

SendanSOS.com – API v0.81 (May 2011)

SendanSOS.com has developed and published a RESTlike API to enable software developers to develop their own client applications using the services provided by SendanSOS.com.

Such a client might be a website, an iOS app, an Android app, or a Windows Phone 7 app.

The API does not make full use of the http semantics. All requests, whether fetching or updating, must be POST requests.

All request and response data are in JSON format.

The url is:

`http://SendanSOS.com/api.asmx/action`

The method is always:

POST

The Request Header must contain the name/value pair:

Content-Type: application/json

All actions, except Login, ForgotPassword and AddUser require the userToken parameter. The userToken is returned by the Login action.

If an application error occurs, the http response status code will be 500. The text of the error will appear in the http response status description **and** the http response body.

All Date/Time data is in string format: yyyy-M-d-H and must be UTC based. A separate Timezone field is used adjust for local time. [eg. 2011-5-23-7 and Timezone:-12]

Please note: We have developed client-side bindings for Java, C# and jQuery - to wrap the details of dealing with Http and JSON.

List of Actions:

Login

Request Body:

```
{email: 'bob@abc.com', password: 'xxxxxx'}
```

Response Body:

```
{'d': 'xxxx-aaaa-yyyyy'}
```

Description:

If the email and password combination are valid, a user token will be returned. The user token is needed for all subsequent requests for that user session.

ForgotPassword

Request Body:

```
{email: 'bob@abc.com'}
```

Response Body:

```
{"d": "Success - Your password has been sent to your email address."}
```

Description:

If the email is for an existing user, the system will send the user's password to the supplied email address.

AddUser

Request Body:

```
{  
  name: 'name',  
  email: 'emailaddress',  
  mobile: 'mobilenumber', // country space network space number eg. 64 27 111111  
  password: 'password'  
}
```

Response Body:

```
{"d": "usertoken"}
```

Description:

The password must be at least 6 characters long.

The email address must not be already in use.

The mobile number is optional. If it is not used, it must be set to null (no quotes). If the mobile number (if present) must be a valid mobile number. It must contain the country, the network and the local number with one space between them and no 0 prefixes.

On success, the system will send a welcome message to the supplied email address.

The user token is returned representing the logged on user.

UpdateUser**Request Body:**

```
{
  userToken:'token',
  name:'name',
  email:'emailaddress',
  mobile:'mobilenumber',
  password:'password'
}
```

Response Body:

```
{"d": "Success - Details updated"}
```

Description:

The password is optional. If it is not to be updated, it must be set to null.

The mobile number is optional. If it is not used, it must be set to null (no quotes). If the mobile number (if present) must be a valid mobile number. It must contain the country, the network and the local number with one space between them and no 0 prefixes.

GetUser**Request Body:**

```
{userToken: 'myUserToken'}
```

Response Body:

```
{"d":
  {
    "__type": "PubPerson",
    "PersonId": "1111111",
  }
}
```

```
"Name": "Trev",
"Email": "bob@abc.com",
"CellPhone": "6411999999",
"SOSBalance": 99,
"CreateDate": "2010-2-28-7",
"IsMember": true,
"HasCellPhone": true,
"HasTwitter": true
}
}
```

Description:

Returns the details of the logged on user.

InviteAContact**Request Body:**

```
{userToken: 'myusertoken', email: 'contactemail', name: 'contactname' }
```

Response Body:

```
{"d": "Success - Invitation has been sent to contactname"}
```

Description:

The system will send an email to the supplied email address. The email contains a link to the InviteContact.aspx page on the SendanSOS.com website. The page contains a button the person can click to accept the invite from the logged on user to be an emergency contact. If the invite is accepted the system will send an email notification to the logged on user and a welcome email to the new contact.

OfferToBeAContact**Request Body:**

```
{userToken: 'myusertoken', email: 'email', name: 'name' }
```

Response Body:

```
{"d": "Success - Offer has been sent to name"}
```

Description:

The system will send an email to the supplied email address. The email will contain a link to the OfferToBeAContact.aspx page on the SendanSOS.com website. The page contains a button the person can click to accept the offer from the logged

on user to be their emergency contact. If the offer is accepted the system will send an email notification to the logged on user and a welcome email to the new user.

DeleteContact

Request Body:

```
{userToken: 'myusertoken', contactId:'1234567' }
```

Response Body:

```
{"d": "Success - Contact deleted"}
```

Description:

The system will delete the specified contact for the logged on user.

The supplied contactId must be a contact for the logged on user.

Any pending notification will not be sent to the contact.

GetContacts

Request Body:

```
{userToken: 'myusertoken' }
```

Response Body:

```
{  
  "d": [  
    {  
      "__type": "PubPerson",  
      "PersonId": "111111",  
      "Name": "Zac",  
      "Email": "zac@abc.com",  
      "CellPhone": " ",  
      "SOSBalance": 3,  
      "CreatedDate": "2010-2-28-7",  
      "IsMember": true,  
      "HasCellPhone": false,  
      "HasTwitter": true  
    },  
    {  
      "__type": "PubPerson",  
      "PersonId": "222222",  
      "Name": "Fred",  
      "Email": "fred@abc.com",  
      "CellPhone": " ",  
      "SOSBalance": 50,  
      "CreatedDate": "2011-4-27-3",  
    }  
  ]  
}
```

```
        "IsMember":false,
        "HasCellPhone":false,
        "HasTwitter":false
    }
    ]...
}
```

Description:

Returns all the contacts for the logged on user.

