stores
Configuration
GENERAL
General
Web
Design
Currency Setup
Store Email Addresses
Contacts
Reports
Content Management
TECHNOMILE SALESFORCE API EXTENSION
Salesforce API Configuration
Order Configuration
Customer Configuration
Salesforce Lead Mapping
Salesforce Account Mapping
Salesforce Product Mapping
Salesforce Opportunity Mapping
CATALOG
Save Config

Salesforce API Configuration

General

Enable API: Enable [GLOBAL]

Username*: ••••••••••••••••••••••••••••••••••• [STORE VIEW]

Password*: ••••••••• [STORE VIEW]

Token*: ••••••••••••••••••••••••• [STORE VIEW]

WSDL File*: lib/soapclient/enterprise.wsdl.xml [STORE VIEW]
Path to your WSDL file, relative to web root. Eg. lib/soapclient/enterprise.wsdl.xml

Client Type: Enterprise [GLOBAL]
IMPORTANT: Salesforce WSDL type, please refer to your Salesforce manual.

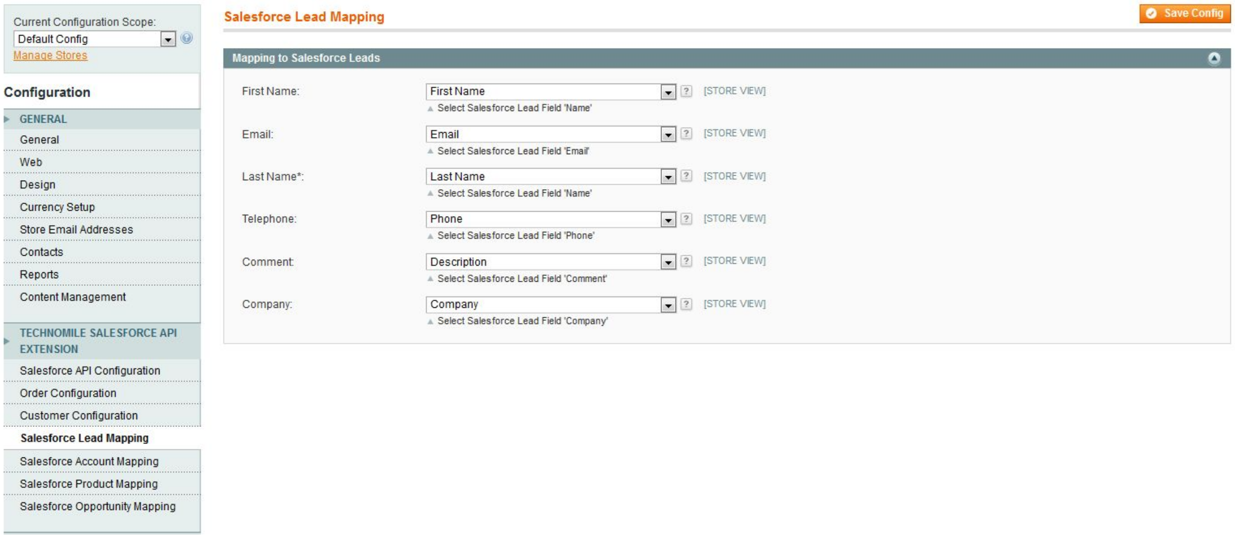
8. Save the configurations.
9. This will connect the magento app to SalesForce through SOAP.

8. Mapping Magento and Salesforce Attributes

Map Magento Contact Fields to respective Salesforce Lead Fields

1. In ‘TechnoMile Salesforce API Extension’ section, click on ‘Salesforce Lead Mapping’.

2. Map the respective fields.



Salesforce Lead Mapping

Mapping to Salesforce Leads

First Name: First Name [STORE VIEW]
Select Salesforce Lead Field 'Name'

Email: Email [STORE VIEW]
Select Salesforce Lead Field 'Email'

Last Name*: Last Name [STORE VIEW]
Select Salesforce Lead Field 'Name'

Telephone: Phone [STORE VIEW]
Select Salesforce Lead Field 'Phone'

