



# **Portunity ISPSuite Webrequests**

*Version 1.5 [03.06.2014]*

## **API-Programmierhandbuch für Endkunden (Deutsch)**

**PORTUNITY**

Portunity GmbH  
Werner Seelenbinder Str.23  
42477 Radevormwald  
Germany

<http://www.portunity.de>

---

Copyright © 2001-2007 by Portunity GmbH, Germany

Jede nicht genehmigte Vervielfältigung dieses Handbuches sowie der Portunity ISPSuite Software wird strafrechtlich verfolgt. Die Rechte an dieser Dokumentation und die Rechte an der ISPSuite Software liegen ausschließlich bei der Portunity GmbH.

Warenzeichen werden ohne Gewährleistung der freien Verwendbarkeit benutzt. „Portunity“, „Portalsuite“ und „ISPSuite“ sind eingetragene Warenzeichen der Portunity GmbH Deutschland. „Microsoft“, „Windows“ und „Internet Explorer“ sind eingetragene Warenzeichen der Microsoft Corporation USA.

Die in diesem Handbuch abgedruckten Beispielprogramme und –Templates demonstrieren verschiedene Eigenschaften der ISPSuite. Sie dienen ausschließlich zu Lernzwecken. Sie dürfen diese Beispielprogramme und –Templates bearbeiten und für den eigenen Gebrauch modifizieren. Sie dürfen die Beispielprogramme und –Templates jedoch nicht als Ganzes oder in Teilen in kompilierter Form oder als Quelltext weitergeben oder verkaufen. Sie dürfen die Beispielprogramme und –Templates in eigenen Anwendungen verwenden, solange die resultierenden Programme nicht einem Beispielprogramm bzw. –Template oder einem Teil eines Beispielprogrammes bzw. –Templates in Erscheinung oder Funktion gleichen.

Es wird ferner keine Garantie für die Richtigkeit und Vollständigkeit des Inhaltes dieses Handbuches übernommen. Da sich Fehler, trotz aller Bemühungen, nie vollständig vermeiden lassen, sind wir für Hinweise jederzeit dankbar.

Herausgeber:

Portunity GmbH  
Werner Seelenbinder Str.23  
42477 Radevormwald  
Germany

Telefon: 49 (0)202 / 69555-0  
Fax: 49 (0)202 / 69555-190  
eMail: [info@portunity.de](mailto:info@portunity.de)  
Internetseite Providing Deutschland: <http://www.portunity.net>

Dokumentation: Björn Rücker, Alexander Reetz, Emanuel Heymann  
Korrektur und Qualitätssicherung: Bernd Schnell und Christian Dahmen  
Satz und typographische Gestaltung: Björn Rücker  
Bilder: Björn Rücker

---

## Inhaltsverzeichnis

<b>1</b>	<b>EINFÜHRUNG UND ÜBERBLICK IN DIE WEBREQUESTS.....</b>	<b>4</b>
1.1	ZIELGRUPPE .....	4
1.2	ALLGEMEINE FUNKTIONSWEISE.....	5
1.3	HTTP-REQUEST UND –ANTWORT .....	6
1.4	SERVERDATEN UND URL's .....	7
1.5	Globale Webrequest-Funktion in PHP4 / PHP5.....	7
<b>2</b>	<b>DOKUMENTATION DER WEBREQUESTS .....</b>	<b>11</b>
2.1	PRODUCT-DSL.....	11
2.2	PRODUCT-VOIP.....	14

---

# 1 Einführung und Überblick in die Webrequests

Die ISPSuite stellt die zentrale Datenbank und das Webinterface der Portunity GmbH dar. Die ISPSuite bietet den Kunden mit den Webrequests eine Schnittstelle, mit denen einzelne Objekte und deren Eigenschaften (Daten) in eigenen Websites und Webinterfaces aus der ISPSuite abgerufen werden können sowie einzelne Aktionen ausgeführt werden können (z.B. Telefonanrufe bei VoIP initiieren u.a.). Die bereitgestellten Funktionen sind je nach Webrequest und Objekt sehr unterschiedlich.

Grundsätzlich funktioniert die Kommunikation zwischen Ihren Applikationen und der ISPSuite über „normale“ http-Verbindungen auf Port 80 und kann aus quasi jeder Programmiersprache verwendet werden, welche eine solche http-Socketverbindung absetzen kann. Insbesondere geht dies sehr einfach in den beliebten „Web-Sprachen“ PHP, ASP und Perl, aber auch aus den „klassischen Hochsprachen“ wie C usw. können natürlich http-requests abgesetzt und ausgewertet werden.

