Entera Simple REST API v2

1. Description

Using Entera REST API you can easily integrate functional responsible for sending of FAX, SMS and IVR into your software or solution.

Key features of our messaging platform:

- Alphanumeric sender, which can be customized for your needs. Recipients will see it as message sender.
- Multiple sender values. You can define several sender aliases for your company.
- Your unique identifiers for each message, which can be used later to perform deep integration with all variety of your solution components.
- Message tags for easing of identification, in case you are running multiple SMS campaigns.
- Messages can be scheduled to specific date. Great for holiday or birthday congratulations.
- You can define your own message lifetime. This feature can help you to build OTP generation scenario.
- Bulk sending of your messages - hi speed bulk messages sending.
- Merge sending of your messages. You can customize your message using macro parameters to include recipients name or any other variable.
- Extended API with additional functions, which you can use to build your own user interface.
- HTTPS security.

2. Methods description

In Simple REST API we are providing 14 methods:

- **AddressBook** method allows you to create your own address books and later use them to perform bulk or merge sending.
- **AddressBookRecord** method designed to edit elements of your address books.
- **SendFax** method performs sending of single/multiple fax messages.
- **SendFaxBroadcast** method starts fax broadcast.
- **SendMergeFax** method initiates merge delivery process.
- **CancelFax** method cancels delivery of pending Fax delivery.
- **OutboundFax** method returns statistics for your fax messages, including status, delivery time etc.
- **SendSms** method performs sending of single/multiple SMS messages.
- **SendSmsBroadcast** method initiates sending of SMS broadcast.
- **OutboundSms** method returns statistics for your SMS messages, including status, delivery time etc.
- **SendIvr** method performs sending of single/multiple IVR messages.
- **SendIvrBroadcast** method starts IVR broadcast.
- **CancelIvr** method cancels performing of pending IVR delivery.
- **OutboundIvr** method returns statistics for your IVR messages, including status, collected information, delivery time etc.

3. Address book methods

3.1. AddressBook

3.1.1. Retrieving the data

Request URI: http://rest.entera.net/api/v2/AddressBook

Request Type: GET

Required HTTP headers: Login, Password

Description: If you call this method without parameters, it will return all of your address books.

Example of response:

```
{"Result":true,
"ErrorCode":0,
"Message":"",
"AddressBooks":
[
  {"Id":3,"Description":"Users personal address book","RecordsCount":3,"IsGroup":false,"Records":
    [{"Id":1,"AddressBookId":3,"Phone":"905350000001","Metadata":["Name 1","Position 1","name1@entera.com.tr"]},
     {"Id":2,"AddressBookId":3,"Phone":"905350000002","Metadata":["Name 2","Position 2","name2@entera.com.tr"]},
     {"Id":3,"AddressBookId":3,"Phone":"905350000003","Metadata":["Name 3","Position 3","name3@entera.com.tr"]}]
  },

  {"Id":5,"Description":"Test address book","RecordsCount":2,"IsGroup":true,"Records":null}
]}
```

3.1.2. Retrieving the data

Request URI: http://rest.entera.net/api/v2/AddressBookRecord?AddressBookId=5

Request Type: GET

Required HTTP headers: Login, Password

Description: Method will return address book specified in “AddressBookId” parameter.

Example of response:

```
{"Result": true,
"ErrorCode":0,
"Message": "",
"AddressBooks":
[
  {"Id": 5,"Description": "Test address book","RecordsCount": 2,"IsGroup": true,"Records":
    [{"Id": 6,"AddressBookId": 5,"Phone": "905350000001","Metadata": ["Name 1","Position 1","name1@entera.com.tr"]},
     {"Id": 7,"AddressBookId": 5,"Phone": "905350000002","Metadata": ["Name 2","Position 2","name2@entera.com.tr"]}]
  }
]}
```

3.1.3. Saving the data

Request URI: http://rest.entera.net/api/v2/AddressBook

Request Type: POST

Required HTTP headers: Login, Password

Content type: text/plain

Request body: json object

Description: Method will create new address book and returns its Id.

Example of request body:

```
{"Description":"Test address book",
"Id":0,
"Records":
[
{"AddressBookId":0,
"Id":0,
"Metadata":["Name 1","Position 1","name1@entera.com.tr"],
"Phone":"905350000001"},
{"AddressBookId":0,
"Id":0,
"Metadata":["Name 2","Position 2","name2@entera.com.tr"],
"Phone":"905350000002"}
]}
```

Request parameters size limitations:

Description – 255 symbols max.

Phone: 32 symbols max.

Metadata: 512 symbols max.

Example of response:

{"Result":true,"ErrorCode":0,"Message":"",AddressBookId:5 }

Please note that you can use received “AddressBookId” later for sending bulk and merge messages.

3.1.4. Deleting the data

Request URI: http://rest.entera.net/api/v2/AddressBook?AddressBookId=3

Request Type: DELETE

Required HTTP headers: Login, Password

Description: Method will delete address book specified in “AddressBookId” parameter.

