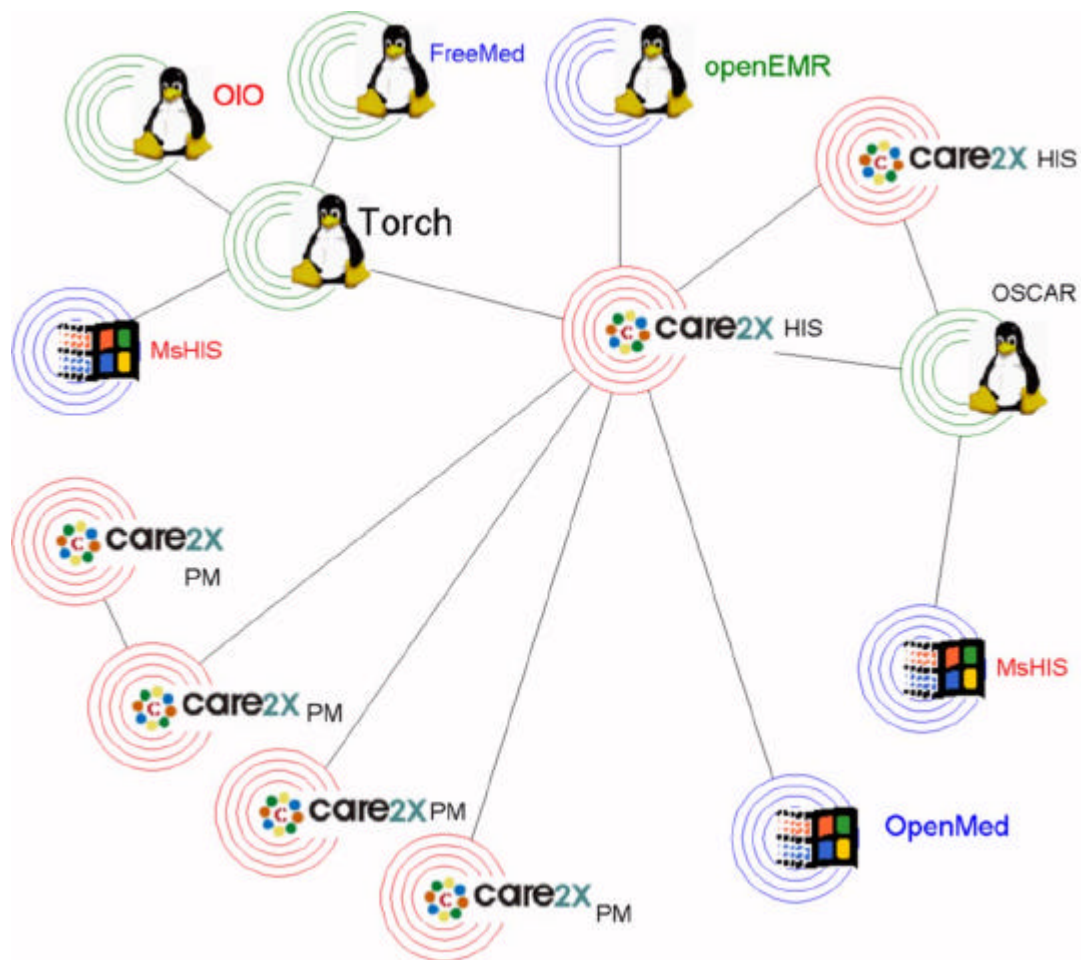


HXP - PCD

Healthcare Xchange Protocol
Procedure Call Dictionary
Draft proposal 0.1



March 14, 2004

Table of Contents

1.0 Introduction

2.0 HXP Error Reporting

3.0 Heading Parameter

4.0 Person Calls

5.0 Encounter Calls

6.0 Electronic Medical Record Calls (EMR)

7.0 Department Calls

8.0 Ward Calls

9.0 Room Calls

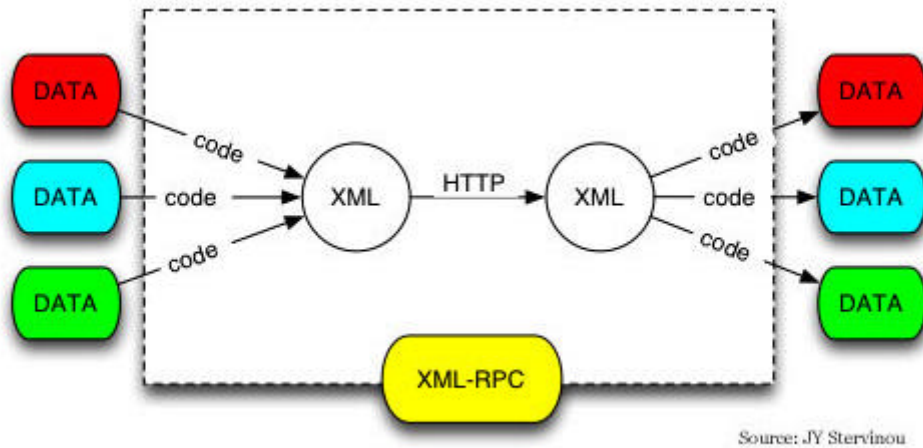
10.0 System Information Calls

11.0 Anonymized Data Calls

1.0 Introduction

On the programming level, HXP uses XML-RPC (Remote Procedure Calling protocol) that works over the Internet, Intranet and local host.

An XML-RPC message is an HTTP-POST request. The body of the request is in XML. A procedure executes on the server and the value it returns is also formatted in XML.



Procedure parameters can be scalars, numbers, strings, dates, etc.; and can also be complex record and list structures.

2.0 HXP Error Reporting

Error codes of HXP start at 1000. The following codes are draft proposals version 0.1.

| Error code | Error string | Meaning |
|------------|-----------------------|--|
| 1000 | _ERROR_AUTH_BADHEADER | Bad or missing header argument/parameter |
| 1001 | _ERROR_AUTH_USER | Wrong user. |
| 1002 | _ERROR_AUTH_PW | Wrong password. |
| 1003 | _ERROR_AUTH_AREA | No permission for the protected area. |
| 1004 | _ERROR_AUTH_LOCKED | The user permission is locked. |
| 1005 | _ERROR_SQL_RESULT | The sql query failed. |
| 1006 | _ERROR_NORESULT | The procedure returned no result. |
| 1007 | _ERROR_PROC_NOSUPPORT | The procedure is not supported |
| 1008 | _ERROR_PROC_NOEXISTS | The procedure is not available |

2.1 Error handling