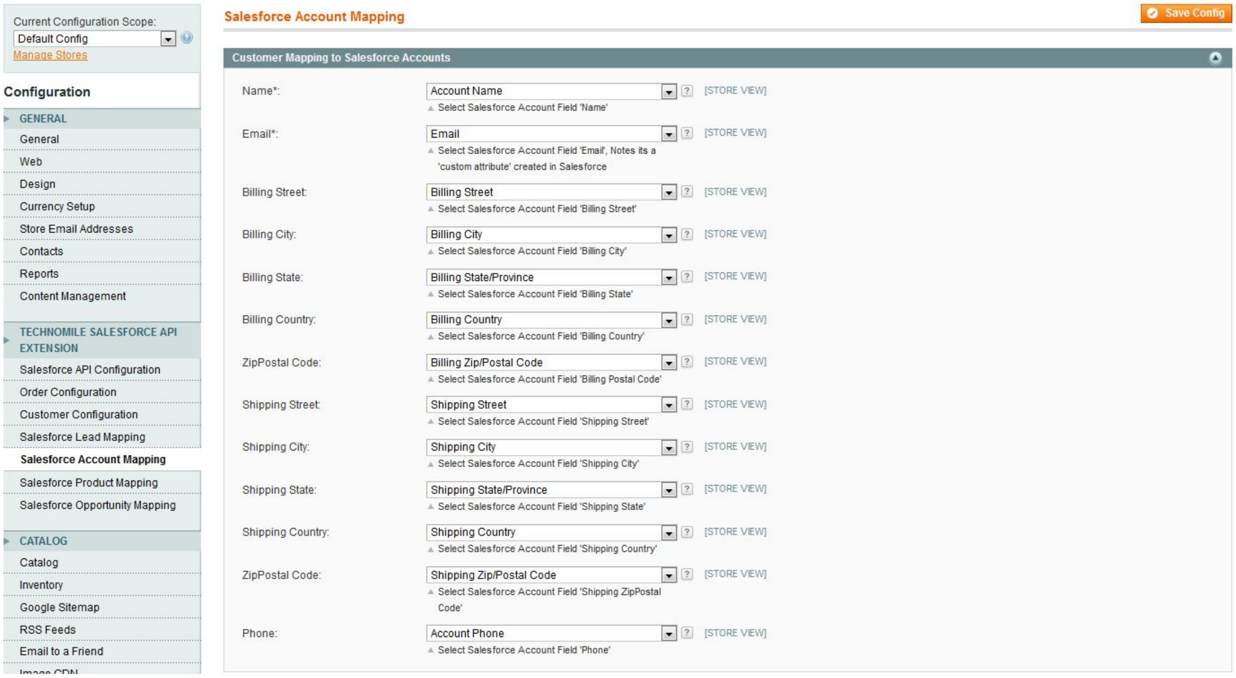
Comment: Description [STORE VIEW]
Select Salesforce Lead Field 'Comment'

Company: Company [STORE VIEW]
Select Salesforce Lead Field 'Company'

3. Click ‘Save Config’ button.

Map Magento Customer Fields to respective Salesforce Account Fields

1. In ‘TechnoMile Salesforce API Extension’ section, click on ‘Salesforce Account Mapping’.
2. Map the respective fields.



Salesforce Account Mapping

Customer Mapping to Salesforce Accounts

Name*: Account Name [STORE VIEW]
Select Salesforce Account Field 'Name'

Email*: Email [STORE VIEW]
Select Salesforce Account Field 'Email', Notes its a 'custom attribute' created in Salesforce

Billing Street: Billing Street [STORE VIEW]
Select Salesforce Account Field 'Billing Street'

Billing City: Billing City [STORE VIEW]
Select Salesforce Account Field 'Billing City'

Billing State: Billing State/Province [STORE VIEW]
Select Salesforce Account Field 'Billing State'

Billing Country: Billing Country [STORE VIEW]
Select Salesforce Account Field 'Billing Country'

ZipPostal Code: Billing Zip/Postal Code [STORE VIEW]
Select Salesforce Account Field 'Billing Postal Code'

Shipping Street: Shipping Street [STORE VIEW]
Select Salesforce Account Field 'Shipping Street'

Shipping City: Shipping City [STORE VIEW]
Select Salesforce Account Field 'Shipping City'

Shipping State: Shipping State/Province [STORE VIEW]
Select Salesforce Account Field 'Shipping State'

Shipping Country: Shipping Country [STORE VIEW]
Select Salesforce Account Field 'Shipping Country'