Example of response:

{"Result":true,"ErrorCode":0,"Message":""}

3.2. AddressBookRecord

3.2.1. Saving the data

Request URI: http://rest.entera.net/api/v2/AddressBookRecord

Request Type: POST

Required HTTP headers: Login, Password

Content type: text/plain

Request body: json object

Description: This method will add record to your personal address book.

Example of request body:

```
{"AddressBookId":0,
"Id":0,
"Metadata":["Name 1","Position 1","name1@entera.com.tr"],
"Phone":"905350000001"}
```

Request parameters size limitations:

Phone: 32 symbols max.

Metadata: 512 symbols max.

Example of response:

```
{"Result":true,"ErrorCode":0,"Message":"","RecordId":42}
```

Please note that if you will set “AddressBookId” parameter method will add this record to specified address book.

3.2.2. Updating the data

Request URI: http://rest.entera.net/api/v2/AddressBookRecord

Request Type: POST or PUT

Required HTTP headers: Login, Password

Content type: text/plain

Request body: json object

Description: This method will update specified record.

Example of request body:

```
{"AddressBookId":0,
"Id":42,
"Metadata":["Name 1","Position 1","name1@entera.com.tr"],
"Phone":"905350000002"}
```

Request parameters size limitations:

Phone: 32 symbols max.

Metadata: 512 symbols max.

Example of response:

```
{"Result":true,"ErrorCode":0,"Message":""}
```

Please make sure that you are specifying “Id” in json object if you would like to update existing record.

3.2.3. Deleting the data

Request URI: http://rest.entera.net/api/v2/AddressBookRecord?RecordId=42

Request Type: DELETE

Required HTTP headers: Login, Password

Description: Method will delete address book record specified in “RecordId” parameter.

Example of response:

```
{"Result":true,"ErrorCode":0,"Message":""}
```

4. Sending fax methods

4.1. SendFax

Request URI: http://rest.entera.net/api/v2/SendFax

Request Type: POST

Required HTTP headers: Login, Password

Content type: text/plain

Request body: json object

Description: Method will send fax to specified recipients.

Example of request body:

```
{"Alias":"Your message tag",

"Attachments":[{"Name":"1.txt","Base64Content":"RmF4IHRlc3QgMQ=="},

                {"Name":"2.txt","Base64Content":"RmF4IHRlc3QgMg=="}],

"ExternalId":"Your unique id",

"Recipients":["905350000001","905350000002"],

"ScheduleDate":"31\/12\/2013 15:00:00"}
```

Request parameters size limitations:

Alias: 255 symbols max.

ExternalId: 42 symbols max.

Name: 255 symbols max.

Recipient: 32 symbols max.

Example of response:

```
{"Result":true,"ErrorCode":0,"Message":"","TaskId:"123"}
```

Please note that received “TaskId” can be used later to request status of this message.

4.2. SendFaxBroadcast

Request URI: http://rest.entera.net/api/v2/SendFaxBroadcast

Request Type: POST

Required HTTP headers: Login, Password

Content type: text/plain

Request body: json object

Description: Method will initiate fax broadcast to specified recipients.

Example of request body:

```
{"AddressBookIds":[3,5],

"Alias":"Your message tag",

"Attachments":[{"Name":"1.txt","Base64Content":"RmF4IGJyb2FkY2FzdCB0ZXN0IDE="},

                {"Name":"2.txt","Base64Content":"RmF4IGJyb2FkY2FzdCB0ZXN0IDI="}],

"ExternalId":"Your unique id",

"Recipients":["905350000001","905350000002"],
```

"ScheduleDate":"11\/07\/2013 15:00:00",

"NotificationType":1}

Request parameters description:

AddressBookIds – Array of address book Id's. All recipients from these address books will be included to delivery.

NotificationType: 0 – off, 1 – email notification.

Request parameters size limitations:

Same as for SendFax method.

Example of response:

{"Result":true,"ErrorCode":0,"Message":"","TaskId:"123"}

Please note that in this example we are using 3 and 5 in "AddressBookIds" parameter. These numbers are "AddressBookId" values received from "AddressBook" method. Message will be sent to all recipients from these address books.

Also, please note that except address books in this example we are using 2 raw numbers, they will be added to delivery as well.

4.3. SendMergeFax

Request URI: http://rest.entera.net/api/v2/SendMergeFax

Request Type: POST

Required HTTP headers: Login, Password

Content type: text/plain

Request body: json object

Description: Method will initiate merge fax delivery to specified recipients.

Example of request body:

```
{"AddressBookIds":[3,5],
"Alias":"Your message tag",
"Attachments":[{"Name":"merge_template.docx","Base64Content":"RmF4IGJyb2FkY2FzdCB0ZXN0IDE="}],
"ExternalId":"Your unique id",
"Recipients":["905350000001","905350000002"],
"ScheduleDate":"11\/07\/2013 15:00:00",
"NotificationType":1}
```

Request parameters description:

Same as for SendFaxBrodcast method.

Request parameters size limitations:

Same as for SendFax method.

