Hostbill Integration Manual

Automatic and fast provisioning of BackupAgent for service providers using HostBill

1. About BackupAgent For HostBill

BackupAgent for HostBill is a module which allows your customers to quickly and conveniently order and manage a ready BackupAgent services from your HostBill client area.

2. Installation & Configuration

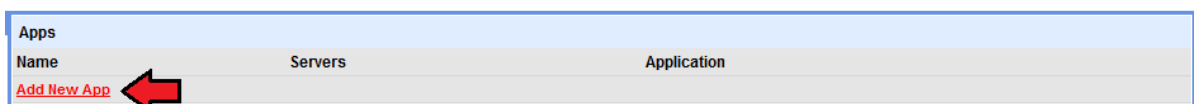The only additional requirement for running this module is to install the mcrypt extension, see:http://php.net/manual/en/book.mcrypt.php

2.1 Upload the Module Files

Upload and unzip the module into your main HostBill directory:

2.2 Add New App

Log in to your HostBill and go to 'Settings' → 'Apps'. Next, click on 'Add New App' as shown on the following screen.

2.3 Configure The App

Choose 'backupagent' from an Application dropdown menu, enter your server details, test configuration and when the result will be successful press on 'Add New App' button.

2.4 Verify The App

As a result of the previous action, you will see a new app which is ready to use. Let's proceed with configuration of the module.

2.5 Add New Order Page

Now let's add new order page and new product. To do so, go to 'Settings' → 'Products & Services' and click on 'Add new order page'.

2.6 Configure The Order Page

Configure your new order page and once you are ready, click on 'Add new order page' button.

2.7 Add New Product

When you have new order page, you can add a new product. For this purpose click on 'Add new product' button as presented on the screen below.

2.8 Configure The Product

Perform basic configuration of the product and go to 'Connect with App' category. Choose 'backupagent' from a 'Third party app' dropdown menu. Afterwards enter same connection details that you entered while configuring the app and test the connection to make sure that details are correct. Now you can configure the App for your product. When your product is ready for action, press on 'Save Changes' button.

2.9 Import The Components

Go to 'Components' category and import the components prepared for this module by simply pressing on 'Import' button and browsing the 'components.yml' file.

2.10 Verify The Import Process

This is what you are going to see after a successful import.

2.11 Enable Client Functions

The last configuration step is enabling some of the client functions. Go to 'Client Functions' category and enable 'Reset Password' and 'Login Details'. Don't forget to save the changes.

3. Order Placing

3.1 Placing An Order

Now your product is ready to use by your customers. Let's see how the ordering process looks. As you can see on the following screen, your customers can choose a suitable quota and maximum bandwidth, their user profile and account type.

3.2 Client Area

This is what your customers are going to see in your HostBill client area. Customers can easily access their BackupAgent control panel owing to 'Log In To Client Area' button, monitor current resource usage and easily upgrade/downgrade the product.

From now on, you know everything about module installation, configuration and usage. You can read our tips and you are ready to go!

3.3 Tips

3.3.1 Client area info/login button

The info table template is loaded from "includes/modules/Hosting/backupagent/templates/{template_name}/clientarea.tpl", where {template_name} is your current template (default/nextgen/etc...). If custom template directory won't be found, then the nextgen's one will be loaded (includes/modules/Hosting/backupagent/templates/nextgen/clientarea.tpl).

3.3.2 Provided details

Make sure that details you enter while configuring the app and product in your HostBill system are correct. Mistake in one field might cause module to not work properly e.g. wrong parent reseller value will unable product from being created.

3.3.3 Debugging issues

In some cases the module integration might not work at once. Step-by-step debugging of the module might be required to locate the actual cause in the PHP code.

HostBill has error_reporting turned off by default, so on critical error it will render blank page. If you will stumble upon any issues you can enable error reporting by editing your file:

/includes/config.php

by adding/editing line:

$enable_debug=true;

you can also do this by inserting in the code (somewhere at the start of the script):

error_reporting(E_ALL); ini_set('display_errors',1);

Usually this is enough, but in some cases it is required to add debug lines of code to produce an error, for example by using *die();* until something useful appeared on the screen.