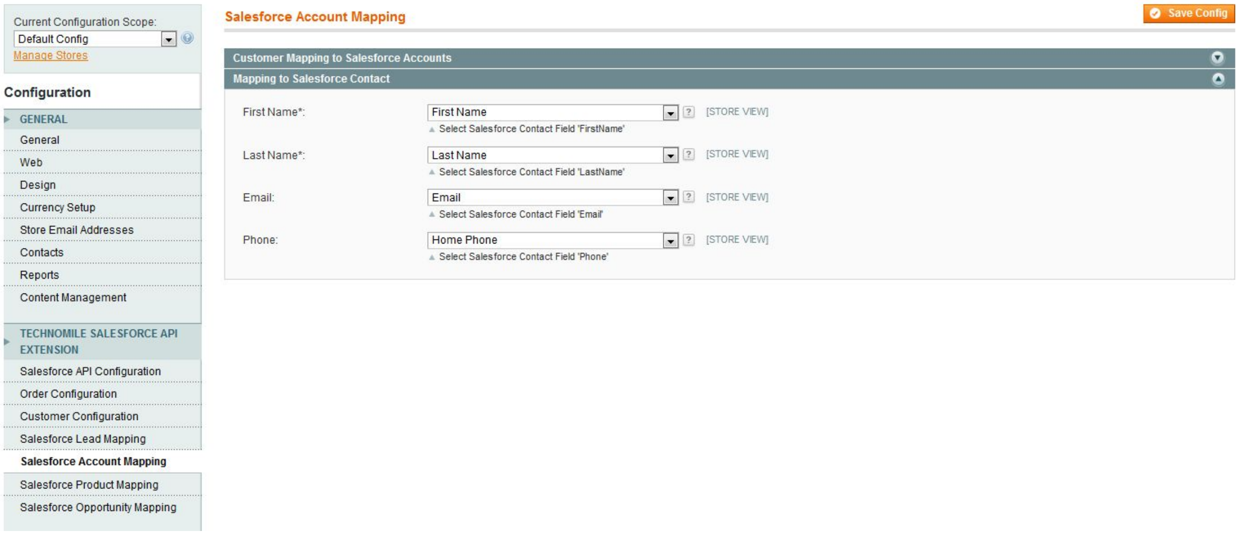
ZipPostal Code: Shipping Zip/Postal Code [STORE VIEW]
Select Salesforce Account Field 'Shipping ZipPostal Code'

Phone: Account Phone [STORE VIEW]
Select Salesforce Account Field 'Phone'

3. Click ‘Save Config’ button.

Map Magento Customer Fields to respective Salesforce Contact Fields

1. In ‘TechnoMile Salesforce API Extension’ section, click on ‘Salesforce Account Mapping’.
2. Map the respective fields.



Salesforce Account Mapping

Customer Mapping to Salesforce Accounts

Mapping to Salesforce Contact

First Name*: First Name [STORE VIEW]
Select Salesforce Contact Field 'FirstName'

Last Name*: Last Name [STORE VIEW]
Select Salesforce Contact Field 'LastName'

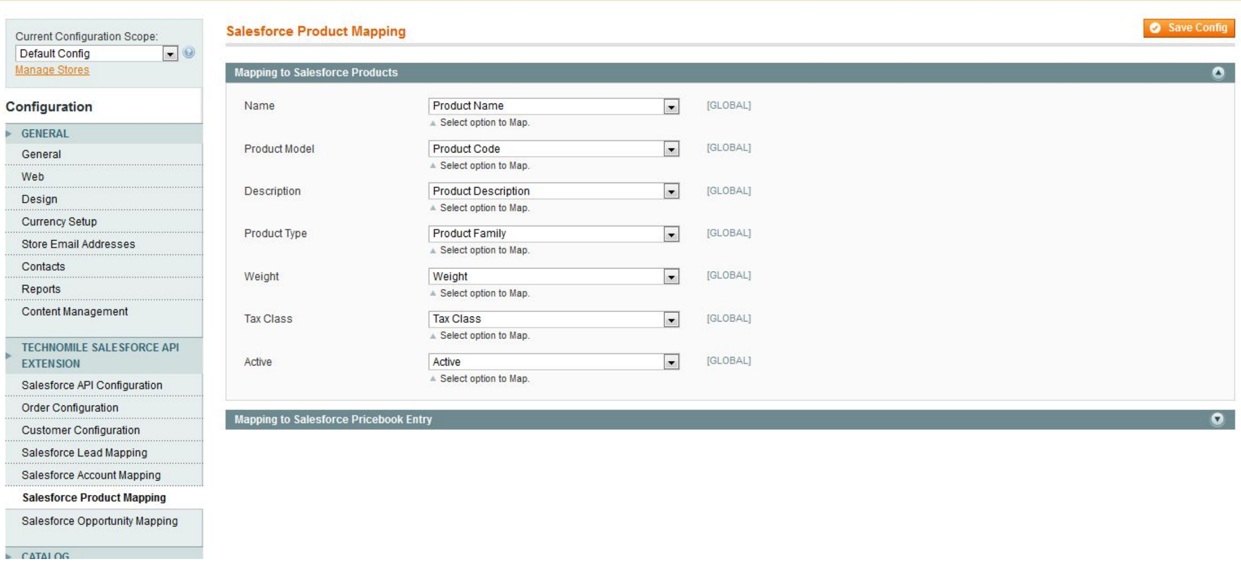
Email: Email [STORE VIEW]
Select Salesforce Contact Field 'Email'