Example of response:

```
{"Result":true,"ErrorCode":0,"Message":"","TaskId:"123"}
```

This method accepts only one file with attachment – template file. File should have .docx format.

For more details on how to create template file please check corresponding chapter.

Please note that in this example we are using 3 and 5 in “AddressBookIds” parameter. These numbers are “AddressBookId” values received from “AddressBook” method. Message will be sent to all recipients from these address books.

Also, please note that except address books in this example we are using 2 raw numbers, they will be added to delivery as well.

4.3.1. Using metadata and macro parameters for sending merge faxes.

For using “SendMergeFax” method you need to submit .docx or .txt template with macro parameters to substitute them with values which you have added as metadata during address book creation.

Macro parameters are numbers surrounded by percent symbol. Number is index of an item in Metadata array (starting from 1).

Example parameters for .txt template: %1%, %2%, %3%

Example parameters for .docx template: { %1% }, { %2% }, { %3% }

Please note that for .docx template parameters should be inside a field (CTRL+F9 or Insert -> Quick Parts -> Field)

For example, we have two address book records:

{"Id": 8,"AddressBookId": 7,"Phone": "905350000003","Metadata": ["John Doe","TV model 55","100$",”30$”,”5350000005”, … ,”Value N”]}

{"Id": 9,"AddressBookId": 7,"Phone": "905350000004","Metadata": ["Jane Doe","TV model 55","100$",”15$”,”5350000005”, … ,”Value N”]}

We can construct merge template like this:

Dear %1%, we just received new product – %2%, price is %3%. Your discount is %4%. Call us: %5%

Result fax messages will be:

Dear John Doe, we just received new product - TV model 55, price is 100$. Your discount is 30$. Call us: 5350000005

Dear Jane Doe, we just received new product - TV model 55, price is 100$. Your discount is 15$. Call us: 5350000005

Also, please note that maximum size of metadata array is 512 symbols (including json formatting symbols).

4.4. CancelFax

Request URI: http://rest.entera.net/api/v2/CancelFax?TaskId=134320f8d097a74aa2e38a360d688800

Request Type: POST

Required HTTP headers: Login, Password

Content type: text/plain

Request body: json object

Description: Method will initiate fax cancellation process. Please note that task can be cancelled only before start of actual delivery.

Example of response:

```
{"Result":true,"ErrorCode":0,"Message":""}
```

5. Fax reporting methods

5.1. OutboundFax

5.1.1. Requesting message status

Request URI: http://rest.entera.net/api/v2/OutboundFax?TaskId=134320f8d097a74aa2e38a360d688800

Request Type: GET

Required HTTP headers: Login, Password

Request body: empty

Description: Method will return information for requested fax message.

Example of received response:

```
{"Result": true,

"ErrorCode":0,
    "Message": null,
    "ResultCollection": [{
        "Id": 11,
        "Type": 7,
        "IncomingDate": "2014-02-19T17:19:29.75899",
        "DepartmentId": 1,
        "UserId": 1,
        "Originator": "127.0.0.1",
        "MailId": "ea33d19251fdf04eaf5c3f79d13193bf",
        "TaskId": "ea33d19251fdf04eaf5c3f79d13193bf",
        "Subject": "",
        "ScheduleDate": "0001-01-01T00:00:00",
        "ExternalId": "",
        "Priority": 3,
        "AttachmentsCount": 1,
        "RecipientsCount": 1,
        "Pages": 0,
        "PrintAttempts": 0,
        "Status": 6,
        "Price": 0.0000,
        "Cancelled": false,
        "Recipients": [{
            "Id": 11,
            "TaskId": "ea33d19251fdf04eaf5c3f79d13193bf",

            "Pages": 0,

            "Recipient": "905350000001",

            "Status": 6,

            "DeliveryAttempts": 1,

            "DisconnectReason": 1,

            "IsdnCause": 13456,

            "Price": 0.0000,

            "ProcessedDate": "2014-02-19T17:21:00",

            "DtmfData": ""

        }]

    }]}
```

5.1.2. Requesting last sent messages

Request URI: http://rest.entera.net/api/v2/OutboundFax?LastMessages=3

Request Type: GET

Required HTTP headers: Login, Password

Request body: empty

Description: Method will return last X sent fax messages.

Example of received response:

See below in 5.1.3.

5.1.3. Requesting information for specified period

Request URI: http://rest.entera.net/api/v2/OutboundFax?DateFrom=14-02-2014&DateTo=14-02-2014

Request Type: GET

Required HTTP headers: Login, Password

Request body: empty

Description: Method will return all fax messages, submitted during requested period.

