

# SOAP-Schnittstelle für den IBAN-Rechner

Stand: 15.3.2008

URL: <https://ssl.ibanrechner.de/soap/>

WSDL: siehe unten.

## 4 Funktionen: `validate_iban`, `calculate_iban`, `get_dutch_banks`, `get_swiss_banks`

### 1. `validate_iban`

Validiert die gegebene IBAN. Die Benutzer-ID und das Passwort sind dieselben, die auch für das Einloggen als Premium-Benutzer verwendet werden.

*Eingabe:* `validate_ibanRequest` mit den Feldern:

**iban:** xsd:string - die zu validierende IBAN.  
**user:** xsd:string - Ihre Premium-Abonnement-Benutzer-ID.  
**password:** xsd:string - Ihr Passwort.

*Ausgabe:* `tns:IBANValResStruct` mit den Feldern:

**iban:** die validierte IBAN

**result:** 'passed' für eine gültige oder 'failed' für eine ungültige IBAN

**checks:** ein Array mit den durchgeführten Tests; kann die Elemente 'length' (für eine zum Land passende Länge der IBAN), 'bank\_code' (fürs Nachschlagen der Bankleitzahl), 'account\_number' (für einen Prüfziffern-Check der Kontonummer) und 'iban\_checksum' (Check der IBAN-Prüfsumme) enthalten.

**bic\_candidates:** ein Array mit null, einem oder mehreren Elementen vom Typ `BICstruct`. Jedes `BICstruct`-Element hat die Attribute `bic`, `wwwcount`, `sampleurl` und `city`. Das Attribut `bic` enthält einen BIC (Bank Identifier Code), der eventuell zu der Bankleitzahl gehört, die in der IBAN enthalten ist. Ob das wirklich der Fall ist, kann man anhand der anderen Felder beurteilen: wenn `wwwcount` größer ist als Null, wurde dieser BIC im WWW gefunden, und zwar auf so vielen verschiedenen Seiten, wie durch `wwwcount` angegeben ist. Eine Beispiel-URL für eine Seite, wo der BIC gefunden wurde, steht im Feld `sampleurl`. Wenn `wwwcount` dagegen Null ist, stammt der BIC aus einer Liste der Nationalbank oder der EZB. Wenn eine Stadt angegeben ist, bedeutet das ebenfalls, dass der BIC aus einer Liste der Nationalbank oder EZB stammt. Die angegebene Stadt ist nicht unbedingt die, wo sich die zum Konto gehörende Zweigstelle befindet. Es kann auch die Stadt sein, wo die Bank ihre Zentrale hat.

**country:** der ISO-Ländercode (die ersten beiden Zeichen der IBAN)

**bank\_code:** die nationale Bankleitzahl bzw. für die Niederlande, wo es keine Bankleitzahlen gibt, ein Teil des BIC.

**bank:** Name der Bank, falls bekannt

**bank\_address:** eventuell über die Bank bekannte Adressdaten.

**bank\_url:** URL der Website der Bank, falls bekannt.

**branch:** Name der Zweigstelle, falls bekannt.

**branch\_code:** Zweigstellenummer, falls bekannt.

**account\_number:** die nationale Kontonummer.

**account\_validation\_method:** Name des nationalen Validierungsalgorithmus der Kontonummer, falls es einen gibt und er bekannt ist.

**account\_validation:** für deutsche und Schweizer Kontonummern, Erläuterungen zur Kontonummervalidierung (in deutscher Sprache)

**length\_check:** 'passed' (Länge passt zum Land) oder 'failed' (IBAN hat für dieses Land die falsche Länge)

**account\_check** (nicht für jedes Land): 'passed' (Kontonummern-Prüfziffer stimmt) oder 'failed' (Fehler bei der Kontonummern-Validierung). Wenn der Algorithmus unbekannt ist oder es keine Prüfziffer gibt, ist das Ergebnis 'passed' oder leer.

**bank\_code\_check** (nicht für jedes Land verfügbar): nationaler Bankcode wurde im Bankleitzahlenverzeichnis gefunden ('passed') bzw. nicht gefunden ('failed')

**iban\_checksum\_check:** 'passed' oder 'failed' (Korrektheit der beiden Ziffern unmittelbar nach dem Ländercode in der IBAN)

**data\_age** (nicht für alle Länder): Alter der BIC-Daten und anderer Bankdaten (nicht definiert für Daten, die im WWW gefunden wurden). Format: yyyyymmdd.