Phone: Home Phone [STORE VIEW]
Select Salesforce Contact Field 'Phone'

3. Click ‘Save Config’ button.

Map Magento Product Fields to respective Salesforce Product Fields

1. In ‘TechnoMile Salesforce API Extension’ section, click on ‘Salesforce Product Mapping’.
2. Map the respective fields.



Salesforce Product Mapping

Mapping to Salesforce Products

Name: Product Name [GLOBAL]
Select option to Map.

Product Model: Product Code [GLOBAL]
Select option to Map.

Description: Product Description [GLOBAL]
Select option to Map.

Product Type: Product Family [GLOBAL]
Select option to Map.

Weight: Weight [GLOBAL]
Select option to Map.

Tax Class: Tax Class [GLOBAL]
Select option to Map.

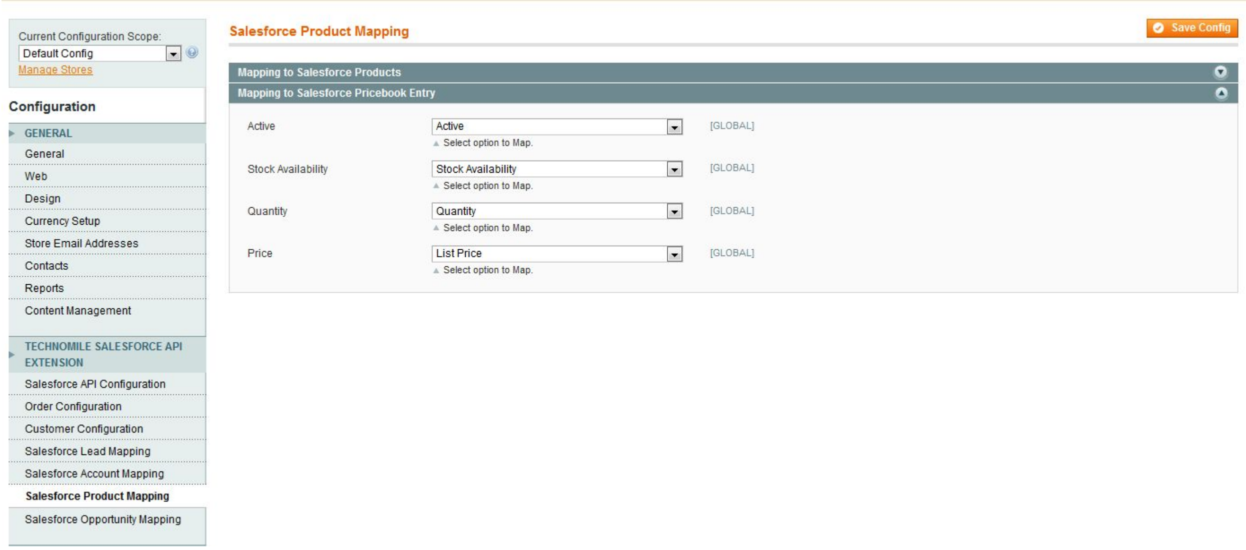
Active: Active [GLOBAL]
Select option to Map.

Mapping to Salesforce Pricebook Entry

3. Click ‘Save Config’ button.

Map Magento Product Fields to respective Salesforce PriceBook Fields

1. In ‘TechnoMile Salesforce API Extension’ section, click on ‘Salesforce Product Mapping’.
2. Map the respective fields.