Example of received response:

```
{"Result": true,
"Message": null,
"ResultCollection": [{
    "Id": 1,
    "Type": 1,
    "IncomingDate": "2014-02-14T17:34:06.713944",
    "DepartmentId": 1,
    "UserId": 1,
    "Originator": "127.0.0.1",
    "MailId": "18410139867.20140214173409@gmail.com",
    "TaskId": "7d85833c23f3b646bfa12f932d795b4e",
    "Subject": "deneme",
    "ScheduleDate": "0001-01-01T00:00:00",
    "ExternalId": "",
    "Priority": 3,
    "AttachmentsCount": 1,
    "RecipientsCount": 1,
    "Pages": 1,
    "PrintAttempts": 1,
    "Status": 6,
    "Price": 0.0000,
    "Cancelled": false,
    "Recipients": [{
        "Id": 1,
        "TaskId": "7d85833c23f3b646bfa12f932d795b4e",
        "Pages": 0,
        "Recipient": "00902123558444",
        "Status": 6,
        "DeliveryAttempts": 0,
        "DisconnectReason": 0,
        "IsdnCause": 0,
        "Price": 0.0000,
        "ProcessedDate": "0001-01-01T00:00:00",
        "DtmfData": null
    }]
}, {
    "Id": 2,
    "Type": 1,
"IncomingDate": "2014-02-14T17:34:58.911636",
"DepartmentId": 1,
"UserId": 1,
"Originator": "127.0.0.1",
"MailId": "875686285.20140214173505@gmail.com",
"TaskId": "e66c97843fd41a44838f798673572713",
"Subject": "deneme2",
"ScheduleDate": "0001-01-01T00:00:00",
"ExternalId": "",
"Priority": 3,
"AttachmentsCount": 1,
"RecipientsCount": 1,
"Pages": 1,
"PrintAttempts": 1,
"Status": 6,
"Price": 0.0000,
"Cancelled": false,
"Recipients": [{
"Id": 2,
"TaskId": "e66c97843fd41a44838f798673572713",
"Pages": 0,
"Recipient": "00902123558444",
"Status": 6,
    "DeliveryAttempts": 0,

    "DisconnectReason": 0,

    "IsdnCause": 0,

    "Price": 0.0000,

    "ProcessedDate": "0001-01-01T00:00:00",

    "DtmfData": null

}]

}]}
```

5.2. Status values description

2 – Outbound fax is pending

3 – Inbound fax is pending

4 – Inbound fax delivered

5 – Failed to deliver inbound fax

6 – Outbound fax delivered

7 - Failed to deliver outbound fax

12 - Task cancelled

13 - Error during processing

14 – Merge or broadcast fax is pending

15 – Merge or broadcast fax is processed

16 – Not enough funds to perform operation

5.3. Disconnect reason values description

0 - Ok

1 - Ok

2 - Busy

3 - Wrong number

4 - No answer

5 - No answer

6 - Wrong number

7 - Wrong number

8 - Wrong number

9 - Busy

10 - Busy

11 - Busy

12 - Busy

13 - Wrong number

14 - Busy

15 - Transmission error

16 - Not a fax device

17 - Not a fax device

18 - Transmission error

19 - Transmission error

20 - Transmission error

21 - Transmission error

22 - Transmission error

23 - Transmission error

24 - Transmission error

25 - Transmission error

26 - Transmission error

27 - No answer

28 - Transmission error

29 - Transmission error

5.4. ISDN cause values description

Please contact support for getting up to date list.

5.5. Incoming type description

1 - Fax outbound request

2 - Fax exchange request

3 - Fax inbound message received

4 - Sms outbound request

5 - Sms exchange request

6 - Sms inbound message received

7 - Ivn outbound request

8 - Fax broadcast outbound request

9 - Fax merge outbound request

10 - Sms broadcast outbound request

11 - Sms merge outbound request

6. Sending SMS methods

6.1. SendSms

Request URI: http://rest.entera.net/api/v2/SendSms

Request Type: POST

Required HTTP headers: Login, Password

Content type: text/plain

Request body: json object

Description: Method will send sms to specified recipients.

Example of request body:

```
{"Alias":"Your message tag",

"Base64Body":"Your base64 encoded message body. Body will be ignored if Base64Body is not empty.",

"Body":"Your plaintext message body. To avoid specific characters issues we recommend to use Base64Body.",

"ExternalId":"Your unique id",

"Recipients":["905350000001","905350000002"],

"ScheduleDate":"31\/12\/2013 15:00:00",

"MessageExpiration":60,

"Sender":"TEST"}
```

Request parameters description:

Alias - Message alias for intergration with external solutions (optional).

Base64Body - Base 64 encoded SMS body. Body parameter will be ignored if this parameter is not null.

Body – Plaintext SMS body. This parameter or Base64Body should be not empty.

ExternalId - External ID for integration with external solutions (optional).

Recipients - Recipients collection (mandatory).

ScheduleDate - Message can be scheduled for specific date. Date format is "dd-MM-yyyy HH:mm:ss" (optional).

MessageExpiration – Message will expire if destination phone will be offline for this period (optional).

Sender – This value will be indicated as message sender (mandatory). To register your sender please contact your account manager.

Request parameters size limitations:

Alias: 42 symbols max.

ExternalId: 42 symbols max.

Body: 2048 symbols max.

Recipient: 32 symbols max.

Sender: Alphanumeric string – 11 symbols max. Numeric string – 14 symbols max.

MessageExpiration: 2880 minutes max (48 hours).