It is important that the client create a common routine to handle possible error messages being returned by the responding server.

3.0 Heading Parameter

Each procedure call contains a heading parameter as the first parameter. This parameter is a struct and must contain by default the following keys:

Compulsary keys and data.

| <i>Key</i> | <i>Data</i> |
|------------|-------------|
| usr | Username |
| pw | Password |

When the header is missing, the application will return an `_ERROR_AUTH_BADHEADER` error message and 1000 error code.

Optional keys and data:

| <i>Key</i> | <i>Data</i> |
|------------|---------------|
| lang | Language code |
| sid | session ID |
| version | HXP version |

The availability of optional keys can be detected either on the fly or during configuration by using the following system procedure calls:

```
System.Header.Keys.lang  
System.Header.Keys.sid  
System.Header.Keys.version
```

If the key is supported by the application, the returned value should be integer 1.

Other keys and data can be added according to the application's design but these are not considered standard and should be documented and published.

3.1 Sample call with header parameter (php, no error handling)

```
$header['usr']='demo';  
$header['pw']='demo';  
  
$hxpclient->query('Person',$header,2004560032);  
  
$response = $hxp_client->getResponse();  
  
echo $response['name_last'];
```

4.0 Person Calls

Person

Returns a struct of the person's registration data:

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | struct |
| sample php query | <code>\$hxpclient->query('Person',\$header,\$PID)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$pid = 20031009900;

$hxpclient->query('Person',$header,$pid);

$response=$hxpclient->getResponse();

