

Code Samples: vCenter Chargeback API

Last Updated: 11/4/09

The following code samples were presented during the VMware Coffee Talk on vCenter Chargeback APIs on 11/4/09. To download the Webex recording and slide presentation, please visit:

<http://communities.vmware.com/thread/234488>

For additional API-related resources, such as API Programming Guide and API References, please visit the vCenter Chargeback documentation page: http://www.vmware.com/support/pubs/vcbm_pubs.html

To share and collect feedback on additional code samples, please visit the VMware Developer Community's sample code repository: <http://communities.vmware.com/community/developer/codecentral>

vCenter Chargeback API: Generate Report Response

```
<?xml version="1.0" encoding="UTF-8"?>
<Response status="success" isValidLicense="true"
xmlns="http://www.vmware.com/vcenter/chargeback/1.0.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ReportTemplates>
    <ReportTemplate>
      <Name>Sample Report</Name>
      <Description>Description</Description>
      <ReportType>COST_REPORT</ReportType>
      <CreatedOn>Mon Oct 26 12:03:29 GMT+05:30 2009</CreatedOn>
      <CreatedByUser>1</CreatedByUser>
      <Reports>
        <Report>
          <DataDetails>
            <Duration>
              <From>Tue May 27 12:26:40 IST 2008</From>
              <To>Mon Oct 26 02:46:40 IST 2009</To>
            </Duration>
            <DataDetail>
              <CostModel id="2">
                <Name>My Cost Model</Name>
                <Description>Description</Description>
              </CostModel>
              <EntityCostDetails entityId="3559">
                <Name>Name</Name>
                <Type>VM</Type>
                <EntityCost>17156.25</EntityCost>
                <CategorizedCosts>
                  <CategoryCost id="1">
                    <Name>CPU</Name>
                    <Value>17156.25</Value>
                  </CategoryCost>
                  <CategoryCost id="99">
                    <Name>OtherCosts</Name>
                    <Value>0.00</Value>
                  </CategoryCost>
                </CategorizedCosts>
                <CategorizedUsage>
                  <CategoryUsage id="1">
                    <Name>CPU</Name>
                    <Value>1.71</Value>
                    <AdditionalInfo>
                      <Usage>1.71</Usage>
                      <ValidDuration>36028800</ValidDuration>
                    </AdditionalInfo>
                  </CategoryUsage>
                </CategorizedUsage>
                <InformationMessages>
                  <InformationMessage>
                    <TotalCost>0</TotalCost>
                    <StartTime>Mon Aug 04 20:14:00 GMT+05:30 2008</StartTime>
                    <EndTime>Sun Aug 31 00:00:00 GMT+05:30 2008</EndTime>
                    <Entity>MyVM</Entity>
                  </InformationMessage>
                </InformationMessages>
              </EntityCostDetails>
            </DataDetail>
          </DataDetails>
        </Report>
      </Reports>
    </ReportTemplate>
  </ReportTemplates>
</Response>
```

<Message>Usage data not available for the entity MyVM and the resource CPU from Mon Aug 04 20:14:00 GMT+05:30 2008 to Sun Aug 31 00:00:00 GMT+05:30 2008.</Message>
</InformationMessage>
<InformationMessage>
 <TotalCost>0</TotalCost>
 <StartTime>Tue Sep 01 00:00:00 GMT+05:30 2009</StartTime>
 <EndTime>Fri Sep 25 20:14:00 GMT+05:30 2009</EndTime>
 <Entity>MyVM</Entity>
resource CPU from Tue Sep 01 00:00:00 GMT+05:30 2009 to Fri Sep 25 20:14:00 GMT+05:30 2009.</Message>
</InformationMessage>
</InformationMessages>
<CostDetails>
 <UtilizationCost>
 <TotalCost>17156.25</TotalCost>
 <StartTime>Mon Aug 04 20:14:00 GMT+05:30 2008</StartTime>
 <EndTime>Fri Sep 25 20:14:00 GMT+05:30 2009</EndTime>
 <Entity>MyVM</Entity>
 <Parent>MyHost</Parent>
 <TotalDuration>36028800</TotalDuration>
 <AttributionPercentage>1.0</AttributionPercentage>
 <BaseRate>1.0000</BaseRate>
 <RateFactor>1</RateFactor>
 <UsageData>1.71</UsageData>
 <Resource>CPU : GHz Hour</Resource>
 <TotalMissingDataDuration>0</TotalMissingDataDuration>
 </UtilizationCost>
 <FixedCost>
 <TotalCost>0.0</TotalCost>
 <StartTime>0</StartTime>
 <EndTime>0</EndTime>
 <Entity>MyVM</Entity>
 <Parent>MyHost</Parent>
 <TotalDuration>36028800</TotalDuration>
 <AttributionPercentage>1.0</AttributionPercentage>
 <Source></Source>
 <CostPerDuration>0.0</CostPerDuration>
 <Duration>0.0</Duration>
 </FixedCost>
 <ReservationCost>
 <TotalCost>17156.25</TotalCost>
 <StartTime>Mon Aug 04 20:14:00 GMT+05:30 2008</StartTime>
 <EndTime>Fri Sep 25 20:14:00 GMT+05:30 2009</EndTime>
 <Entity>MyVM</Entity>
 <Parent>MyHost</Parent>
 <TotalDuration>36028800</TotalDuration>
 <AttributionPercentage>1.0</AttributionPercentage>
 <BaseRate>1.0000</BaseRate>
 <RateFactor>1</RateFactor>
 <ReservationData>1.71</ReservationData>
 <Resource>CPU : GHz Hour</Resource>
 </ReservationCost>
 <UsageReservationCost>
 <TotalCost>17156.25</TotalCost>
 <StartTime>Mon Aug 04 20:14:00 GMT+05:30 2008</StartTime>

```
<EndTime>Fri Sep 25 20:14:00 GMT+05:30 2009</EndTime>
<Entity>MyVM</Entity>
<Parent>MyHost</Parent>
<TotalDuration>36028800</TotalDuration>
<AttributionPercentage>1.0</AttributionPercentage>
<BaseRate>1.0000</BaseRate>
<RateFactor>1</RateFactor>
<ReservationData>1.71</ReservationData>
<UsageReservationData>1.23</UsageReservationData>
<Resource>CPU : GHz Hour</Resource>
<TotalMissingDataDuration>0</TotalMissingDataDuration>
</UsageReservationCost>
</CostDetails>
</EntityCostDetails>
</DataDetail>
</DataDetails>
</Report>
</Reports>
</ReportTemplate>
</ReportTemplates>
</Response>
```