Example of response:

```
{"Result":true,"ErrorCode":0,"Message":"","TaskId:"123"}
```

Please note that received “TaskId” can be used later to request status of this message.

6.2. SendSmsBroadcast

Request URI: http://rest.entera.net/api/v2/SendSmsBroadcast

Request Type: POST

Required HTTP headers: Login, Password

Content type: text/plain

Request body: json object

Description: Method will initiate SMS broadcast to specified recipients.

Example of request body:

```
{"AddressBookIds":[3,5],
"Alias":"Your message tag",
"Base64Body":"Your base64 encoded message body. Body will be ignored if Base64Body is not empty.",
"Body":"Your plaintext message body. To avoid specific characters issues we recommend to use Base64Body.",
"ExternalId":"Your unique id",
"Recipients":["905350000001","905350000002"],
"ScheduleDate":"11\/07\/2013 15:00:00",
"MessageExpiration":60,
"Sender":"TEST"}
```

Request parameters description:

AddressBookIds – Array of address book Id's. All recipients from these address books will be included to delivery.

Rest of parameters described in SendSms method.

Request parameters size limitations:

Same as for SendSms method.

Example of response:

```
{"Result":true,"ErrorCode":0,"Message":"","TaskId:"123"}
```

Please note that in this example we are using 3 and 5 in “AddressBookIds” parameter. These numbers are “AddressBookId” values received from “AddressBook” method. Message will be sent to all recipients from these address books.

Also, please note that except address books in this example we are using 2 raw numbers, they will be added to delivery as well.

6.2.1. Using metadata and macro parameters for sending SMS

With both “SendSms” and “SendSmsBroadcast” you can use macro parameters to substitute them with values which you have added as metadata during address book creation.

You can use 4 predefined macro parameters:

- %phone%
- %name% or %1%
- %position% or %2%
- %email% or %3%

For example, we have an address book record:

```
{"Id":6,"AddressBookId":5,"Phone":"905350000001","Metadata":["Name 1","Position 1","name1@entera.com.tr"]}
```

We can construct SMS body using these parameters:

Hello %name%, your position is: %position%, your email is: %email%, your phone is: %phone%

Or

Hello %1%, your position is: %2%, your email is: %3%, your phone is: %phone%

Our solution will transform SMS for this recipient to:

Hello Name 1, your position is: Position 1, your email is: name1@entera.com.tr, your phone is: 905350000001

Please note that number specified in “Phone” parameter will be used for actual delivery.

If you would like to use more than 4 parameters, you can use as macro parameter index of value in “Metadata” array surrounded by percent sign.

For example:

{"Id": 8,"AddressBookId": 7,"Phone": "905350000003","Metadata": ["John Doe","TV model 55","100$",”30%”,”5350000005”, … ,”Value N”]}

{"Id": 9,"AddressBookId": 7,"Phone": "905350000004","Metadata": ["Jane Doe","TV model 55","100$",”15%”,”5350000005”, … ,”Value N”]}

We can construct message like this:

Dear %1%, we just received new product - %2%, price is %3%. Your discount is %4%. Call us: %5%

Result message will be:

Dear John Doe, we just received new product - TV model 55, price is 100$. Your discount is 30%. Call us: 5350000005

Dear Jane Doe, we just received new product - TV model 55, price is 100$. Your discount is 15%. Call us: 5350000005

Please make sure that “Metadata” array contains necessary number of values, otherwise macro parameter will be ignored.

Also, please note that maximum size of metadata array is 512 symbols (including json formatting symbols).

7. SMS reporting methods

7.1. OutboundSms

7.1.1. Requesting message status

Request URI: http://rest.entera.net/api/v2/OutboundSms?TaskId=134320f8d097a74aa2e38a360d688800

Request Type: GET

Required HTTP headers: Login, Password

Request body: empty

Description: Method will return information for requested SMS message.

Example of received response:

```
{
    "Result": true,
    "ErrorCode":0,
    "Message": "",
    "Incoming": [{
        "IncomingDate": "2013-10-31T12:30:45.448745",
        "TaskId": "134320f8d097a74aa2e38a360d688800",
        "Direction": 0,
        "DepartmentId": 1,
        "UserId": 4,
        "Sender": "TEST",
        "SmsAlias": "tag 1",
        "ExternalId": "",
        "ScheduleDate": "0001-01-01T00:00:00",
        "Status": 6,
        "MessageBody": "single message",
        "SmsCount": 1,
        "RecipientsCount": 1,
        "TotalCost": 0.0001,
        "OriginatorIP": "127.0.0.1",
        "Recipients": [{
            "ProcessedDate": "2013-10-31T12:30:55",
            "TaskId": "134320f8d097a74aa2e38a360d688800",
      "Destination": "905350000001",
      "SmsCount": 1,
      "Status": 6,
      "GsmCode": 0,
      "Price": 0.0001,
      "MessageBody": ""
    }]
  }]
}
```

7.1.2. Requesting last sent messages

Request URI: http://rest.entera.net/api/v2/OutboundSms?LastMessages=3

Request Type: GET

Required HTTP headers: Login, Password

Request body: empty

Description: Method will return last X sent SMS messages.