**IBANformat:** eine textuelle Beschreibung des IBAN-Formats für das gegebene Land, zum Beispiel: 'DEkk BBBB BBBB CCCC CCCC CC'.

**formatcomment:** eine Erklärung (auf Englisch) des IBANformat-Strings, zum Beispiel: 'B = sort code (BLZ), C = account No.'

**balance:** die Anzahl noch verfügbarer Berechnungen, bevor Sie Ihr Konto neu aufladen müssen. Diese Zahl ist für Kunden, die unbegrenzt viele Berechnungen gebucht haben, bedeutungslos.

## 2. calculate\_iban

Berechnet für die angegebene nationale Kontonummer die IBAN. Die Benutzer-ID und das Passwort sind dieselben, die auch für das Einloggen als Premium-Benutzer verwendet werden.

*Eingabe:* calculate\_ibanRequest mit den Feldern:

**country:** xsd:string - Land als ISO-Code

**bankcode:** xsd:string - nationaler Bankcode (Bankleitzahl), BC-Nummer für die Schweiz. Für folgende Länder sollte die komplette nationale Kontonummer inklusive Bank- und Zweigstellenummer als ganzes im Feld account übergeben werden, und das Feld bankcode sollte leer gelassen werden: IT, BE, ES, EE, SK, CZ, HU, LU, PL, FR, GR

**account:** xsd:string - nationale Kontonummer

**user:** xsd:string - Ihre Premium-Abonnement-Benutzer-ID.

**password:** xsd:string - Ihr Passwort.

**bic:** xsd:string - relevant nur für NL/GB: der BIC (oder zumindest die ersten 4 Stellen).

Für GB können BICs aus dem Web-Crawl genutzt werden, aber diese Liste ist unvollständig. Falls für eine britische Kontonummer kein BIC angegeben wird und für den angegebenen Sort Code auch kein BIC im Web-Crawl gefunden wird, wird das in einer Fehlermeldung angegeben. Für NL ist die BIC-Angabe nötig, damit die Bank eindeutig spezifiziert ist.

**legacy\_mode:** xsd:string - nur für CH: mit (Wert: 1) oder ohne (Wert: 0) Validierung arbeiten (schnell, aber unzuverlässig, oder deutlich langsamer, aber zuverlässig). Das langsame Programm stammt von Swiss Interbank Clearing und besteht auf korrekter Kontonummern-Syntax (inklusive Punkte, Bindestriche usw.).

*Ausgabe:* tns:IBANCalcResStruct mit den Feldern:

**result:** 'passed' oder 'failed' – für eine korrekte oder inkorrekte Kombination aus Kontonummer, Bankcode und Land

**iban:** die berechnete IBAN (falls die nationale Kontonummer und Bankleitzahl eine akzeptable Länge hatten; eine IBAN wird unter Umständen selbst dann berechnet, wenn die Kontonummer nicht richtig validiert werden konnte..)

**checks:** ein Array mit den durchgeführten Tests; kann die Elemente 'length' (für eine zum Land

passende Länge der IBAN), 'bank\_code' (fürs Nachschlagen der Bankleitzahl) und 'account\_number' (für einen Prüfziffern-Check der Kontonummer) enthalten.

**bic\_candidates:** ein Array mit null, einem oder mehreren Elementen vom Typ BICstruct. Jedes BICstruct-Element hat die Attribute bic, wwwcount, sampleurl und city. Das Attribut bic enthält einen BIC (Bank Identifier Code), der eventuell zu der angegebenen Bankleitzahl gehört. Ob das wirklich der Fall ist, kann man anhand der anderen Felder beurteilen: wenn wwwcount größer ist als Null, wurde dieser BIC im WWW gefunden, und zwar auf so vielen verschiedenen Seiten, wie durch wwwcount angegeben ist. Eine Beispiel-URL für eine Seite, wo der BIC gefunden wurde, steht im Feld sampleurl. Wenn wwwcount dagegen Null ist, stammt der BIC aus einer Liste der Nationalbank oder der EZB. Wenn eine Stadt angegeben ist, bedeutet das ebenfalls, dass der BIC aus einer Liste der Nationalbank oder EZB stammt. Die angegebene Stadt ist nicht unbedingt die, wo sich die zum Konto gehörende Zweigstelle befindet. Es kann auch die Stadt sein, wo die Bank ihre Zentrale hat.

