Sample Code: Code Review

```
public function postAction() {
	//avoid notices warnings
	!isset($_POST['livechat_license_number']) ? $livechat_license_number = '' :
	 $livechat_license_number = $_POST['livechat_license_number'];
	!isset($_POST['livechat_groups']) ? $livechat_groups = '0' : $livechat_groups =
	 $_POST['livechat_groups'];
	!isset($_POST['livechat_params']) ? $livechat_params = '' : $livechat_params =
	 $_POST['livechat_params'];
	$config_table = Mage::getSingleton('core/resource')
	 ->getTableName('core_config_data');
	$read = Mage::getSingleton('core/resource')->getConnection('core_read');
	$query = 'SELECT * FROM ' . $config_table;
	$query .= ' WHERE scope="default" AND scope_id=0 AND
	path="livechat/general/license"';
	$results = $read->fetchAll($query);
	$write = Mage::getSingleton('core/resource')->getConnection('core_write');
	//check for existing configurations
	if ($row = array_pop($results)) {
		$license_id = $row['config_id'];
		$query = 'UPDATE ' . $config_table;
		$query .= ' SET value="' . $livechat_license_number . '"';
		$query .= ' WHERE config_id=' . $license_id;
		$write->query($query);
		$query = 'UPDATE ' . $config_table;
		$query .= ' SET value="' . $livechat_groups . '"';
		$query .= ' WHERE config_id=' . ++$license_id;
		$write->query($query);
		$query = 'UPDATE ' . $config_table;
		$query .= ' SET value="' . $livechat_params . '"';
		$query .= ' WHERE config_id=' . ++$license_id;

		$write->query($query);
	} else {
		$query = 'INSERT INTO ' . $config_table;
		$query .= ' (scope, scope_id, path, value)';
		$query .= ' VALUES ("default", 0, "livechat/general/license", "' .
		 $livechat_license_number . '"),';
		$query .= ' ("default", 0, "livechat/advanced/group", "0"),';
		$query .= ' ("default", 0, "livechat/advanced/params", "")';
		$write->query($query);
	}
	// Refresh the config.
	Mage::getConfig()->cleanCache();
	Mage::getConfig()->reinit();
	$this->_redirect('*/*/index');
}
```

According to the provided code sample I have identified it as bad coding. Because I can see some possibilities still there to improve code quality. For the bellow I will try to explain those in breaf.

Assumption: Assume all necessary code for the postAction method is given in sample code. No other relevant code are existing outside the postAction method.

According to the MEQP coding slandered, there are few violations can be found.

WARNING | [x] Opening brace should be on a new line

Bad	Good

public function postAction() {
  ….
}
	public function postAction()
	{
	  ….
	}

ERROR | [ ] Direct use of $_POST Superglobal detected.

Bad	Good
$_POST['field_name'] eg: $_POST['livechat_license_number']	$this->getRequest()->getParam('field_name'); eg: $this->getRequest()- >getParam('livechat_license_number');

WARNING | [ ] Variable "livechat_license_number" is not in valid camel caps format
WARNING | [ ] Variable "livechat_params" is not in valid camel caps format
WARNING | [ ] Variable "livechat_groups" is not in valid camel caps format
WARNING | [ ] Variable "config_table" is not in valid camel caps format
WARNING | [ ] Variable "license_id" is not in valid camel caps format

Bad	Good
Discourage to use $livechat_license_number type variable declarations.	Encurrage to use Camel Case declarations Eg: $livechatLicenseNumber

WARNING | [ ] Possible raw SQL statement 'SELECT * FROM ' detected.

Bad	Good
The idea of ORM is to never run raw queries inside modules. Eg: $query = 'SELECT * FROM ' . $config_table;	ORM helps to do that $query = 'SELECT * FROM ' . $config_table;

WARNING | [ ] fetchAll can be memory inefficient for large data sets.