Example of received response:

See below in 7.1.3.

7.1.3. Requesting information for specified period

Request URI: http://rest.entera.net/api/v2/OutboundSms?DateFrom=30-10-2013&DateTo=31-10-2013

Request Type: GET

Required HTTP headers: Login, Password

Request body: empty

Description: Method will return all SMS messages, submitted during requested period.

Example of received response:

```
{
  "Result": true,
  "ErrorCode":0,
  "Message": "",
  "Incoming": [{
    "IncomingDate": "2013-10-31T12:31:21.52624",
    "TaskId": "5b943868be473c49a79e824605119444",
    "Direction": 0,
    "DepartmentId": 1,
    "UserId": 4,
    "Sender": "TEST",
    "SmsAlias": "tag 3",
    "ExternalId": "",
    "ScheduleDate": "0001-01-01T00:00:00",
    "Status": 28,
    "MessageBody": "Hello %name%, your phone is: %number%",
    "SmsCount": 2,
    "RecipientsCount": 2,
        "TotalCost": 0.0001,
        "OriginatorIP": "127.0.0.1",
        "Recipients": [{
            "ProcessedDate": "2013-10-31T12:31:26",
            "TaskId": "eea9d78bdb022a4ea135a6d227670b74",
            "Destination": "905350000001",
            "SmsCount": 1,
            "Status": 6,
            "GsmCode": 0,
            "Price": 0.0001,
            "MessageBody": "Hello Name 1, your phone is: 905350000001"
        }, {
            "ProcessedDate": "2013-10-31T12:35:47",
            "TaskId": "cef0764f54fbe142b85ae38846d4db10",
            "Destination": "905350000002",
            "SmsCount": 1,
            "Status": 7,
            "GsmCode": 6,
            "Price": 0.0000,
            "MessageBody": "Hello Name 2, your phone is: 905350000002"
        }]
    }, {
        "IncomingDate": "2013-10-31T12:30:57.136576",
"TaskId": "8da7181495061c43ac0c38dc1125a348",
"Direction": 0,
"DepartmentId": 1,
"UserId": 4,
"Sender": "TEST",
"SmsAlias": "tag 2",
"ExternalId": "",
"ScheduleDate": "0001-01-01T00:00:00",
"Status": 19,
"MessageBody": "bulk test",
"SmsCount": 2,
"RecipientsCount": 2,
"TotalCost": 0.0001,
"OriginatorIP": "127.0.0.1",
"Recipients": [{
    "ProcessedDate": "2013-10-31T12:31:02",
    "TaskId": "98c51f2dbb6c284c80aa59e0eab17ded",
    "Destination": "905350000001",
    "SmsCount": 1,
    "Status": 6,
    "GsmCode": 0,
    "Price": 0.0001,
    "MessageBody": ""
```

```
            }, {
                "ProcessedDate": "2013-10-31T12:35:16",
                "TaskId": "92d2092edcf17e43a8c6c685573326e8",
                "Destination": "905350000002",
                "SmsCount": 1,
                "Status": 7,
                "GsmCode": 6,
                "Price": 0.0000,
                "MessageBody": ""
            }]
        }, {
            "IncomingDate": "2013-10-31T12:30:45.448745",
            "TaskId": "134320f8d097a74aa2e38a360d688800",
            "Direction": 0,
            "DepartmentId": 1,
            "UserId": 4,
            "Sender": "TEST",
            "SmsAlias": "tag 1",
            "ExternalId": "",
            "ScheduleDate": "0001-01-01T00:00:00",
            "Status": 6,
            "MessageBody": "single message",
            "SmsCount": 1,
        "RecipientsCount": 1,

        "TotalCost": 0.0001,

        "OriginatorIP": "127.0.0.1",

        "Recipients": [{

            "ProcessedDate": "2013-10-31T12:30:55",

            "TaskId": "134320f8d097a74aa2e38a360d688800",

            "Destination": "905350000001",

            "SmsCount": 1,

            "Status": 6,

            "GsmCode": 0,

            "Price": 0.0001,

            "MessageBody": ""

        }]

    }]

}
```

7.2. Status values description

6 - Message delivered

7 - Failed to deliver message

12 - Message cancelled

13 - Error during processing

19 - Bulk task

20 - Message is sent, but delivery confirmation not yet received

28 - Merge task

7.3. Gsm code values description

0 – No errors.

1 - No directory number in HLR.

5 - No directory number in VLR.

6 - Message was rejected because of no paging response or due to roaming restrictions.

9 - MS authentication failed.

11 - Message was rejected. No SM subscription.

12 - IMEI check failed.

13 - Rejected due to barring of the MS.

21 - Rejected due to missing provision of the SM in the VPLMN.

27 - Rejected due to no paging response.

33 - No further SC address can be added.

34 - Message rejected due to network or protocol failure.

35 - Application context is missing.

36 - Unexpected value in MAP service.