**country:** der ISO-Ländercode (die ersten beiden Zeichen der IBAN)

**bank\_code:** die nationale Bankleitzahl bzw. für die Niederlande, wo es keine Bankleitzahlen gibt, ein Teil des BIC. BC-Nummer für die Schweiz. Es werden Funktionen für Listen von Schweizer und niederländischen Bankcodes bereitgestellt (siehe unten).

**alternative\_bank\_code:** einige wenige Banken betten nicht dieselbe Bankleitzahl in die IBAN ein, die sie ihren Kunden mitteilen. Wenn dieses Feld mit einer BLZ gefüllt ist, ist das möglicherweise der Fall, und die berechnete IBAN ist möglicherweise falsch.

**bank:** Name der Bank, falls bekannt

**bank\_address:** eventuell über die Bank bekannte Adressdaten.

**bank\_url:** URL der Website der Bank, falls bekannt.

**branch:** Name der Zweigstelle, falls bekannt.

**branch\_code:** Zweigstellenummer, falls bekannt.

**account\_number:** die nationale Kontonummer.

**alternative\_account\_number:** für deutsche Konten muss manchmal eine Unterkontonummer angehängt werden, um die korrekte Kontonummer zu erhalten, die in die IBAN eingebettet werden muss. Falls das der Fall ist, erscheint die erweiterte Kontonummer in diesem Feld.

**account\_validation\_method:** Name des nationalen Validierungsalgorithmus der Kontonummer, falls es einen gibt und er bekannt ist.

**account\_validation:** für deutsche und Schweizer Kontonummern, Erläuterungen zur Kontonummervalidierung (in deutscher Sprache)

**length\_check:** 'passed' (Längen sind plausibel), 'failed for bank code' (BLZ-Länge scheint falsch zu sein), oder 'failed for account number' (Kontonummer hat eine unplausible Stellenzahl)

**account\_check** (nicht für jedes Land): 'passed' (Kontonummern-Prüfziffer stimmt) oder 'failed' (Fehler bei der Kontonummern-Validierung). Wenn der Algorithmus unbekannt ist oder es keine Prüfziffer gibt, ist das Ergebnis 'passed' oder leer. Für Deutschland ist auch der Wert 'passed after correction' möglich. In diesem Fall, siehe alternative\_account\_number.

**bank\_code\_check** (nicht für jedes Land verfügbar): nationaler Bankcode wurde im Bankleitzahlenverzeichnis gefunden ('passed') bzw. nicht gefunden ('failed')

**bic\_plausibility\_check:** nur für die Niederlande. Prüft, wie oft niederländische Kontonummern mit dem angegebenen BIC wirklich mit denselben Ziffern beginnen wie die angegebene Kontonummer. Da das nichts beweist, sondern nur eine Schätzung der Plausibilität erlaubt, wird dieser Test nicht für die Entscheidung, ob die Bankverbindung insgesamt stimmt, herangezogen.

**data\_age** (nicht für alle Länder): Alter der BIC-Daten und anderer Bankdaten (nicht definiert für Daten, die im WWW gefunden wurden). Format: yyyyymmdd.

**IBANformat:** eine textuelle Beschreibung des IBAN-Formats für das gegebene Land, zum Beispiel: 'DEkk BBBB BBBB CCCC CCCC CC'.

**formatcomment:** eine Erklärung (auf Englisch) des IBANformat-Strings, zum Beispiel: 'B = sort code

(BLZ), C = account No.'

**balance:** die Anzahl noch verfügbarer Berechnungen, bevor Sie Ihr Konto neu aufladen müssen. Diese Zahl ist für Kunden, die unbegrenzt viele Berechnungen gebucht haben, bedeutungslos.

### 3. get\_dutch\_banks

Liefert eine Liste niederländischer Banken und ihrer BICs.