GetPersonsImAContactFor

Request Body:

```
{userToken: 'myusertoken' }
```

Response Body:

See GetContacts

Description:

Returns all the users for whom the logged on user is a contact.

AddPlan

Request Body:

```
{
  userToken: 'myusertoken',
  plan:
  {
    Itinerary: 'Climbing the Eiger via the south face',
    Route: '172.1, 45.0: 172.9, 45.6: ...', //lng1,lat1: lng2, lat2: ....
    Timezone: '-12',
    DepartureDate: '2011-2-18-8',
    ReturnDate: '2011-2-22-17',
    ReminderDate: '2011-2-22-13',
    ReminderTwitterRequired: true,
    ReminderSMSRequired: true,
    ReminderEmailRequired: false,
    Notifications :
    [
      {
        ContactId : '1234567',
```

```
        TwitterNotify: true,  
        EmailNotify: true,  
    },  
    {  
        ContactId : '1234568',  
        CellPhoneNotify: true,  
        EmailNotify: true,  
    }  
]  
}  
}
```

Response Body:

```
{"d": "Success - Plan created"}
```

Description:

A plan is created for the logged on user, using the details supplied in PubPlan.

Please take care to ensure you provide valid data in the plan object.

(Note: The route data is in the form lng1,lat1: lng2,lat2, lng3,lat3....)

UpdatePlan

Request Body:

See AddPlan

Response Body:

```
{"d": "Success - Plan updated"}
```

Description:

A plan is updated for the logged on user, using the details supplied in PubPlan.

Please take care to ensure you provide valid data in the plan object.

TerminatePlan

Request Body:

```
{userToken: 'xxxxxx', planId: '111111'}
```

Response Body:

```
{"d":"Success - Plan terminated"}
```

Description:

This prevents the logged on user's plan from being sent to the nominated contacts.

GetActivePlans

Request Body:

```
{ userToken: 'myusertoken' }
```

Response Body:

```
{"d":  
[  
  {  
    "__type": "PubPlan",  
    "Status": "L",  
    "PlanId": "111111",  
    "PersonId": "222222",  
    "Itinerary": "From my Android",  
    "Route": "168.0,44.0:169.45.0",  
    "Timezone": -11,  
    "DepartureDate": "2011-5-4-6",  
    "ReturnDate": "2011-5-6-17",  
    "ReminderDate": "2011-5-6-16",  
    "ReminderTwitterRequired": false,  
    "ReminderSMSRequired": false,  
    "ReminderEmailRequired": true,  
    "Notifications":  
    [  
      {  
        "PlanId": "111111",  
        "TwitterNotify": false,  
        "CellPhoneNotify": false,  
        "EmailNotify": true,  
        "Contact":  
        {  
          "__type": "PubPerson",  
          "PersonId": "222222",  
          "Name": "Zac",  
          "Email": "zac@netrix.co.nz",  
          "CellPhone": "",  
          "SOSBalance": 3,  
          "CreatedDate": "2010-2-28-7",  
          "IsMember": true,  
          "HasCellPhone": false,  
          "HasTwitter": true  
        }  
      }  
    ]  
  }  
]
```

```
    }  
  ]  
}
```

Description

Returns the list of active plans (and their notifications) for the logged on user.

GetReminderPlans:

Request Body:

```
{ userToken: 'myusertoken' }
```

Response Body:

See GetActivePlans

Description

Returns the list of plans (and their notifications) for the logged on user that have passed their reminder date, but are not yet overdue.

GetOverduePlans:

Request Body:

```
{ userToken: 'myusertoken' }
```

Response Body:

See GetActivePlans

Description

Returns the list of plans (and notifications) for the logged on user that have passed their overdue date.

GiftCredits:

Request Body:

```
{  
  userToken: 'myusertoken',  
  recipientId: '11111',  
  credits: 10,  
  tellRecipient: true  
}
```

Response Body:

```
{"d": "Success - credits SOS Credits have been gifted to recipientName"}
```

Description

Transfer SOS credits from the logged on user to the recipient. Optionally an email is sent to the recipient telling them who gave the credits.

Classes

class PubPlan

```
{  
    string PlanId           // unique key of the plan  
    string PersonId        // key of the person making the plan  
    string Itinerary       // where the person is going  
    int Timezone           // timezone of client  
    string DepartureDate   // GMT - format yyyy-M-d-H  
    string ReturnDate      // GMT - format yyyy-M-d-H  
    string ReminderDate    // GMT - format yyyy-M-d-H  
    string Status          // active/deleted  
    bool ReminderTwitterRequired // true or false  
    bool ReminderSMSRequired  
    bool ReminderEmailRequired  
    string Route           // lng1,lat1:lng2,lat2:...  
    PubNotification[] Notifications  
}
```

class PubNotification

```
{  
    string PlanId           // key of the plan associated with this notification  
    bool TwitterNotify     // use twitter to notify  
    bool CellPhoneNotify   // use sms to notify  
    bool EmailNotify       // use email to notify  
    PubPerson Contact      // the contact who will be notified  
}
```

class PubPerson

```
{  
    string PersonId        // the unique person id.  
    string Name            // name of the person  
    string Email           // email of the person  
    string CellPhone       // optional cell phone for the person  
    int SOSBalance         // Only used if the person is a subscriber  
    string CreatedDate     // The created date for this person  
    bool IsMember  
    bool HasCellPhone  
    bool HasTwitter  
}
```