In diesem Kapitel möchten wir die vorhandenen Schnittstellen im einzelnen vorstellen.

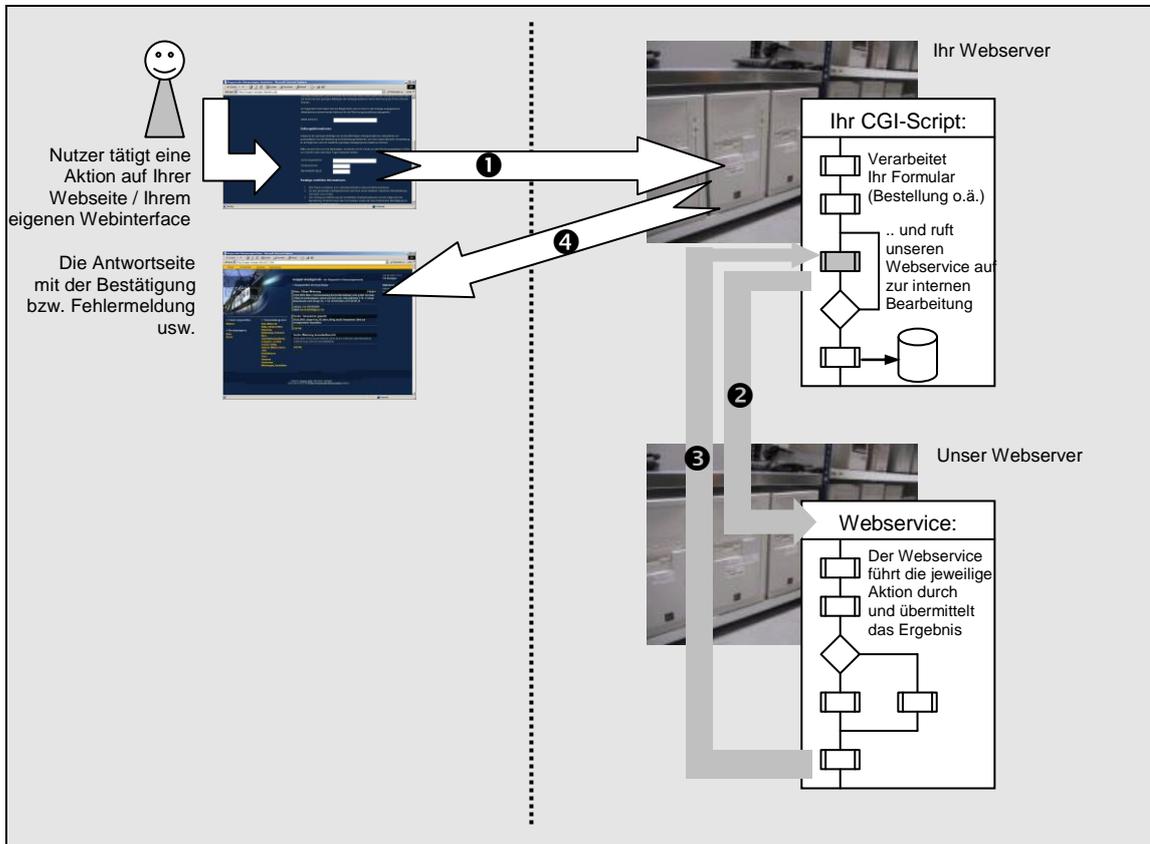
## 1.1 Zielgruppe

Die Webrequest-API richten sich als Schnittstelle an unsere Endkunden und deren technische Systemintegratoren (Systemhäuser, Inhouse-Technik u.a.) sowie an Softwareentwickler, die Portunity-Produkte in eigenen Applikationen einbinden und implementieren möchten.

Die Webrequest-API richtet sich NICHT an unsere Reseller, für diese steht eine wesentlich umfangreichere Webservice-API zur Verfügung, welche nicht mit der Webrequest-API zu verwechseln ist.