Niederländische Kontonummern enthalten keine zuverlässigen Informationen über die Bank. Sie beginnen zwar mit einer Art Bankleitzahl, aber da die Bankkunden ihre Kontonummer zu anderen Banken mitnehmen können, kommt es durchaus vor, dass diese Bankleitzahl nicht zur kontoführenden Bank gehört. Daher muss für das Berechnen eines niederländischen Kontos nicht nur die Kontonummer, sondern auch der BIC an die Funktion calculate\_iban übergeben werden; der BIC kann aus einer Liste gewählt werden, die von der Funktion get\_dutch\_banks geliefert wird.

*Eingabe:* get\_dutch\_banksRequest mit den Feldern:

**user:** xsd:string - Ihre Premium-Abonnement-Benutzer-ID.  
**password:** xsd:string - Ihr Passwort.

*Ausgabe:* tns:BankResStruct mit den Feldern:

**result:** " (leer), wenn alles in Ordnung ist. Andernfalls: eine Fehlermeldung.

**banks:** ein Array von Bank-structs mit den Feldern:

\* name: Bankname.

\* bic: der BIC.

\* bankcode: leer für niederländische Banken.

**balance:** die Anzahl noch verfügbarer Berechnungen, bevor Sie Ihr Konto neu aufladen müssen. Diese Zahl ist für Kunden, die unbegrenzt viele Berechnungen gebucht haben, bedeutungslos.

### 4. get\_swiss\_banks

Wegen der hohen Anzahl von BC-Nummern muss diese Funktion in zwei Schritten benutzt werden:

1. Wenn der Bankname leer gelassen wird, liefert die Funktion get\_swiss\_banks eine Liste aller Namen von Schweizer Banken, aber noch ohne BC-Nummern.
2. Wenn einer der Namen aus der Liste aller Namen von Schweizer Banken übergeben wird, liefert die Funktion alle BC-Nummern für die angegebene Bank.

*Eingabe:* get\_swiss\_banksRequest mit den Feldern:

**bank:** xsd:string - Bankname; wenn dieses Feld leer gelassen wird, liefert die Funktion eine Liste aller Banknamen, andernfalls eine Liste aller BC-Nummern für die angegebene Bank.

**user:** xsd:string - Ihre Premium-Abonnement-Benutzer-ID.

**password:** xsd:string - Ihr Passwort.

*Ausgabe:* tns:BankResStruct mit den Feldern:

**result:** " (leer), wenn alles in Ordnung ist. Andernfalls: eine Fehlermeldung.

**banks:** ein Array von Bank-structs mit den Feldern:

\* name: Bankname.

\* bic: der BIC, falls die Funktion mit einem Banknamen als Parameter aufgerufen wurde. Andernfalls leer.

\* bankcode: wenn die Funktion mit einem Banknamen als erstem Parameter aufgerufen wurde, enthält dieses Feld die BC-Nummer. Andernfalls ist es leer.

**balance:** die Anzahl noch verfügbarer Berechnungen, bevor Sie Ihr Konto neu aufladen müssen. Diese Zahl ist für Kunden, die unbegrenzt viele Berechnungen gebucht haben, bedeutungslos.