Sample Java Code Using vCenter Chargeback APIs

```
package com.vmware.vcenter.chargeback.sample.api.rest;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
import org.apache.commons.httpclient.HttpException;
import org.apache.commons.httpclient.HttpMethod;
import com.vmware.vcenter.chargeback.sample.api.Cleanable;
import com.vmware.vcenter.chargeback.sample.api.ClientAPIBase;
import com.vmware.vcenter.chargeback.sample.exception.InvalidResponseException;
import com.vmware.vcenter.chargeback.sample.javabindings.JavaBindingLayerFactory;
import com.vmware.vcenter.chargeback.sample.session.SessionManager;
import com.vmware.vcenter.chargeback.sample.utils.ClientConstants;
import com.vmware.vcenter.chargeback.sample.utils.CommonUtils;
public abstract class RestClientAPIBase implements ClientAPIBase, Cleanable {
    private Map<String, String> urlParameters = new HashMap<String,
String>();
    private RestResponse<?> actualResponse;
    private HttpMethod httpMethod = null;
    private RestErrorType<?> expectedError = null;
    private boolean validateAPISpecificInfo = true;
    protected RestClientAPIBase(HttpMethod httpMethod) {
        this.httpMethod = httpMethod;
        addUrlParameter(ClientConstants.URL_VERSION_PARAMETER,
ClientConstants.VERSION_1_0_1);
    }
    protected RestClientAPIBase(HttpMethod httpMethod, RestErrorType<?>
expectedError) {
        this(httpMethod);
        this.expectedError = expectedError;
    }
    public Map<String, String> getUrlParameters() {
        return urlParameters;
    }
    public void addUrlParameter(String name, String value) {
        urlParameters.put(name, value);
    }
    public boolean isValidateAPISpecificInfo() {
        return validateAPISpecificInfo;
    }
    public void setValidateAPISpecificInfo(boolean validate) {
        this.validateAPISpecificInfo = validate;
    }
    private boolean validateAPISpecificResponse(RestResponse<?> responseObj) {
        if (!validateAPISpecificInfo) {
            return true;
        }
        // Must override to do validation. Otherwise, API validation fails.
        return validateResponse(responseObj);
    }
    @Override
    public RestResponse<?> execute() {
        try {
            String responseStr =
SessionManager.getInstance().getSession().execute(this);
```

```

        actualResponse = extractResponse(responseStr);
        if (actualResponse.isFailed()) {
            RestErrorType<?> error =
JavaBindingLayerFactory.getInstance().getJavaBindingLayerInstance().extractError(ac
tualResponse);
            if (expectedError == null) {
                CommonUtils.throwAPIFailedException(httpMethod, "The api failed
with an error while success is expected", error);
            }
            if (!CommonUtils.validateErrorResponse(actualResponse,
expectedError)) {
                CommonUtils.throwAPIFailedException(httpMethod, "The error
response validation for the following operation failed", error);
            }
            } else if (!validateAPISpecificResponse(actualResponse)) {
                throw new
InvalidResponseException(CommonUtils.formExceptionMessage(httpMethod, "The response
validation for the following operation failed"), actualResponse);
            }
        } catch (HttpException httpExp) {
            CommonUtils.throwOperationFailedException(httpMethod, "Could not
execute the following operation", httpExp);
        } catch (IOException ioExp) {
            CommonUtils.throwOperationFailedException(httpMethod, "Could not
execute the following operation", ioExp);
        }
        return actualResponse;
    }
    public void setHttpMethod(HttpMethod httpMethod) {
        this.httpMethod = httpMethod;
    }
    public HttpMethod getHttpMethod() {
        return httpMethod;
    }
    public RestResponse<?> getActualResponse() {
        return actualResponse;
    }
    public abstract String getRelativeUrl();
}
package com.vmware.vcenter.chargeback.sample.api.rest.xml;
import org.apache.commons.httpclient.HttpMethod;
import com.vmware.vcenter.chargeback.sample.api.rest.RestClientAPIBase;
import com.vmware.vcenter.chargeback.sample.api.rest.RestErrorType;
import com.vmware.vcenter.chargeback.sample.api.rest.RestResponse;
import com.vmware.vcenter.chargeback.sample.javabindings.JavaBindingLayerFactory;
public abstract class RestXmlClientBaseAPI extends RestClientAPIBase {
    protected RestXmlClientBaseAPI(HttpMethod httpMethod) {
        super(httpMethod);
    }
    protected RestXmlClientBaseAPI(HttpMethod httpMethod,
RestErrorType<?> expectedError) {
        super(httpMethod, expectedError);
    }
    public RestResponse<?> extractResponse(String response) {
        return JavaBindingLayerFactory.getInstance()
            .getJavaBindingLayerInstance().unmarshallResponse(response);
    }
}

