

Бонус при регистрации

Если при регистрации пользователя необходимо начислять ему определенную сумму на лицевой счет, то можно воспользоваться следующим решением.

1. Создаем константы REGISTRATION_BONUS_VALUE и REGISTRATION_BONUS_CAPTION, например:

```
REGISTRATION_BONUS_VALUE = 20  
REGISTRATION_BONUS_CAPTION = Приветственный бонус за регистрацию
```

2. Добавляем наблюдатель:

```
<?php  
  
defined('HOSTCMS') || exit('HostCMS: access denied.');
```



```
/**  
 * Наблюдатель для отслеживания изменения цены товара и рассылки уведомлений по  
 * подпискам.  
 *  
 * @author Maxim Zasorin, KAD Systems (©) 2017  
 * @date 18-12-2017  
 */  
class Siteuser_Observer_RegistrationBonus  
{  
    /**  
     * Сайты и интернет-магазины.  
     *  
     * @var array  
     */  
    static protected $_sites = array(  
        7 => array(9)  
    );  
  
    /**  
     * Регистрирует наблюдатель.  
     *  
     * @return void  
     */  
    static public function attach()  
    {  
        Core_Event::attach('siteuser.onAfterCreate', array(__CLASS__,  
            'addRegistrationBonus'));  
    }  
  
    /**  
     * Деактивирует наблюдатель.  
     *  
     * @return void  
     */
```

```

    */
    static public function detach()
    {
        Core_Event::detach('siteuser.onAfterCreate', array(__CLASS__,
'addRegistrationBonus'));
    }

    /**
    * Добавляет бонус при регистрации.
    *
    * @param Siteuser_Model $oSiteuser
    * @return void
    */
    static public function addRegistrationBonus(Siteuser_Model
$oSiteuser)
    {
        // Если текущего сайта нет в списке или бонус не активирован, то прерываем
        выполнение
        if (!in_array(CURRENT_SITE, array_keys(self::$_sites)) ||
!defined('REGISTRATION_BONUS_VALUE'))
        {
            return;
        }

        foreach (self::$_sites[CURRENT_SITE] as $shopId)
        {
            $oShopSiteuserTransaction =
Core_Entity::factory('Shop_Siteuser_Transaction');
            $oShopSiteuserTransaction->shop_id =
Core_Entity::factory('Shop', $shopId)->id;
            $oShopSiteuserTransaction->siteuser_id = $oSiteuser->id;

            try
            {
                $oShopSiteuserTransaction->active = 1;
                $oShopSiteuserTransaction->amount_base_currency =
                $oShopSiteuserTransaction->amount =
floatval(REGISTRATION_BONUS_VALUE);
                $oShopSiteuserTransaction->type = 1;
                $oShopSiteuserTransaction->description =
defined('REGISTRATION_BONUS_CAPTION')
                ? REGISTRATION_BONUS_CAPTION
                : 'Приветственный бонус за регистрацию';
                $oShopSiteuserTransaction->save();
            }
            catch (Exception $e)
            {
                // Ничего не делаем, если не удалось создать бонус
            }
        }
    }
}

```

```
}
```

3. В наблюдателе указываем для каких сайтов и интернет-магазинов должен работать бонус:

```
/**  
 * Сайты и интернет-магазины.  
 *  
 * @var array  
 */  
static protected $_sites = array(  
    7 => array(9)  
);
```

4. Подключаем наблюдатель в bootstrap.php:

```
Siteuser_Observer_RegistrationBonus::attach();
```