## WSDL

```
<definitions targetNamespace="https://ssl.ibanrechner.de/">
  <types>
    <xsd:schema targetNamespace="https://ssl.ibanrechner.de/">
<xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
<xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />

    <xsd:complexType name="ArrayOfString">
      <xsd:complexContent>
        <xsd:restriction base="SOAP-ENC:Array">
<xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="string[]" />
        </xsd:restriction>
      </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="BICStruct">
      <xsd:all>
<xsd:element name="bic" type="xsd:string" />
<xsd:element name="city" type="xsd:string" />
<xsd:element name="wwwcount" type="xsd:int" />
<xsd:element name="sampleurl" type="xsd:string" />
      </xsd:all>
    </xsd:complexType>

    <xsd:complexType name="ArrayOfBICStruct">
      <xsd:complexContent>
        <xsd:restriction base="SOAP-ENC:Array">
<xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="BICStruct[]" />
        </xsd:restriction>
      </xsd:complexContent>
    </xsd:complexType>

    <xsd:complexType name="BankStruct">
      <xsd:all>
<xsd:element name="name" type="xsd:string" />
<xsd:element name="bic" type="xsd:string" />
<xsd:element name="bankcode" type="xsd:int" />
      </xsd:all>
    </xsd:complexType>

    <xsd:complexType name="ArrayOfBankStruct">
      <xsd:complexContent>
        <xsd:restriction base="SOAP-ENC:Array">
<xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="BankStruct[]" />
        </xsd:restriction>
      </xsd:complexContent>
    </xsd:complexType>
  </types>
</definitions>
```

```
</xsd:complexContent>
</xsd:complexType>
```

```
    <xsd:complexType name="BankResStruct">
      <xsd:all>
        <xsd:element name="result" type="xsd:string"/>
        <xsd:element name="banks" type="tns:ArrayOfBankStruct"/>
        <xsd:element name="balance" type="xsd:int"/>
      </xsd:all>
    </xsd:complexType>
```

```
    <xsd:complexType name="IBANValResStruct">
      <xsd:all>
        <xsd:element name="iban" type="xsd:string"/>
        <xsd:element name="result" type="xsd:string"/>
        <xsd:element name="checks" type="tns:ArrayOfString"/>
        <xsd:element name="bic_candidates" type="tns:ArrayOfBICStruct"/>
        <xsd:element name="country" type="xsd:string"/>
        <xsd:element name="bank_code" type="xsd:string"/>
        <xsd:element name="bank" type="xsd:string"/>
        <xsd:element name="bank_address" type="xsd:string"/>
        <xsd:element name="bank_url" type="xsd:string"/>
        <xsd:element name="branch" type="xsd:string"/>
        <xsd:element name="branch_code" type="xsd:string"/>
        <xsd:element name="account_number" type="xsd:string"/>
        <xsd:element name="account_validation_method" type="xsd:string"/>
        <xsd:element name="account_validation" type="xsd:string"/>
        <xsd:element name="length_check" type="xsd:string"/>
        <xsd:element name="account_check" type="xsd:string"/>
        <xsd:element name="bank_code_check" type="xsd:string"/>
        <xsd:element name="iban_checksum_check" type="xsd:string"/>
        <xsd:element name="data_age" type="xsd:string"/>
        <xsd:element name="IBANformat" type="xsd:string"/>
        <xsd:element name="formatcomment" type="xsd:string"/>
        <xsd:element name="balance" type="xsd:int"/>
      </xsd:all>
    </xsd:complexType>
```

```
    <xsd:complexType name="IBANCalcResStruct">
      <xsd:all>
        <xsd:element name="iban" type="xsd:string"/>
        <xsd:element name="result" type="xsd:string"/>
        <xsd:element name="checks" type="tns:ArrayOfString"/>
        <xsd:element name="bic_candidates" type="tns:ArrayOfBICStruct"/>
        <xsd:element name="country" type="xsd:string"/>
        <xsd:element name="bank_code" type="xsd:string"/>
        <xsd:element name="bank" type="xsd:string"/>
        <xsd:element name="bank_address" type="xsd:string"/>
        <xsd:element name="bank_url" type="xsd:string"/>
```

```
<xsd:element name="branch" type="xsd:string"/>
<xsd:element name="branch_code" type="xsd:string"/>
<xsd:element name="account_number" type="xsd:string"/>
<xsd:element name="alternative_account_number" type="xsd:string"/>
<xsd:element name="account_validation_method" type="xsd:string"/>
<xsd:element name="account_validation" type="xsd:string"/>
<xsd:element name="length_check" type="xsd:string"/>
<xsd:element name="account_check" type="xsd:string"/>
<xsd:element name="bank_code_check" type="xsd:string"/>
<xsd:element name="bic_plausibility_check" type="xsd:string"/>
<xsd:element name="data_age" type="xsd:string"/>
<xsd:element name="IBANformat" type="xsd:string"/>
<xsd:element name="formatcomment" type="xsd:string"/>
<xsd:element name="balance" type="xsd:int"/>
</xsd:all>
</xsd:complexType>
</xsd:schema>
</types>
```

```
    <message name="validate_ibanRequest">
<part name="iban" type="xsd:string"/>
<part name="user" type="xsd:string"/>
<part name="password" type="xsd:string"/>
</message>
```

```
    <message name="validate_ibanResponse">
<part name="return" type="tns:IBANValResStruct"/>
</message>
```

```
    <message name="calculate_ibanRequest">
<part name="country" type="xsd:string"/>
<part name="bankcode" type="xsd:string"/>
<part name="account" type="xsd:string"/>
<part name="user" type="xsd:string"/>
<part name="password" type="xsd:string"/>
<part name="bic" type="xsd:string"/>
<part name="legacy_mode" type="xsd:int"/>
</message>
```

```
    <message name="calculate_ibanResponse">
<part name="return" type="tns:IBANCalcResStruct"/>
</message>
```

```
    <message name="get_dutch_banksRequest">
<part name="user" type="xsd:string"/>
<part name="password" type="xsd:string"/>
</message>
```

```
    <message name="get_dutch_banksResponse">
```

```
<part name="return" type="tns:BankResStruct"/>
</message>
```

```
    <message name="get_swiss_banksRequest">
<part name="bank" type="xsd:string"/>
<part name="user" type="xsd:string"/>
<part name="password" type="xsd:string"/>
</message>
```

```
    <message name="get_swiss_banksResponse">
<part name="return" type="tns:BankResStruct"/>
</message>
```

```
    <portType name="IBAN CalculatorPortType">
```

```
      <operation name="validate_iban">
        <documentation>
```