## 1.2 Allgemeine Funktionsweise

Der Ablauf der verschiedenen Webrequests ist prinzipiell immer identisch, z.B. der Abruf von Statistiken eines DSL-Zugangs. Das Zusammenspiel der einzelnen Systeme haben wir in Abbildung 1 dargestellt:



**Abbildung 1: Allgemeine Funktionsweise der Webrequests und des Zusammenspiels der Systeme**

Der Nutzer tätigt auf Ihrer Webseite oder Ihrem eigenen Webinterface eine Aktion, z.B. er will eine aktuelle Statistik eines DSL-Zugangs darstellen oder ähnliches. Durch Klick auf den Absendenbutton werden die Daten zusammen mit einer Anforderung einer Bearbeitungs-CGI-Seite zu Ihrem Server / Workspace-Account Ihres Providers übertragen (1).

Die Daten werden auf Ihrem Server wie gewohnt von Ihrem CGI-Script (ASP, PHP o.a.) in Empfang genommen. Irgendwo innerhalb Ihres Scriptes befindet sich dann allerdings der Teil wo die Aktion von unserer Seite benötigt wird. Ihr Script setzt nun einen Request – für den Kunden völlig unsichtbar – an unseren Webserver an das Webrequest-Script ab und übermittelt die jeweilig notwendigen Daten um die gewünschte Aktion (z.B. neuen Unterkunden anlegen oder Transaktion X einstellen usw.) auszuführen. (2).

Unser Server überprüft die erhaltenen Daten, führt die gewünschte Aktion aus (bzw. im Fehlerfall wenn z.B. die dazu nötigen Daten unvollständig sind auch nicht) und erzeugt eine Antwortseite, welche an Ihr Script auf Ihrem Server zurückgesendet wird und neben einer Information ob die Aktion erfolgreich war oder nicht auch evt. angeforderte Daten zurückliefert (3).

Die erhaltene Antwortseite kann dann von Ihnen ausgewertet werden und je nach erhaltenen Informationen können von Ihrem Script die Daten weiterverarbeitet werden oder der Nutzer kann entsprechend von Ihrem Script informiert werden usw. Ihr Script schickt dann entsprechend eine Antwortseite für den weiteren Verlauf an den Nutzer als Antwort auf das ausgefüllte Formular (4).

### 1.3 Http-Request und –Antwort

Um Daten aus der ISPSuite über einen Webrequest abzurufen, müssen Sie einen sogenannten http-Request gegen unseren Webserver absetzen und je nach Webrequest unterschiedliche Daten übermitteln. Für jeden Webrequest gibt es jeweils eine URL die entsprechend angesprochen werden muß.

Zur Übermittlung von Parametern gibt es zwei Verfahren, welche Sie wahrscheinlich bereits von der Formularverarbeitung von HTML und CGI-Scripten kennen: GET und POST. Beim GET-Verfahren werden die Parameter an die URL in einer speziellen Syntax angehängt, beim POST-Verfahren werden die Parameter in dem Header des Http-Requests – für den Nutzer unsichtbar – mitgesendet.

Bei den Webrequests ist es egal, ob Sie die Daten im GET oder im POST-Verfahren übertragen.

Ein http-Request im GET-Verfahren unterliegt einer technischen Längenbeschränkung, die jedoch für die Webrequests i.d.R. keine so große Rolle spielt, da nur sehr wenige Parameter übertragen werden müssen.

Des weiteren wichtig für die Kommunikation ist der Zeichensatz der Kodierung, die ISPSuite verwendet derzeit standardmässig noch „ISO8859-1“, wobei wir mittelfristig planen auch UTF8 zu unterstützen. Die UTF8-Unterstützung wird dann jedoch über spezielle zusätzliche Parameter steuerbar sein.

Die Rückgabewerte sind im ASCII-Modus folgendermaßen formatiert: Je Zeile (Trennzeichen \n) steht grundsätzlich ein Rückgabewert. Der Name des Rückgabewertes steht vorne. Rückgabename und Rückgabewert sind von einem Doppelpunkt, gefolgt von einem Leerzeichen getrennt. Leerzeilen sind zu ignorieren. Bitte beachten Sie, daß die einzelnen Zeilen mit einem Zeilenumbuch versehen sind (\n). Falls Sie den Request über einen Webbrowser abgesetzt haben, werden Sie das Ergebnis natürlich in einer langen Zeile hintereinander sehen, da die Zeilenumbruch-Tags (<br>) nicht vorhanden sind !