echo $response['name_last'];
```

Returned struct has the following keys:

| | |
|------------------|--|
| pid | Person Identifier (numeric) |
| name_last | Last or family name |
| name_first | First or given name |
| name_middle | Middle name |
| name_2 | Second name |
| name_3 | Third name |
| name_maiden | Maiden name |
| name_others | Other name or aliases |
| sex | Sex (m = male), (f = female) |
| date_birth | Date of birth (YYYY-MM-DD) |
| title | Title |
| date_reg | Registration date (YYYY-MM-DD) |
| blood_group | Blood group (A,B,AB,O) |
| addr_str | Address: street name |
| addr_str_nr | Address: street number |
| addr_zip | Address: Zip code |
| addr_citytown_nr | Address: City or town's number (usually for internal use only) |
| phone_1_code | Phone area code #1 |

| | |
|--------------------|---|
| pid | Person Identifier (numeric) |
| phone_1_nr | Phone number #1 |
| phone_2_code | Phone area code #2 |
| phone_2_nr | Phone number #2 |
| cellphone_1_nr | Cellphone #1 |
| cellphone_2_nr | Cellphone #2 |
| fax | Fax number |
| email | Email address |
| civil_status | Civil status (single, married, divorced, widowed) |
| photo_filename | Filename of the stored photo ID |
| ethnic_orig | Ethnic origin |
| sss_nr | Social security system number/ID |
| nat_id_nr | National ID number |
| religion | Religion |
| mother_pid | Mother's PID |
| father_pid | Father's PID |
| contact_person | Contact person's name |
| contact_pid | Contact person's PID |
| contact_relation | Relation to contact person |
| death_date | Date of person's death |
| death_encounter_nr | Related encounter number related to death |
| death_cause | Cause of death (free text) |
| death_cause_code | Cause of death code (e.g. ICD10) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmddhhmmss) |

Person.Basic

Returns the person's basic data.

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | struct |
| sample php query | <code>\$hxpclient->query('Person.Basic',\$header,\$PID)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$pid = 20031009900;

$hxpclient->query('Person.Basic',$header,$pid);

$response=$hxpclient->getResponse();

echo $response['addr_str'];
```

Returns a struct of person's registration data with the following keys:

| | |
|------------------|--|
| pid | Person Identifier (numeric) |
| name_last | Last or family name |
| name_first | First or given name |
| name_middle | Middle name |
| name_2 | Second name |
| name_3 | Third name |
| name_maiden | Maiden name |
| name_others | Other name or aliases |
| sex | Sex (m = male), (f = female) |
| date_birth | Date of birth (YYYY-MM-DD) |
| title | Title |
| blood_group | Blood group (A,B,AB,O) |
| addr_str | Address: street name |
| addr_str_nr | Address: street number |
| addr_zip | Address: Zip code |
| addr_citytown_nr | Address: City or town's number (usually for internal use only) |
| photo_filename | Filename of the stored photo ID |

Person.List

Returns an array of struct containing the data of persons who are not hidden, nor deleted, nor locked.

| | |
|------------------|--|
| parameters | ((struct) header) |
| return | Array of struct |
| sample php query | <code>\$hxpclient->query('Person.List',\$header)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$hxpclient->query('Person.List',$header);

$response=$hxpclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}

```

The struct contains the following keys:

| | |
|------------------|--|
| pid | Person Identifier (numeric) |
| name_last | Last or family name |
| name_first | First or given name |
| name_middle | Middle name |
| name_2 | Second name |
| name_3 | Third name |
| name_maiden | Maiden name |
| name_others | Other name or aliases |
| sex | Sex (m = male), (f = female) |
| date_birth | Date of birth (YYYY-MM-DD) |
| title | Title |
| date_reg | Registration date (YYYY-MM-DD) |
| blood_group | Blood group (A,B,AB,O) |
| addr_str | Address: street name |
| addr_str_nr | Address: street number |
| addr_zip | Address: Zip code |
| addr_citytown_nr | Address: City or town's number (usually for internal use only) |
| phone_1_code | Phone area code #1 |
| phone_1_nr | Phone number #1 |
| phone_2_code | Phone area code #2 |
| phone_2_nr | Phone number #2 |
| cellphone_1_nr | Cellphone #1 |
| cellphone_2_nr | Cellphone #2 |
| fax | Fax number |
| email | Email address |

| | |
|--------------------|---|
| pid | Person Identifier (numeric) |
| civil_status | Civil status (single, married, divorced, widowed) |
| photo_filename | Filename of the stored photo ID |
| ethnic_orig | Ethnic origin |
| sss_nr | Social security system number/ID |
| nat_id_nr | National ID number |
| religion | Religion |
| mother_pid | Mother's PID |
| father_pid | Father's PID |
| contact_person | Contact person's name |
| contact_pid | Contact person's PID |
| contact_relation | Relation to contact person |
| death_date | Date of person's death |
| death_encounter_nr | Related encounter number related to death |
| death_cause | Cause of death (free text) |
| death_cause_code | Cause of death code (e.g. ICD10) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmddhhmmss) |

Person.Encounter.ENR

Person.Encounter.ENR.List

Alias of Person.Encounter.ENR

Person.Encounter.ENR.Current

Returns the person's current encounter number (numeric).

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | (integer) Current encounter number |
| sample php query | <code>\$hxpclient->query('Person.Encounter.ENR.Current',\$header,\$PID)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';
```

```

$hxpcclient->query('Person.Encounter.ENR.Current',$header);

$response = $hxpcclient->getResponse();

echo $response;

