

InTime API – Exempel kod

Exempel kod för att anropa Intime HTTP API.

Översikt funktioner

C#

[Meddelanden](#)

[Skapa meddelande](#)

[Ta bort ett schemalagt meddelande](#)

[Ta emot status](#)

PHP

[Meddelanden](#)

[Skapa meddelande](#)

[Ta bort ett schemalagt meddelande](#)

[Ta emot status](#)

C# - Meddelanden

En generell klass för att anropa Intime API som används i samtliga exempel.

```
public class IntimeAPI
{ private string serviceurl { get; set; } private
  string user { get; set; }
  private string hash { get; set; }

  public IntimeAPI(string serviceurl, string user, string hash)
  { this.serviceurl = serviceurl;
    this.user = user;
    this.hash = hash;
  }

  public string DeleteMessage(string xml)
  { return ServiceRequest(string.Format("{0}/{1}?user={2}&hash={3}", this.serviceurl, "delete.ashx", this.user,
this.hash), xml);
  }

  public string SendMessage(string xml)
  { return ServiceRequest(string.Format("{0}/{1}?user={2}&hash={3}", this.serviceurl, "create.ashx", this.user,
this.hash), xml);
  }

  private string ServiceRequest(string url, string xml)
  { byte[] byteArray = StrToByteArray(xml);
```

```

    HttpWebRequest myReq =
    (HttpWebRequest)WebRequest.Create(url); myReq.Method = "POST";
    myReq.ContentType = "text/xml"; myReq.ContentLength =
    byteArray.Length;
    myReq.Credentials = System.Net.CredentialCache.DefaultNetworkCredentials;

    Stream newStream = myReq.GetRequestStream();
    newStream.Write(byteArray, 0, byteArray.Length);
    newStream.Close();

    HttpWebResponse myResponse = (HttpWebResponse)myReq.GetResponse();
    Stream stream = myResponse.GetResponseStream();
    StreamReader reader = new StreamReader(stream);

    string result = reader.ReadToEnd();

    return result;
  }

  private static byte[] StrToByteArray(string str){
    System.Text.UTF8Encoding encoding = new System.Text.UTF8Encoding();
    return encoding.GetBytes(str);
  }
}

```

Skapa meddelande

```

string serviceurl = "https://...";
string username = "username";
string hash = "hash";

```

```

IntimeAPI api = new IntimeAPI(serviceurl, username, hash);

```

```

StringBuilder xml_send = new StringBuilder();
xml_send.Append("<?xml version='1.0' encoding='iso-8859-1'?'>");
xml_send.Append("<message>");
xml_send.Append(" <flash>0</flash>");
xml_send.Append(" <multisms>0</multisms>");
xml_send.Append(" <senddate>2012-02-29T14:41:00</senddate>");
xml_send.Append(" <sendertitle>Simon test</sendertitle>");
xml_send.Append(" <body>Mitt test meddelande</body>");
xml_send.Append(" <statusurl>http://www.kunden.se/status.php</statusurl>");
xml_send.Append(" <recipients>");
xml_send.Append(" <recipient transid='12345678'>0703027930</recipient>"); xml_send.Append("
</recipients>");
xml_send.Append("</message>");

```

```

string response = "";
response = api.SendMessage( xml_send.ToString() );

```

```

// TODO Handel the response... See API documentation...

```

Ta bort ett schemalagt meddelande

```

string serviceurl = "https://...";
string username = "username";
string hash = "hash";

```

```

IntimeAPI api = new IntimeAPI(serviceurl, username, password);

```

```
StringBuilder xml_delete = new StringBuilder();
xml_delete.Append("<?xml version='1.0' encoding='iso-8859-1'?'>");
xml_delete.Append("<message><messageid>2319974</messageid></message>");
```

```
string response = "";
response = api.DeleteMessage(xml_delete.ToString());
```

```
// TODO Handel the response... See API documentation...
```