Bei jedem Webrequest gibt es die folgenden globalen Rückgabewerte, welche in den einzelnen Webrequest nicht nochmal extra aufgeführt sind. Die globalen Rückgabewerte fangen alle mit dem Prefix „WR\_“ an:

Keyname:	Beschreibung und Informations:
WR_Status	Ob der Webrequest-Aufruf / -Ausführung grundsätzlich erfolgreich war oder nicht. Mögliche Werte:  ok = war erfolgreich  FAILED = war NICHT erfolgreich
WR_Comment	Kurzer Informationstext zusätzlich zum Status (Fehlermeldung)

	oder sonstige Information)
WR_Warning (Optional, evt. auch mehrmals)	Warnungstext, wird ausgegeben z.B. wenn unbekannte Felder übergeben wurden oder es zu sonstigen nicht-kritischen Verhalten kommt.
WR_Version	Versionsnummer des Webrequests im Format (x.y), wobei x=Hauptversionsnummer und y=unterversionsnummer. Jedes Webrequestmodul hat eine eigene Versionsnummer!
WR_RequestIP	IP-Adresse von der der Webrequest aufgerufen wurde (Normalerweise Ihre Server-IP-Adresse)
WR_RequestDateTime	Datum und Uhrzeit des ISPSuite-Webrequest-Servers im Format YYYY-MM-DD SS:MM:SS

## 1.4 Serverdaten und URL's

Bitte verwenden Sie folgende Serverdaten und URL's um die http-Requests abzusetzen:

Keyname:	Beschreibung und Informations:
Servername:	wr.ispsuite.portunity.de
Port:	80
Http-Request	GET oder POST
Verzeichnis und Filename	/webrequests/XXXXX  (wobei XXXX der Name des jeweiligen Webrequest-Modules darstellt, z.B. „product-dsl“ wenn Sie Infos zu Ihrem DSL-Zugang abrufen möchten !)

## 1.5 Globale Webrequest-Funktion in PHP4 / PHP5

In der Sprache PHP kann der Webrequest besonders elegant und einfach angesteuert werden. Hier eine globale Funktion die für alle Webrequest der ISPSuite verwendet werden kann und welche Sie entweder in Ihrem Hauptprogramm oder in einer separaten PHP-Datei, welche Sie dann per „include“ einbinden, ablegen können.

Die Funktion baut den http-Request mit den Parametern zusammen und setzt diesen ab, parst anschließend die zurückgegebenen Informationen und überführt die Rückgabewerte in ein Array. Die genaue Funktionsweise dieser Funktion haben wir in den Kommentaren innerhalb des Codes erläutert.

Die Funktion kann relativ einfach auch in andere Sprachen wie ASP, Perl usw. portiert sowie um eigene Funktionalitäten (z.B. Queueing in einer Tabelle falls Verbindung nicht möglich sein sollte mit entsprechender Wiederholung u.ä.) erweitert werden.

**Nach der Funktion befindet sich ein kleines Anwendungs-Beispiel für den Webrequest „“ um alle Daten xxxx auszulesen und anzuzeigen.**