```

```

}
package com.vmware.vcenter.chargeback.sample.api.rest.rbac;
import org.apache.commons.httpclient.methods.PostMethod;
import com.vmware.vcenter.chargeback.sample.api.rest.RestErrorType;
import com.vmware.vcenter.chargeback.sample.api.rest.RestRequest;
import com.vmware.vcenter.chargeback.sample.api.rest.RestResponse;
import com.vmware.vcenter.chargeback.sample.api.rest.xml.RestXmlClientBaseAPI;
import com.vmware.vcenter.chargeback.sample.javabindings.JavaBindingLayerFactory;
import com.vmware.vcenter.chargeback.sample.javabindings.jaxb.v1_0_1.ResponseType;
import com.vmware.vcenter.chargeback.sample.utils.ClientConstants;
public class RestLoginAPI extends RestXmlClientBaseAPI {
    private static final String LOGIN_URL_SUFFIX = "/login";
    private String userName;
    private String password;
    public String getRelativeUrl() {
        return LOGIN_URL_SUFFIX;
    }
    public RestLoginAPI(String userName, String password, RestErrorType<?> error) {
        super(new PostMethod(), error);
        setUserName(userName);
        setPassword(password);
    }
    @Override
    public RestRequest formRequest() {
        return JavaBindingLayerFactory.getInstance()
            .getJavaBindingLayerInstance().formLocalUserLoginRequest(
                getUserName(), getPassword());
    }
    @Override
    public boolean validateResponse(RestResponse<?> responseObj) {
        if (responseObj != null) {
            ResponseType response = (ResponseType) responseObj
                .getResponseObject();
            if (response.getStatus().value().equals(ClientConstants.SUCCESS)) {
                return true;
            }
        }
        return false;
    }
    @Override
    public void cleanup() {
        // TODO Auto-generated method stub
    }
    public String getUserName() {
        return userName;
    }
    public void setUserName(String userName) {
        this.userName = userName;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
}

```

Sample Java Code using vCenter Chargeback APIs

```
package com.vmware.vcenter.chargeback.sample.api.rest.rbac;
import org.apache.commons.httpclient.methods.PutMethod;
import com.vmware.vcenter.chargeback.sample.api.rest.RestRequest;
import com.vmware.vcenter.chargeback.sample.api.rest.RestResponse;
import com.vmware.vcenter.chargeback.sample.api.rest.xml.RestXmlClientBaseAPI;
import com.vmware.vcenter.chargeback.sample.javabindings.JavaBindingLayerFactory;
import com.vmware.vcenter.chargeback.sample.utils.rbac.RbacConstants;
/**
 * @author ssakdeo
 */
public class RestLogoutAPI extends RestXmlClientBaseAPI {
    public RestLogoutAPI() {
        super(new PutMethod());
    }
    @Override
    public String getRelativeUrl() {
        return RbacConstants.LOGOUT_URL_SUFFIX;
    }
    @Override
    public RestRequest formRequest() {
        return null;
    }
    @Override
    public boolean validateResponse(RestResponse<?> responseObj) {
        return
JavaBindingLayerFactory.getInstance().getJavaBindingLayerInstance().isResponseSuccessful(responseObj);
    }
    @Override
    public void cleanup() {
    }
}
```