Salesforce Product Mapping

Mapping to Salesforce Products

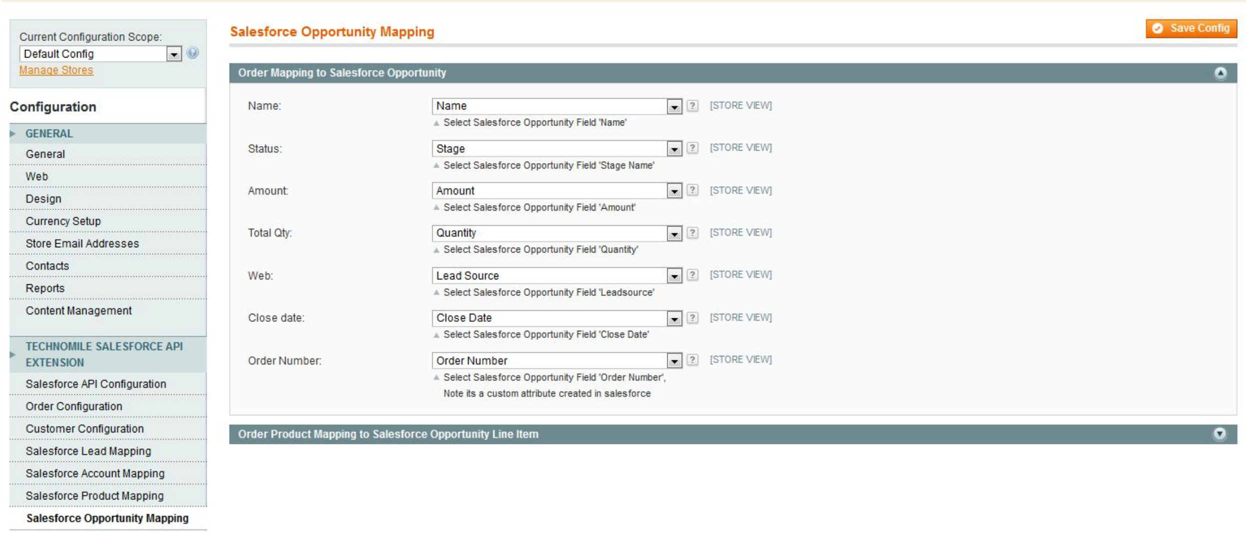
Mapping to Salesforce Pricebook Entry

Active	Active Select option to Map.	[GLOBAL]
Stock Availability	Stock Availability Select option to Map.	[GLOBAL]
Quantity	Quantity Select option to Map.	[GLOBAL]
Price	List Price Select option to Map.	[GLOBAL]

3. Click ‘Save Config’ button.

Map Magento Order Fields to respective Salesforce Opportunity Fields

1. In ‘TechnoMile Salesforce API Extension’ section, click on ‘Salesforce Opportunity Mapping’.
2. Map the respective fields.