Bitte fragen Sie unseren Support nach der PHP-Datei – sie müssen dieses Beispiel nicht abtippen ;-)

```
<?
/**
 * Funktion ISPSuiteWebrequest_MakeRequest
 *
 * Diese Funktion setzt einen http-Request gegen die ISPSuite-Webreqst-API ab und gibt
 * die Antwortwerte in einem Array zurück.
 * @param string $sWebrequestname
 * @param array $a_sData
 * @return array
 */
function ISPSuiteWebrequest_MakeRequest($sWebrequestname, $a_sData)
{
    // #####
    // ### Vorbereitende Tätigkeiten, Default-Values setzen          ###
    // #####
    // ### Default-Values
    $sWebrequestHost = "wr.ispsuite.portunity.net";
    // ### übergebens Array in http-Getdaten-String konvertieren
    $sGetdata = "";

    foreach ($a_sData as $sAktKey => $sAktValue)
    {
        // ### leere Keys nicht mitsenden
        if (trim($sAktKey) != "")
        {
            // ### beim ersten Value kein Trennzeichen
            if ($sGetdata != "")
                $sGetdata .= "&";

            // ### Akt. Key und Value http-kodiert an Poststring anhängen
            $sGetdata .= urlencode($sAktKey)."=".urlencode($sAktValue);
        }
    } // foreach: Schleife über übergeben Post-Array zur Konvertierung
    // #####
    // ### http-Request absetzen und Antwort auslesen          ###
    // #####

    // ### Request-URL zusammenbauen
    $sRequestURL =
"http://".$sWebrequestHost."/webrequests/".$strtolower($sWebrequestname)."/";

    // ### Prüfen Curl auf dem Server verfügbar ist
    if (function_exists('curl_init') == true)
    {
        // ### Session initialisieren
        $ch = curl_init();

        // ### CURLOPT_URL hinzufügen
        curl_setopt($ch, CURLOPT_URL, $sRequestURL);
    }
}
```

```
curl_setopt($ch, CURLOPT_RETURNTRANSFER,1);
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $sGetdata);

// ### Request aufrufen
$sCurlResult = curl_exec($ch);

// ### Ergebnis Zeilenweise in Array schreiben
$a_sLines = explode ("\n", $sCurlResult);

//## Session beenden
curl_close($ch);
}
else
{
// ### Parameter für file() an URL anhängen
$sRequestURL .= "?".$sGetdata;

// ### URL PER file() aufrufen
// ### Request absetzen und Antwort auslesen
$a_sLines = file($sRequestURL);
}

// #####
// ### http-Antwort in Rückgabe-Array überführen #####
// #####
// ### ... und jede Zeile einzeln durchscannen
for ($iFor1=0; $iFor1<sizeof ($a_sLines) ; $iFor1++)
{
// ### ... aber nur Zeilen mit Inhalt bearbeiten
if (substr_count($a_sLines[$iFor1],":") > 0)
{
// ### Akt. Zeile in Variablenname und -value splitten
$a_sAktLine = explode (":",$a_sLines[$iFor1],2);
// ### und in unser neues Array schreiben
$a_sResult [trim($a_sAktLine[0])] = trim($a_sAktLine[1]);
} // if: nur Zeilen mit Inhalt bearbeiten
} // if: jede Zeile durchscannen

// ### Das fertig zusammengebaute Array zurückgeben
return ($a_sResult);

} // function ISPSuiteWebrequest_MakeRequest

// #####
// ### Beispielaufruf für Webservice "SMS-Versand" #####
// #####

// ### Verschickt eine SMS

// ### Authentifizierung gegenüber Webservice
// ### mit den Zugangsdaten des VoIP-Produktes
$a_sPostdata ["sProductLogin"] = "sip0001";
$a_sPostdata ["sProductCode"] = "abcdef";

// ### Parameter für den Webservice
$a_sPostdata ["sCommand"] = "SendSMS";
```

```
$a_sPostdata ["sSMSText"] = "Schon mal eine SMS gelesen?";  
$a_sPostdata ["sSMSNumber"] = "0173xxxxx";  
  
// ### Webservice ausführen  
$a_sResult = ISPSuiteWebrequest_MakeRequest ("product-voip", $a_sPostdata);  
  
// ### und Ergebnis anzeigen  
print_r ($a_sResult);  
  
?>
```

## 2 Dokumentation der Webrequests

### 2.1 Product-DSL

Mit dem Webrequest „Product-DSL“ können Sie Informationen über Ihren DSL-Zugang auslesen, z.B. die Einzelverbindungsdaten oder sich die grafischen Statistiken (Stichwort MRTG / RRD) zurückgeben lassen.

<b>Webrequest-Name:</b> <i>(Case-Sensitive !)</i>	<b>product-dsl</b>
--	--------------------

Variablen die übergeben werden müssen:

<b>Keyname:</b>	<b>Beschreibung und Informations:</b>
sProductLogin	Loginname des DSL-Zugangs, z.B. sd2/dsl-portunity.net-adsl0001
sProductCode	Zum Loginnamen des DSL-Zugangs zugehörige Kennwort.
sCommand	Befehl den Sie zu diesem DSL-Zugang absetzen möchten, zur Verfügung stehen:  <ul style="list-style-type: none"> <li>- Connections: Gibt abgeschlossene Verbindungen zurück weitere Parameter optional möglich</li> <li>- RRDImage: Gibt eine binäre Grafik der RRD-Datei der Traffic- oder Paketauswertung zurück – weitere Parameter benötigt</li> </ul>

Bei einem Request nach dem Befehl „Connections“ können zusätzlich folgende Parameter übergeben werden:

<b>Keyname:</b>	<b>Beschreibung und Informations:</b>						
sCommandPeriod <i>(Optional)</i>	Periode von der die Verbindungsdaten zurückgegeben werden sollen, mögliche Werte:  <table style="width: 100%; border: none;"> <tr> <td style="width: 150px;">NotCalced <i>(Default)</i></td> <td>Gibt alle noch nicht abgerechneten abgeschlossenen Verbindungen zurück – unabhängig davon ob diese bereits übertragen wurden oder noch nicht</td> </tr> <tr> <td>Today</td> <td>Gibt alle abgeschlossenen Verbindungen des heutigen Tages zurück – unabhängig davon ob diese bereits übertragen wurden oder noch nicht</td> </tr> <tr> <td>Yesterday</td> <td>wie Today, nur vom gestrigen Tag</td> </tr> </table>	NotCalced <i>(Default)</i>	Gibt alle noch nicht abgerechneten abgeschlossenen Verbindungen zurück – unabhängig davon ob diese bereits übertragen wurden oder noch nicht	Today	Gibt alle abgeschlossenen Verbindungen des heutigen Tages zurück – unabhängig davon ob diese bereits übertragen wurden oder noch nicht	Yesterday	wie Today, nur vom gestrigen Tag
NotCalced <i>(Default)</i>	Gibt alle noch nicht abgerechneten abgeschlossenen Verbindungen zurück – unabhängig davon ob diese bereits übertragen wurden oder noch nicht						
Today	Gibt alle abgeschlossenen Verbindungen des heutigen Tages zurück – unabhängig davon ob diese bereits übertragen wurden oder noch nicht						
Yesterday	wie Today, nur vom gestrigen Tag						

	MonthThis	wie Today, nur vom aktuellen Monat
	MonthLast	wie Today, nur vom vorherigen Monat

Bei der Infosignatur „Connections“ erfolgt die Rückgabe der Verbindungen (neben den globalen Rückgabekeys) im folgenden Format:

Keyname:	Beschreibung und Informations:
Connection_X: <i>(Nur bei WS_Status=OK !)</i>	<p>Die einzelnen Verbindungen werden jeweils als ein Key pro Verbindung zurückgegeben, wobei X im Keyname durch eine fortlaufende Nummer ersetzt wird. Verbindungslog-Datensatz im semikolon-separiertem Format:</p> <p>IP;Start;Ende;Sekunden;Bytes Out;Bytes IN;Packets Out;Packets In;Termcause</p> <p>Wobei:</p> <p>IP: IP-Adresse der Verbindung (bei statischer IP-Adresse immer die gleiche, ansonsten die der Verbindung zugewiesene dynamische IP-Adresse)</p> <p>Start: Start-Zeitpunkt, sekundengenau im mysql-Format („yyyy-mm-dd hh:mm:ss“)</p> <p>Ende: End-Zeitpunkt, sekundengenau im mysql-Format („yyyy-mm-dd hh:mm:ss“)</p> <p>Sekunden: Anzahl der Sekunden der Verbindung (Ende - Start)</p> <p>Bytes Out: Anzahl der gesendeten Bytes</p> <p>Bytes In: Anzahl der empfangenen Bytes</p> <p>Packets Out: Anzahl der gesendeten Pakete</p> <p>Packets In: Anzahl der empfangenen Pakete</p> <p>Termcause: Grund der Verbindungstrennung</p>

Bei einem Request mit dem Befehl „RRDImage“ können zusätzlich einige der folgenden Parameter übergeben werden:

Keyname:	Beschreibung und Informations:
sCommandRRDType <i>(Optional)</i>	Typ der RRD-Datei, zur Verfügung stehen: „traffic“:        Trafficauswertung (=Default) „packets“:       Paketauswertung
sCommandPeriod <i>(Optional)</i>	Zeitraum von dem die Grafik erstellt werden soll, zur Verfügung stehen: „day“:            Tagesgrafik (=Default) „week“:          Wochengrafik „month“:         Monatsgrafik „year“:          Jahresgrafik
iCommandWidth <i>(Optional)</i>	Breite der zurückzulieferenden Grafik-Datei. Standardbreite ist 400 Pixel. Bitte beachten Sie, dass bei zu kleiner Breite einige Texte nicht mehr vernünftig dargestellt werden können.
iCommandHeight <i>(Optional)</i>	Höhe der zurückzulieferenden Grafik-Datei. Standardhöhe ist 125 Pixel. Bitte beachten Sie, dass bei zu kleiner Höhe einige Texte nicht mehr vernünftig dargestellt werden können.

Bei einem Request mit dem Befehl „RRDImage“ erfolgt bei Erfolg die Rückgabe direkt als binäre Datendatei zur Speicherung oder direkten Ausgabe im Browser, ansonsten folgen die üblichen globalen Rückgabekeys.