Validates the given IBAN. The user ID and password you need to pass with the request are the same you use for logging in as a premium user. This service is available to all premium users who have paid their subscription fee. For more details on the subscription model, see [www.iban-bic.com](http://www.iban-bic.com).<br>

Output fields are:<br>

iban: the IBAN that was validated.<br>

result: 'passed' or 'failed' - for a valid or invalid IBAN<br>

checks: an array of the checks performed (can contain elements such as 'length', 'bank\_code', 'account\_number', 'iban\_checksum').<br>

bic\_candidates: an array of BICs that are associated with the given national bank code. May be empty or may contain one or more elements. The BICstruct element has the attributes bic, wwwcount, sampleurl, and city. If a wwwcount greater than zero is given, the BIC was harvested from the Web (and found on as many pages as indicated by wwwcount, for example at the URL given by sampleurl). If wwwcount is zero, the BIC comes from a national bank or the ECB. If city is given, this also indicates that the BIC comes from a national bank or the ECB. The given city does not necessarily reflect the location of the given branch - it can also be the location of the headquarters.<br>

country: the ISO country code (first two letters of the IBAN)<br>

bank\_code: the national bank code. Part of the BIC for NL (where no national bank code exists).<br>

bank: bank name, if known<br>

bank\_address: some address data, if known<br>

bank\_url: URL of website, if known<br>

branch: branch name, if known<br>

branch\_code: branch code, if known<br>

account\_number: the national bank account number<br>

account\_validation\_method: name of the validation algorithm for the national account number<br>

account\_validation: for German or Swiss account numbers, an explanation (in German)<br>

length\_check: 'passed' or 'failed' - for the right number of characters for the given country<br>

account\_check (not provided for every country): 'passed' or 'failed' (checksum validation; if the algorithm is unknown, or if there is no checksum, the result is 'passed' or empty).<br>

bank\_code\_check (not provided for every country): lookup of national bank code was successful ('passed') or not ('failed')<br>

iban\_checksum\_check: 'passed' or 'failed' (correctness of the two digits right after the country code in the IBAN)<br>



data\_age (not given for all countries): age of the BIC and other bank-related data (not defined for data harvested from the Web). Format: yyyyymmdd.<br>

IBANformat: a string describing the IBAN format for the given country, for example: 'DEkk BBBB BBBB CCCC CCCC CC'.<br>

formatcomment: an explanation of the IBANformat string, for example: 'B = sort code (BLZ), C = account No.'.<br>

balance: the number of remaining calculations you can do before having to recharge your account. This does not apply to customers with a subscription which includes an unlimited number of calculations.