Salesforce Opportunity Mapping

Order Mapping to Salesforce Opportunity

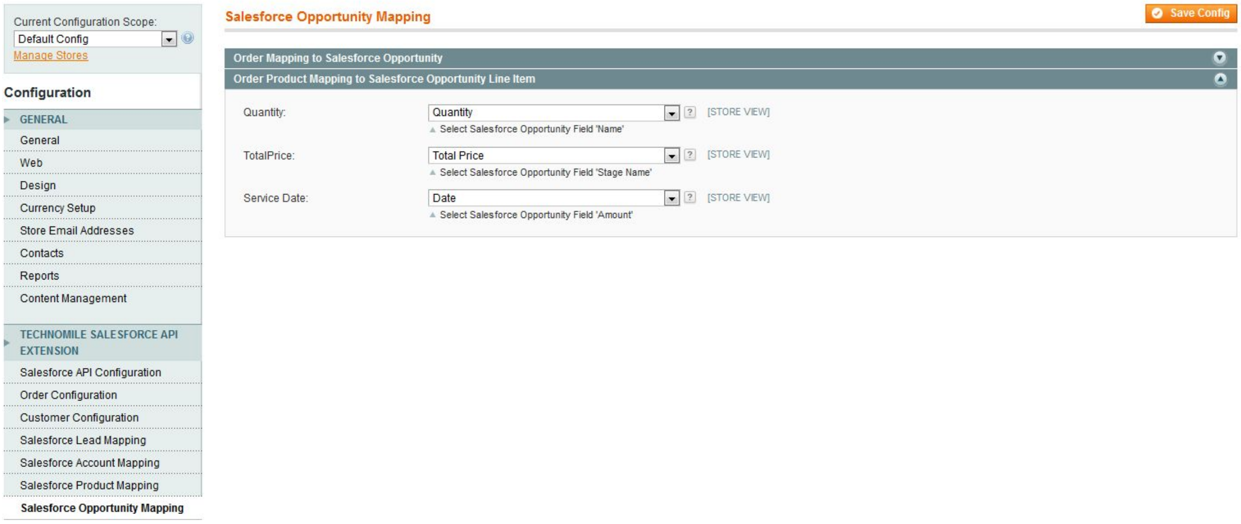
Name:	Name Select Salesforce Opportunity Field 'Name'	[STORE VIEW]
Status:	Stage Select Salesforce Opportunity Field 'Stage Name'	[STORE VIEW]
Amount:	Amount Select Salesforce Opportunity Field 'Amount'	[STORE VIEW]
Total Qty:	Quantity Select Salesforce Opportunity Field 'Quantity'	[STORE VIEW]
Web:	Lead Source Select Salesforce Opportunity Field 'Leadsource'	[STORE VIEW]
Close date:	Close Date Select Salesforce Opportunity Field 'Close Date'	[STORE VIEW]
Order Number:	Order Number Select Salesforce Opportunity Field 'Order Number', Note its a custom attribute created in salesforce	[STORE VIEW]

Order Product Mapping to Salesforce Opportunity Line Item

3. Click ‘Save Config’ button.

Map Magento Order Fields to respective Salesforce PriceBook2 Fields

1. In ‘TechnoMile Salesforce API Extension’ section, click on ‘Salesforce Opportunity Mapping’.
2. Map the respective fields.



Salesforce Opportunity Mapping

Order Mapping to Salesforce Opportunity

Order Product Mapping to Salesforce Opportunity Line Item

Quantity: Quantity [STORE VIEW]
Select Salesforce Opportunity Field 'Name'

TotalPrice: Total Price [STORE VIEW]
Select Salesforce Opportunity Field 'Stage Name'

Service Date: Date [STORE VIEW]
Select Salesforce Opportunity Field 'Amount'

3. Click ‘Save Config’ button.

9. Decode Source Guardian Code

- Check PHP Configuration in the server to see whether it has any Source Guardian modules or not?
- In order to check, you need to create a test file in the document root with the following script with the name **phpinfo.php**
  E.g.: http://www.domainname.com/phpinfo.php
  ```
  <? php
  phpinfo ();
  ?>
  ```
- Search for **Source Guardian** from out. If the module exists, then skip next Step else proceed with next Step
- The Source Code of TechnoMile Salesforce extension is encrypted. In regarding to that, you need a Source Guardian Loaders to load the TechnoMile Salesforce connect successfully.
- Open the following link to download the specific Source Guardian Loader & follow the instructions
  http://www.sourceguardian.com/ixeds/download.php

- In order to detect the required loader and know how to install it, please enter the phpinfo link to the phpinfo () page that is running on your server. E.g.: http://www.domainname.com/phpinfo.php
- Download the required loader ixed.xx.lin on the Server
- Move the downloaded file ixed.xx.lin to the location mentioned in the steps given by Source Guardian
- Open your php.ini file from the location
- Restart the web server

10. Heroku Configuration

- Sign up https://signup.heroku.com/
- Install the Heroku Toolbelt https://toolbelt.heroku.com/
- Login https://id.heroku.com/login
- Go to link below and click on Deploy to heroku for installing Magento http://www.technomile.com/capabilities/application-development/heroku/magento-salesforce-integration

Heroku Setup on Local Machine

Run the following commands using git bash

1. heroku login
2. heroku keys:add
3. git config --global user.email "*YOUR_EMAIL_ADDRESS*"
4. git config --global user.name "*YOUR_NAME*"
5. git clone git@heroku.com:*NAME_OF_YOUR_HEROKU_APP*.git (ex: git clone git@heroku.com:heroku-magento-salesforce.git)
6. cd *NAME_OF_YOUR_HEROKU_APP*
7. Type this command for database details *heroku config*
8. CLEARDB_DATABASE_URL: mysql://USERNAME:PASSWORD@HOSTNAME/DATABASE_NAME?reconnect=true

Magento Database Configuration

Find a xml file at this location *app/etc/local.xml.sample* in your *HEROKU_APP* and rename this file to local.xml.Next need to write database credentials and current date in this local.xml file as below.