Achtung: Wir bitten darum diese Schnittstelle nicht dazu zu verwenden alle RRD-Grafiken aller Zugänge regelmässig zu ziehen (z.B. alle 5 Minuten oder 1 x pro Stunde) da dies eine sehr große Last für unseren Webserver auf dem der Webrequest-Dienst läuft sowie zudem sehr viel unnötigen Traffic bedeuten würde (den nicht nur wir sondern letztendlich auch Sie irgendwo bezahlen müssen). Bei 1 Request pro Std. wären das immerhin 120 Kbyte (8 Grafiken à 15 KByte) bei 100 Zugängen = 11,72 MByte \* 24 Std. / Tag = 281 MByte / Tag, im 5 Minuten-Takt wären das bereits 3,3 GByte pro Tag (wenn das 100 Kunden machen...). Die Schnittstelle ist dazu gedacht Ihnen die Grafiken für eigene Seiten (z.B. Ihr Firmen-Intranet) zu präsentieren und wir gehen hier von üblichen Besuchsverhalten aus.

Wir behalten uns bei Missbrauch oder zu hoher Last jederzeit eine Sperrung der Webrequest-API vor !

## 2.2 Product-VoIP

Mit dem Webrequest „Product-VoIP“ können Sie Informationen über Ihren VoIP-Account auslesen, z.B. die Einzelverbindungsdaten. Außerdem können Sie Telefonverbindungen über den Webrequest initiieren.

<b>Webrequest-Name:</b> <i>(Case-Sensitive !)</i>	<b>product-voip</b>
--	---------------------

Variablen die übergeben werden müssen:

Keyname:	Beschreibung und Informations:
sProductLogin	Loginname des VoIP-Accounts, z.B. „sip0001“ oder „iax0001“
sProductCode	Zum Loginnamen des VoIP-Accounts zugehörige Kennwort.
sCommand	<p>Befehl den Sie zu diesem VoIP-Account absetzen möchten, zur Verfügung stehen:</p> <ul style="list-style-type: none"> <li>- Connections: Gibt abgeschlossene Verbindungen zurück weitere Parameter optional möglich</li> <li>- Connect: Wählt eine Zielrufnummern und stellt eine Verbindung zum Sip-/IAX-Account oder einer anderen Zielrufnummer her. Die beiden Enden der Kommunikation werden als „links“ und „rechts“ bezeichnet, wobei zur „linken“ Kommunikationsseite die Verbindung zuerst aufgebaut wird. – weitere Parameter benötigt</li> </ul> <p><b>Wichtiger Hinweis zum Befehl „Connect“:</b> <i>Wir machen an dieser Stelle auf eigentlich etwas selbstverständliches aufmerksam: Diese Funktion darf nicht dazu verwendet werden, bei den Zielnummern unerwünschte Gespräche zu initiieren. Insbesondere „Stalking“- und „Spam“- oder auch „Spit“-sowie „Werbe“-Anrufe sind strikt untersagt. Alle Transaktionen und Verbindungsaufbauten werden protokolliert. Missbräuche führen unter Umständen nicht nur eine Accountsperre sondern auch strafrechtliche Anzeigen und Ermittlungen nach sich!</i></p> <ul style="list-style-type: none"> <li>- SendSMS: Verschickt eine SMS an eine deutsche Mobilfunknummer. – weitere Parameter benötigt</li> <li>- SendFax: Verschickt eine PDF-Datei per Fax an eine beliebige weltweite Faxnummer. – weitere Parameter benötigt</li> </ul>

Bei einem Request mit dem Befehl „Connect“ müssen zusätzlich folgende Parameter übergeben werden:

Keyname:	Beschreibung und Informations:
sCommandLeftProtocol	<p>Protokoll der linken Kommunikationsseite, mögliche Werte:</p> <ul style="list-style-type: none"> <li>– „Account“</li> <li>– „TEL“ (auch „TEL_MOBILE“ und „TEL_NATIONAL“)</li> <li>– „SIP“</li> <li>– „IAX“</li> </ul> <p>Bei „Account“ wird die Verbindung per SIP oder IAX zu dem VoIP-Account aufgebaut, der durch das Produkt abgebildet wird.</p> <p>Bei „TEL“ wird die Verbindung zu einer Telefonnummer aufgebaut, welche im Destination-Parameter definiert werden muß. Durch Netzcodes können im Destination-Parameter auch Optionen für den Call aktiviert oder deaktiviert werden (z.B. ENUM-Lookup ausschalten, Auswahl von Premiumrouting u.a.). Durch diese Option entstehen Kosten für den Anruf die dem Account belastet werden ; bei Prepaidaccounts muß ausreichend Deckung vorhanden sein !</p> <p>Neben „TEL“ gibt es auch noch die Sonderformen „TEL_NATIONAL“ und „TEL_MOBILE“ – hierbei wird die Zielrufnummer zusätzlich überprüft . Bei TEL_NATIONAL sind nur Zielrufnummern im nationalen Festnetz erlaubt, bei TEL_MOBILE sind sowohl nationale Festnetz als auch Mobilfunknetznummern zugelassen. Internationale Nummern sind bei beiden Variaten nicht zulässig und werden mit einer Fehlermeldung quittiert – wohingegen bei „TEL“ auch internationale Nummern zugelassen werden.</p> <p>Bei „SIP“ und „IAX“ wird die Verbindung zu einer SIP oder IAX-Adresse aufgebaut. Die Zieladressen werden im Parameter Destination angegeben.</p>
sCommandLeftDestination	Definiert die Telefonnummer oder Zieladresse (SIP/IAX) in Abhängigkeit des vorherigen Parameters.
sCommandRightProtocol	Rechtes Gegenstück zum linken Parameter „ConnectLeftProtocol“, mögliche Werte sind auch hier „SIP“, „IAX“, „Account“ und „TEL“ (Beschreibung siehe bei „ConnectLeftProtocol“).
sCommandRightDestination	Definiert die Telefonnummer oder Zieladresse (SIP/IAX) in Abhängigkeit des vorherigen Parameters.
sCommandAnnouncement	Definiert eine Ansage die zwischen dem Annehmen der

<i>(Optional)</i>	Kommunikation durch die linken Kommunikationsseite und dem Verbindungsaufbau zur rechten Kommunikationsseite abgespielt werden soll. Wird der Parameter leer gelassen, wird eine neutrale Standardansage vom System verwendet. Ansonsten kann eine in dem Account im Ansagenmanager hinterlegte Ansage (Dateiname !) benannt werden.
-------------------	--

### Beispiel-Request als URL (GET):

```
http://wr.ispsuite.portunity.de/webrequests/product-voip/?sProductLogin=sip0001&sProductCode=secret&sCommand=Connect&sCommandLeftProtocol=TEL&sCommandLeftDestination=0202123123123&sCommandRightProtocol=TEL&sCommandRightDestination=020299999
```

Anstelle von „sip0001“ und dem Kennwort „secret“ sowie anstelle der Telefonnummern sind natürlich Accountspezifische und eigene Daten einzusetzen.

Bei erfolgreichem Request wird zuerst eine Verbindung zur Telefonnummer 0202123123123 aufgebaut und nachdem der Anruf entgegengenommen wurde wird nach einer kurzen Systemansage eine Verbindung zur Nummer 0202999999 aufgebaut.

Bei einem Request mit dem Befehl „SendSMS“ müssen zusätzlich folgende Parameter übergeben werden:

Keyname:	Beschreibung und Informations:
sSMSText	Der Text der SMS-Nachricht, minimum 2 Zeichen, maximal 160 Zeichen.
sSMSNumber	Die Zielrufnummer im Format 00-„Ländercode“, „Nummer“, aktuell werden nur Mobilfunknummern in Deutschland (49) unterstützt, eine gültige Nummer ist z.B. „0049175111xxxx“ (anstelle der x weitere Ziffern).

Bei einem Request mit dem Befehl „SendFax“ werden folgende Parameter benötigt:

Keyname:	Beschreibung und Informations:
sDestination	Die Zielrufnummer im Format 00-„Ländercode“, „Nummer“. Eine gültige Nummer ist z.B. „004920269555xxxx“ (anstelle der x weitere Ziffern). Sollte eine Ländersperre in Ihrem SIP Account eingestellt sein, greift diese natürlich auch beim Faxversand.
sFax	Die PDF-Datei, die wir als Fax verschicken.
sEmail	Eine Email-Adresse an die die Versandbestätigung geschickt wird.
mimeType	Der Mime-Typ der PDF-Datei. Da wir im Moment nur den

	PDF-Versand unterstützen geben Sie hier bitte immer „application/pdf“ an.
--	---