</documentation>

<input message="tns:validate\_ibanRequest"/>

<output message="tns:validate\_ibanResponse"/>

</operation>

<operation name="calculate\_iban">

<documentation>

Calculates an IBAN for the given national account number. The user ID and password you need to pass with the request are the same you use for logging in as a premium user. This service is available to all premium users who have paid their subscription fee. For more details on the subscription model, see [www.iban-bic.com](http://www.iban-bic.com).<br>

Parameters are:<br>

country - the 2-letter ISO country code.<br>

bankcode - The national bank code (BC number for Switzerland). For the following countries, the whole national account number, including any bank codes and branch codes, should be passed with the account parameter, and the bankcode parameter should stay empty: IT, BE, ES, EE, SK, CZ, HU, LU, PL, FR, GR<br>

account - the national account number<br>

user - your unique user ID<br>

password - your password<br>

bic - the BIC (or at least a sufficiently long prefix) for GB, NL (GB: web crawl can be used, but does not provide all BICs. If the bic parameter is omitted, and the BIC is not found in the Web crawl, this will be shown with an error message in the output field result). This parameter is ignored for countries other than GB and NL.<br>

legacy\_mode: If set to 1 for Switzerland, the calculator does not use the external Java program provided by Swiss Interbank Clearing. This parameter has no effect for countries other than Switzerland. For Switzerland, the Java program validates account numbers, but is slower and requires all characters to be entered correctly (such as dots, hyphens etc.). The legacy mode is faster, but does not validate account numbers, and is less likely to produce correct results.<br><br>

Output fields are:<br>

result: 'passed' or 'failed' - for a valid or invalid account number/bank code/country combination<br>

iban: the calculated IBAN (if the national account number/bank code had the right length; an IBAN may be provided even if the account number validation fails.)<br>

checks: an array of the checks performed (can contain elements such as 'length', 'bank\_code', 'account\_number').<br>

bic\_candidates: an array of BICs that are associated with the given national bank code. May be empty or may contain one or more elements. The bic element has the attributes bic, wwwcount, sampleurl, and city. If a wwwcount greater than zero is given, the BIC was harvested from the Web (and found on as many pages as indicated by wwwcount, for example on the page indicated by sampleurl). If wwwcount is zero, the BIC comes from a national bank or the ECB. If city is given, this also indicates

that the BIC comes from a national bank or the ECB. The given city does not necessarily reflect the location of the given branch - it can also be the location of the headquarters.<br>  
country: the ISO country code (first two letters of the IBAN)<br>  
bank\_code: the national bank code, if it exists. BIC for NL (where no national bank code exists), BC-Nummer for Switzerland. There are functions provided for obtaining Dutch BICs and Swiss BC numbers.<br>  
bank: bank name, if known<br>  
bank\_address: some address data, if known<br>  
bank\_url: URL of website, if known<br>  
branch: branch name, if known<br>  
branch\_code: branch code, if known<br>  
account\_number: the national bank account number<br>  
alternative\_account\_number: sometimes, a subaccount number needs to be appended to a German account number before it is embedded in an IBAN. If this is the case, the extended account number appears in this field.<br>  
account\_validation\_method: name of the validation algorithm for the national account number<br>  
account\_validation: for German or Swiss account numbers, an explanation (in German)<br>  
length\_check: 'passed', 'failed for bank code', or 'failed for account number' - for the right number of characters for the given country and account number/bank code<br>  
account\_check (not provided for every country): 'passed' or 'failed'; for Germany also possible: passed after correction - in that case see alternative\_account\_number (checksum validation; if the algorithm is unknown, or if there is no checksum, the result is 'passed' or empty).<br>  
bank\_code\_check (not provided for every country): lookup of national bank code was successful ('passed') or not ('failed')<br>  
bic\_plausibility\_check: NL only. Checks how often account numbers of the given bank actually have the given prefix. Since this does not guarantee anything, this check is ignored for the overall pass/fail decision that is reported in the result field. But for some Dutch banks, it can give you an idea of how likely it is that the right BIC was provided.<br>  
data\_age (not given for all countries): age of the BIC and other bank-related data (not defined for data harvested from the Web). Format: yyyyymmdd.<br>  
IBANformat: a string describing the IBAN format for the given country, for example: 'DEkk BBBB BBBB CCCC CCCC CC'.<br>  
formatcomment: an explanation of the IBANformat string, for example: 'B = sort code (BLZ), C = account No.'<br>  
balance: the number of remaining calculations you can do before having to recharge your account. This does not apply to customers with a subscription which includes an unlimited number of calculations.