8. Sending IVR methods

8.1. SendIvr

Request URI: http://rest.entera.net/api/v2/SendIvr

Request Type: POST

Required HTTP headers: Login, Password

Content type: text/plain

Request body: json object

Description: Method will send IVR message to specified recipients.

Example of request body:

```
{"Alias":"Your message tag",
"Attachments":[{"Name":"1.wav","Base64Content":"QnVsayBmYXggdGVzdA=="}],
"ExternalId":"external id 1",
"Recipients":["905350000001"],
"ScheduleDate":"",
"MaxCallDuration":60000,
"CallAnswerTimeout":30000,
"InterDigitTimeout":3000,
"FirstDigitTimeout":6000,
"MaxDTMFDigits":3}
```

Request parameters description:

Alias - Message alias for later identifying in the GUI (optional).

Name – File name of sound message (mandatory).

Base64Content - Base 64 encoded content of IVR sound message (mandatory).

ExternalId - External ID for integration with external solutions (optional).

Recipients - Recipients collection (mandatory).

ScheduleDate - Message can be scheduled for specific date. Date format is "dd-MM-yyyy HH:mm:ss" (optional).

MaxCallDuration - Call duration (ms) - call will end after reaching this value (optional). If 0 is sent, system default value will be used - 60 seconds.

CallAnswerTimeout - Call pickup timeout (ms) (optional). If 0 is sent, system default value will be used - 20 seconds.

InterDigitTimeout - Timeout (ms) before DTMF digits (optional). If 0 is sent, system default value will be used - 3 seconds.

FirstDigitTimeout - Timeout (ms) to press first DTMF digit after main message ends (optional). If 0 is sent, system default value will be used - 6 seconds.

MaxDTMFDigits - Maximum number of DTMF to be collected from the call (optional). If 0 is send, system default value will be used - 25 digits.

Request parameters size limitations:

Alias: 255 symbols max.

ExternalId: 42 symbols max.

Name: 255 symbols max.

Recipient: 32 symbols max.

MaxCallDuration: 120000 ms.

CallAnswerTimeout: 60000 ms.

InterDigitTimeout: 6000 ms.

FirstDigitTimeout: 12000 ms.

MaxDTMFDigits: 96 digits.

Example of response:

```
{"Result":true,"ErrorCode":0,"Message":"","TaskId:"123"}
```

Please note that received “TaskId” can be used later to request status of this message.

8.2. SendIvrBroadcast

Request URI: http://rest.entera.net/api/v2/SendIvrBroadcast

Request Type: POST

Required HTTP headers: Login, Password

Content type: text/plain

Request body: json object

Description: Method will send IVR message to specified recipients.

Example of request body:

```
{"AddressBookIds":[3,5],
"Alias":"Your message tag",
"MainMessage":[{"Name":"1.wav","Base64Content":"QnVsayBmYXggdGVzdA=="}],
"ExternalId":"external id 1",
"Recipients":["905350000001"],
"NotificationType":1,
"ScheduleDate":"",
"MaxCallDuration":60000,
"CallAnswerTimeout":30000,
"InterDigitTimeout":3000,
"FirstDigitTimeout":6000,
"MaxDTMFDigits":3}
```

Request parameters description:

AddressBookIds – Array of address book Id's. All recipients from these address books will be included to delivery.

NotificationType: 0 – off, 1 – email notification.

Rest of parameters described in SendIvr method.

Request parameters size limitations:

Same as for SendIvr method.

Example of response:

```
{"Result":true,"ErrorCode":0,"Message":"","TaskId:"123"}
```

Please note that received "TaskId" can be used later to request status of this message.

8.3. CancelIvr

Request URI: http://rest.entera.net/api/v2/CancelIvr?TaskId=134320f8d097a74aa2e38a360d688800

Request Type: POST

Required HTTP headers: Login, Password

Content type: text/plain

Request body: json object

Description: Method will initiate IVR cancellation process. Please note that task can be cancelled only before start of actual delivery.

Example of response:

```
{"Result":true,"ErrorCode":0,"Message":""}
```

9. IVR reporting methods

9.1. OutboundIvr

9.1.1. Requesting message status

Request URI: http://rest.entera.net/api/v2/OutboundIvr?TaskId=94e35673411f2a439314ae59a5b35328

Request Type: GET

Required HTTP headers: Login, Password

Request body: empty

Description: Method will return information for requested IVR message.