1. Current date format like:<date><![CDATA[*Tue, 16 Jun 2015 06:29:23 +0000*]]></date>

2. <host><![CDATA[*YOUR_DATABASE_HOST_NAME*]]></host>
3. <username><![CDATA[*YOUR_DATABASE_USERNAME*]]></username>
4. <password><![CDATA[*YOUR_DATABASE_PASSWORD*]]></password>
5. <dbname><![CDATA[*YOUR_DATABASE_DATABASE_NAME*]]></dbname>

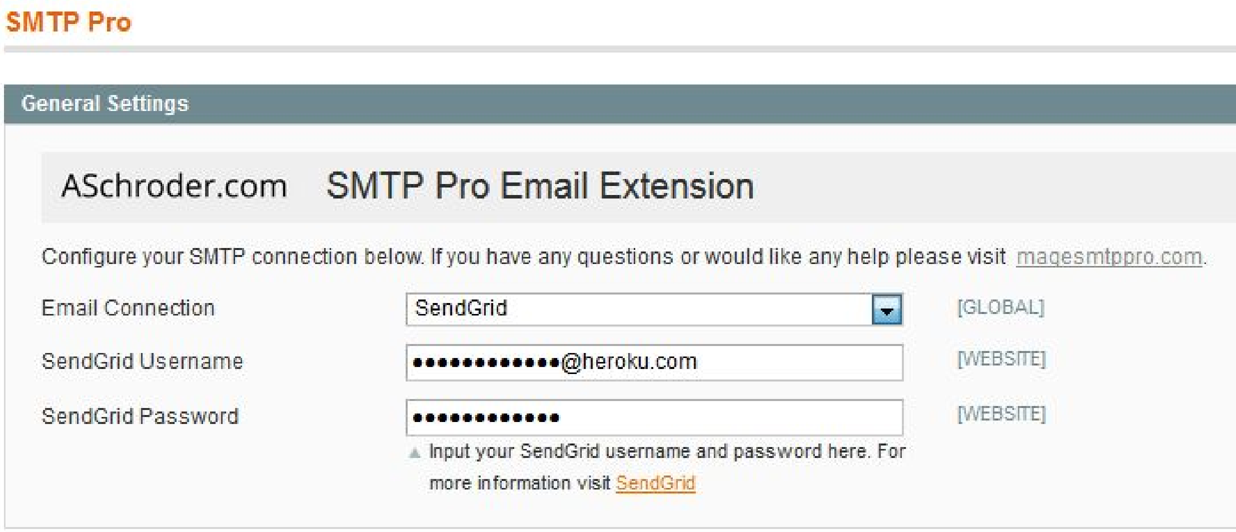
Deploy code on Server.

Run following commands using git bash.

1. git init
2. git add .
3. git commit -m *"YOUR_COMMIT_MESSAGE"*
4. git push heroku master

Update credentials for the SendGrid and in Magento.

1. Login to Magento Admin panel.
2. Goto System **>>** Configuration



SMTP Pro

General Settings

ASchroder.com SMTP Pro Email Extension

Configure your SMTP connection below. If you have any questions or would like any help please visit magesmtppro.com.

Email Connection: SendGrid [GLOBAL]

SendGrid Username: ●●●●●●●●●●●●@heroku.com [WEBSITE]

SendGrid Password: ●●●●●●●●●●●● [WEBSITE]

Input your SendGrid username and password here. For more information visit SendGrid

3. Select Email Connection as 'SendGrid' and add SendGrid Username and Password.
4. Click on ‘Save Template’.

11. Uninstall Magento Salesforce Connect

- Login into Magento Admin

- Go to System **>>** Configuration **>>** TechnoMile Salesforce **>>** Extension Options.
- Select No from the drop down and Click on Save Config. Or
- Go to FTP Client.
- Go to Magento Folders.
- Follow the path app\etc\modules and select file TechnoMile_Salesforce.xml & Right Click the file to EDIT.
- Press Ctrl+F and change TRUE with FALSE
- Save the TechnoMile_Salesforce.xml.