</documentation>  
<input message="tns:calculate\_ibanRequest"/>  
<output message="tns:calculate\_ibanResponse"/>  
</operation>

<operation name="get\_dutch\_banks">  
<documentation>

Returns a list of Dutch banks and their BICs. <br>  
Dutch national account numbers do not contain reliable information about the bank. There is a bank code (of variable length) embedded in Dutch account numbers, but it is possible for bank clients to keep an account number when switching banks, and therefore there exist accounts where the bank code does not match the bank. When calculating a Dutch IBAN, you need to pass an account number and a BIC to the function calculate\_iban; the BIC can be picked from the list which is returned by the

```
function get_dutch_banks.<br>
Output fields:<br>
result: empty if everything is okay, otherwise an error message.<br>
banks: an array of bank structs with the fields:<br>
* name: the name of the bank.<br>
* bic: the BIC.<br>
* bankcode: this field is left empty for Dutch banks.<br>
balance: the number of remaining calculations you can do before having to recharge your account. This
does not apply to customers with a subscription which includes an unlimited number of calculations.
</documentation>
<input message="tns:get_dutch_banksRequest"/>
<output message="tns:get_dutch_banksResponse"/>
</operation>
```

```
    <operation name="get_swiss_banks">
    <documentation>
```

Returns a list of Swiss banks and their BC numbers. <br>

Due to the large number of BC numbers, this function needs to be used in two steps:<br>

1. If called with an empty bank name, get\_swiss\_banks returns a list of all names of Swiss banks, but still without BC numbers.<br>
2. If called with one of the bank names that were obtained in step 1, the function returns all BC numbers for the given bank.<br>

Output fields:<br>

result: empty if everything is okay, otherwise an error message.<br>

banks: an array of bank structs with the fields:<br>

\* name: the name of the bank.<br>

\* bic: if the function was called with a non-empty bank name, the BIC. Otherwise, this field is left empty.<br>

\* bankcode: if the function was called with a non-empty bank name as first parameter, the bankcode field contains a BC number. Otherwise, it is empty.<br>

balance: the number of remaining calculations you can do before having to recharge your account. This does not apply to customers with a subscription which includes an unlimited number of calculations.

</documentation>

```
<input message="tns:get_swiss_banksRequest"/>
```

```
<output message="tns:get_swiss_banksResponse"/>
```

```
</operation>
```

```
</portType>
```

```
    <binding name="IBAN CalculatorBinding" type="tns:IBAN CalculatorPortType">
```

```
<soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
```

```
    <operation name="validate_iban">
```

```
<soap:operation soapAction="https://ssl.ibanrechner.de/#validate_iban" style="rpc"/>
```

```
    <input>
```

```
<soap:body use="encoded" namespace="https://ssl.ibanrechner.de/"
```

```
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
```

```
</input>
```

```
<output>
<soap:body use="encoded" namespace="https://ssl.ibanrechner.de/"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
</output>
</operation>
```

```
<operation name="calculate_iban">
<soap:operation soapAction="https://ssl.ibanrechner.de/#validate_iban" style="rpc"/>
```

```
<input>
<soap:body use="encoded" namespace="https://ssl.ibanrechner.de/"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
</input>
```

```
<output>
<soap:body use="encoded" namespace="https://ssl.ibanrechner.de/"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
</output>
</operation>
```

```
<operation name="get_dutch_banks">
<soap:operation soapAction="https://ssl.ibanrechner.de/#get_dutch_banks" style="rpc"/>
```

```
<input>
<soap:body use="encoded" namespace="https://ssl.ibanrechner.de/"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
</input>
```

```
<output>
<soap:body use="encoded" namespace="https://ssl.ibanrechner.de/"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
</output>
</operation>
```

```
<operation name="get_swiss_banks">
<soap:operation soapAction="https://ssl.ibanrechner.de/#get_dutch_banks" style="rpc"/>
```

```
<input>
<soap:body use="encoded" namespace="https://ssl.ibanrechner.de/"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
</input>
```

```
<output>
<soap:body use="encoded" namespace="https://ssl.ibanrechner.de/"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/>
</output>
</operation>
</binding>
```

```
<service name="IBAN Calculator">  
  <port name="IBAN CalculatorPort" binding="tns:IBAN CalculatorBinding">  
<soap:address location="http://www.iban-rechner.de/soap/index.php"/>  
</port>  
</service>  
</definitions>
```