Bad	Good
$results = $read->fetchAll($query); is not good for lager result. But according to given query it is returned single row. By considering the given implementation, it is not doing correct way. $config_table = Mage::getSingleton('core/resource')->getTableName('core_config_data');	Magento ORM helps to do that like below, We can use getModal or getSingleton to access the core_config_data resource. But getModel is not suit for this because it load core_config_data instance each time it calls. So better way to do is using getSingleton, it initiate instance very first time it calls and use initiated resource after each time it required. So code should be re-arrange like

$read = Mage::getSingleton('core/resource')->getConnection('core_read'); $query = 'SELECT * FROM ' . $config_table; $query .= ' WHERE scope="default" AND scope_id=0 AND path="livechat/general/license"'; $results = $read→fetchAll($query); This implementation is really bad and it violate magento best practices.	$configDataCollection = Mage::getSingleton('core/config_data') ->getCollection() ->addFieldToFilter('scope', 'default') ->addFieldToFilter('scope_id', '0') ->addFieldToFilter('path', 'livechat/general/license'); This implementation is more secure and more efficient while following magento standerds.

Not declared proper default values,

Bad	Good

$config_table = Mage::getSingleton('core/resource')->getTableName('core_config_data');
$read = Mage::getSingleton('core/resource')->getConnection('core_read');
$query = 'SELECT * FROM ' . $config_table;
$query .= ' WHERE scope="default" AND scope_id=0 AND path="livechat/general/license"';
$results = $read→fetchAll($query);
 	
Here the author does not implemented proper default values.
	cont APP_DEFAULT_SCOPE = ‘default’;
	cont APP_DEFAULT_SCOPE_ID = ‘0’;
	cont APP_DEFAULT_PATH = ‘livechat/general/license’
	…
	
	public function postAction()
	{
	…
	$configDataCollection = Mage::getSingleton('core/config_data')
	->getCollection()
	->addFieldToFilter('scope', 'default')
	->addFieldToFilter('scope_id', '0')
	->addFieldToFilter('path', 'livechat/general/license');
	…
	}

WARNING | [ ] Increment and decrement operators must be bracketed when used in string concatenation

Bad	Good
$query .= ' WHERE config_id=' . ++$license_id;	$query .= ' WHERE config_id=' . (++$license_id);

WARNING | [ ] Possible raw SQL statement 'INSERT INTO ' detected.

Bad	Good
$query = 'INSERT INTO ' . $config_table; $query .= ' (scope, scope_id, path, value)'; $query .= ' VALUES ("default", 0, "livechat/general/license", "' . $livechat_license_number . '"),'; $write→query($query); SQL raw queries used inside the code and it is not recommended to apply this type of	$data = array('scope'=>'default','scope_id'=>'0','path'=>'livechat/general/license','value'=>$livechat_license_number); $model = Mage::getSingleton('core/config_data')->setData($data); $insertId = $model→save()→getId(); Best way to do is using Singleton model.

implementation	

Possible raw SQL statement 'UPDATE ' detected

Bad	Good
$query = 'UPDATE ' . $config_table; $query .= ' SET value="' . $livechat_license_number . '"'; $query .= ' WHERE config_id=' . $license_id; $write→query($query); SQL raw update queries used inside the code and it is not recommended to apply this type of implementation	$data = array('scope'=>'default','scope_id'=>'0','path'=>'livechat/general/license','value'=>$livechat_license_number); $model = Mage::getSingleton('core/config_data')->load($license_id)->addData($data); $insertId = $model->setId($license_id)->save(); Best way to do is using Singleton model.

WARNING | [ ] Data access method QUERY detected outside of Resource Model

Bad	Good
$write→query($query); Wrong implementation is apply in the code.	Better implementation and clean code can be achieve through accessing model implementation.

No exception handling

Bad	Good

Can not see exception handling inside the code.	There should be exception handling like below,
	
	$model = Mage::getSingleton('core/config_data')->setData($data);
	try {
	  $insertId = $model->save()->getId();
	} catch (Exception $e){
	  echo $e->getMessage();
	}

No PHPDoc comment applied

Bad	Good
Can not see PHPDoc comment found.	To improve the quality of the code, there need to be PHP Doc comments within the code base.

By considering all the quality factors, the provided code is not written in magento coding slandered and it is not good quality code.