Example of received response:

```
{"Result": true,
"ErrorCode":0,
"Message": null,
"ResultCollection": [{
    "Id": 143,
    "Type": 7,
    "IncomingDate": "2014-02-28T19:03:57.346181",
    "DepartmentId": 1,
    "UserId": 1,
    "Originator": "127.0.0.1",
    "TaskId": "94e35673411f2a439314ae59a5b35328",
    "ScheduleDate": "0001-01-01T00:00:00",
"ExternalId": "",
"Priority": 3,
"RecipientsCount": 2,
"Status": 15,
"Price": 0.0000,
"Cancelled": false,
"Recipients": [{
    "Id": 254,
    "TaskId": "c4c102ebd5e1444ea63c3d928db3e8fb",
    "Pages": 0,
    "Recipient": "05324676407",
    "Status": 6,
    "DeliveryAttempts": 1,
    "DisconnectReason": 1,
    "IsdnCause": 13456,
    "Price": 0.0000,
    "ProcessedDate": "2014-02-28T19:06:37",
    "DtmfData": "42"
}, {
    "Id": 253,
    "TaskId": "93cf0292d35af44095b2b30f3aad7562",
    "Pages": 0,
    "Recipient": "05324676407",
        "Status": 7,

        "DeliveryAttempts": 1,

        "DisconnectReason": 2,

        "IsdnCause": 13457,

        "Price": 0.0000,

        "ProcessedDate": "2014-02-28T19:06:52",

        "DtmfData": null

      }]

    }]}
```

9.1.2. Requesting last sent messages

Request URI: http://rest.entera.net/api/v2/OutboundIvr?LastMessages=3

Request Type: GET

Required HTTP headers: Login, Password

Request body: empty

Description: Method will return last X sent IVR messages.

Example of received response:

See below in 9.1.3.

9.1.3. Requesting information for specified period

Request URI: http://rest.entera.net/api/v2/OutboundIvr?DateFrom=28-02-2014&DateTo=28-02-2014

Request Type: GET

Required HTTP headers: Login, Password

Request body: empty

Description: Method will return all IVR messages, submitted during requested period.

Example of received response:

```
{"Result": true,

"Message": null,

"ResultCollection": [{

    "Id": 145,

    "Type": 7,

    "IncomingDate": "2014-02-28T19:18:53.302554",

    "DepartmentId": 1,

    "UserId": 1,

    "Originator": "127.0.0.1",

    "TaskId": "f5857dfe42a76f4a891a8c13214a4d11",

    "ScheduleDate": "0001-01-01T00:00:00",

    "ExternalId": "",

    "Priority": 3,

    "RecipientsCount": 2,

    "Status": 15,
"Price": 0.0000,

"Cancelled": false,

"Recipients": [{

    "Id": 257,

    "TaskId": "8e69ee72886ce64980cd2c0d592b6cac",

    "Pages": 0,

    "Recipient": "05324676407",

    "Status": 6,

    "DeliveryAttempts": 1,

    "DisconnectReason": 1,

    "IsdnCause": 13456,

    "Price": 0.0000,

    "ProcessedDate": "2014-02-28T19:19:23",

    "DtmfData": "333"

}, {

    "Id": 256,

    "TaskId": "7d987009884ceb488caf5e55a5305670",

    "Pages": 0,

    "Recipient": "05324676407",

    "Status": 7,

    "DeliveryAttempts": 1,

    "DisconnectReason": 2,

    "IsdnCause": 13457,
            "Price": 0.0000,

            "ProcessedDate": "2014-02-28T19:19:47",

            "DtmfData": null

        }]

    }, {

        "Id": 144,

        "Type": 7,

        "IncomingDate": "2014-02-28T19:18:23.412902",

        "DepartmentId": 1,

        "UserId": 1,

        "Originator": "127.0.0.1",

        "TaskId": "411658e5a980ae4fa42feb0f061c64ca",

        "ScheduleDate": "0001-01-01T00:00:00",

        "ExternalId": "",

        "Priority": 3,

        "RecipientsCount": 1,

        "Status": 6,

        "Price": 0.0000,

        "Cancelled": false,

        "Recipients": [{

            "Id": 255,

            "TaskId": "411658e5a980ae4fa42feb0f061c64ca",

            "Pages": 0,
        "Recipient": "05324676407",
        "Status": 6,
        "DeliveryAttempts": 1,
        "DisconnectReason": 1,
        "IsdnCause": 13456,
        "Price": 0.0000,
        "ProcessedDate": "2014-02-28T19:18:52",
        "DtmfData": "123"
      }]
    }]}
```

9.2. Status values description

You can find this information in chapter 5.2

9.3. Disconnect reason values description

You can find this information in chapter 5.3

9.4. ISDN cause values description

You can find this information in chapter 5.4

9.5. Incoming type description

You can find this information in chapter 5.5

10. Error codes description

0 – No errors.

1 – Error during deserialization of POST request. Please check your HTTP POST code.

2 – Authorization error. Please check your credentials.

3 – Parameters validation error. Please check your values.

4 – Parameters exceeding limit values. Please check your values.

5 – Internal error: failed to load config file. Please contact support.

6 – Internal error: failed to get user. Please contact support.

7 – Internal error: failed to get department. Please contact support.

8 – Customer plan unassigned. Please contact support.

9 – Permissions error. Please contact support.

10 – Schedule date error. Please check date value.

11 – All recipients refused. Please contact support.

12 – Not enough credits. Please contact support.

13 – Internal error: Post request validation failed. Please contact support.

14 – Unexpected error. Please contact support.

15 – Request not contains any changes.

16 – Internal error: Database layer. Please contact support.

17 – Operation forbidden for this user. Please contact support.

18 – User not active. Please contact support.

19 – Method response is empty.