Ta emot status

Skapa en.ashx handler som tolkar ett http post request från intimes servrar.

```
public class Callback : IHttpHandler{ public void
    ProcessRequest(HttpContext context)
    {
        string messageid =
            context.Request.Params.Get("messageid"); string status =
            context.Request.Params.Get("status"); string transid =
            context.Request.Params.Get("transid");

        // TODO ADD My internal logic to save status callbacks
    }

    public bool IsReusable
    {
        get
        {
            return false;
        }
    }
}
```

PHP - Meddelanden

En generell klass för att anropa Intime API som används i samtliga exempel.

```
<?php class
IntimeAPI {
    private $serviceurl;
    private $username;
    private $hash;
    private $certPath;

    public function __construct( $serviceurl, $username, $hash, $certPath = ""
    ){
        // Set parameters
        $this->serviceurl = $serviceurl;
        $this->username = $username;
        $this->hash = $hash;
        $this->certPath = $certPath;
    }

    public function DeleteMessage($xml){ return self::ServiceRequest( $this->serviceurl."/delete.ashx?user=".$this-
    >username."&hash=".$this->hash,
$xml );
}
```

```

public function SendMessage( $xml){ return self::ServiceRequest( $this->serviceurl."/create.ashx?user=".$this->
    >username."&hash=".$this->hash,
$xml );
}

private function ServiceRequest( $url, $xml ){

    // Init request $ch =
    curl_init($url);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_CUSTOMREQUEST, 'POST' );
    curl_setopt($ch, CURLOPT_HTTPHEADER, array("Content-Type: text/xml","Content-length:
    ".strlen($xml)); curl_setopt($ch, CURLOPT_POSTFIELDS, $xml);

    // Https request
    if( strstr($url, 'https') ){
if( $this->certPath == "" ){ curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
        }else{
            $this->certPath = getcwd() . $this->certPath;
if( !file_exists($this->certPath) ){
                Throw new Exception("Could not locate certificate file! ". $this->certPath);
            }

            curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, true);
            curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 2);
            curl_setopt($ch, CURLOPT_CAINFO, $this->certPath );
        }
    }
    // Execute request
    $response = curl_exec($ch);

    // Check for error in request
    $error = curl_error($ch);
    if($error != "" ){
        Throw new Exception("Request curl error : ". $error );
    }
    // Return response
    return $response;
}
}
?>

```

Skapa meddelande

```

<?php
// Intime API server url and user information

$serviceurl = "https://...";
$username = "myusername";
$hash = "hash";

// Skapa en ny instance av API klassen
$intimeAPI = new IntimeAPI( $serviceurl,$username,$hash);

$xml = '<?xml version="1.0" encoding="iso-8859-1"?>
    <message>
        <flash>0</flash>

```

```

<multisms>0</multisms>
<senddate></senddate>
<sendertitle>My subject</sendertitle>
<body>My first message</body>
<statusurl></statusurl>
<recipients>
  <recipient transid="12345">070123456</recipient>
</recipients>
</message>;

```

```

// Send message
$response = $intimeAPI->SendMessage( $xml );

// TODO Handel the response... See API documentation...
?>

```

Ta bort ett schemalagt meddelande

```

<?php
// Intime API server url and user information

$serviceurl = "https://...";
$username = "myusername";
$hash = "hash";

// Skapa en ny instance av API classen
$intimeAPI = new IntimeAPI( $serviceurl,$username,$hash);

$xml_delete = '<?xml version="1.0" encoding="iso-8859-1"?><message><messageid>2319974</messageid></message>;

// Delete message
$response = $intimeAPI->DeleteMessage( $xml_delete );

// TODO Handel the response... See API documentation...
?>

```

Ta emot status

Skapa t.ex .en php fil som tolkar ett http post request från intimes servrar. Sidan måste vara tillgänglig på den adress som definierades när man skickade meddelandet i statusurl taggen.

```

<?php
// mysatuscallback.php

if( isset($_POST['messageid'],$_POST['status'],$_POST['transid']) ){
    // TODO add my status callback code
}else{
    header("Status: 404 Not Found");
}
}
?>

```