```

Person.Encounter.Search

Reserved

Person.Search

Returns an array of structs containing the data of persons who are not hidden,deleted, nor locked.

| | |
|------------------|---|
| parameters | ((struct) header, (string) keyword) |
| return | Array of struct |
| sample php query | <code>\$hxpcclient->query('Person.Search',\$header,\$keyword)</code> |

The struct contains the following keys:

| | |
|------------|------------------------------|
| pid | Person Identifier (numeric) |
| name_last | Last or family name |
| name_first | First or given name |
| sex | Sex (m = male), (f = female) |
| date_birth | Date of birth (YYYY-MM-DD) |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$keyword = 'adam';

$hxpcclient->query('Person.Search',$header,$keyword);

$response=$hxpcclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}

```

Person.Name.First

Returns the person's first (or given) name.

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpcclient->query('Person.Name.First',\$header,\$PID)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Name.First',$header,$pid);

echo $hxpclient->getResponse();
```

Person.Name.Family

Person.Name.Family(header,PID) = call

Returns the person's family (or last) name.

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Name.Family',$header,$pid);

echo $hxpclient->getResponse();
```

Person.Name.Middle

Returns the person's middle name.

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpclient->query('Person.Name.Middle',\$header,\$PID)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Name.Middle',$header,$pid);

echo $hxpclient->getResponse();
```

Person.Name.Maiden

Returns the person's maiden name (family name by birth).

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpclient->query('Person.Name.Maiden',\$header,\$PID)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Name.Maiden',$header,$pid);

echo $hxpclient->getResponse();

```

Person.Name.Second

Returns the person's second name.

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpclient->query('Person.Name.Second',\$header,\$PID)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Name.Second',$header,$pid);

echo $hxpclient->getResponse();

```

Person.Name.Third

Returns the person's third name.

| | |
|------------------|--|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpclient->query('Person.Name.Third',\$header,\$PID)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Name.Third',$header,$pid);

```

```
echo $hxpclient->getResponse();
```

Person.Name.Others

Returns the person's other names.

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpclient->query('Person.Name.Others',\$header,\$PID)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';  
$header['pw']='demopass';  
  
$pid = 100300455;  
  
$hxpclient->query('Person.Name.Others',$header,$pid);  
  
echo $hxpclient->getResponse();
```

Person.Birth.Date

Returns the person's date of birth in YYYY-MM-DD format.

| | |
|------------------|--|
| parameters | ((struct) header, (int) PID) |
| return | (string) |
| sample php query | <code>\$hxpclient->query('Person.Birth.Date',\$header,\$PID)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';  
$header['pw']='demopass';  
  
$pid = 100300455;  
  
$hxpclient->query('Person.Birth.Date',$header,$pid);  
  
echo $hxpclient->getResponse();
```

Person.Birth.Details

Reserved

Person.Died.Date

Returns the person's date when he died.

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpclient->query('Person.Died.Date',\$header,\$PID)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Died.Date',$header,$pid);

echo $hxpclient->getResponse();
```

Person.Died.Cause

Returns the cause of person's death in free text.

| | |
|------------------|--|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpclient->query('Person.Died.Cause',\$header,\$PID)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Died.Cause',$header,$pid);

echo $hxpclient->getResponse();
```

Person.Died.Cause.Code

Returns the cause of death code.

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | struct |
| sample php query | <code>\$hxpclient->query('Person.Died.Cause.Code',\$header,\$PID)</code> |

The returned struct contains the following keys:

| <i>Key</i> | <i>data</i> |
|------------|--------------------------------------|
| code | The code |
| type | The code type e.g. ICD9, ICD10, etc. |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Died.Cause.Code',$header,$pid);
```

```

$response = $hxpclient->getResponse();

echo $response['code'];
echo $response['type'];

```

Person.Died.Encounter.ENR

Returns the encounter number related to the person's death.

| | |
|------------------|--|
| parameters | ((struct) header, (int) PID) |
| return | (integer) |
| sample php query | \$hxpclient->query('Person.Died.Encounter.ENR',\$header,\$PID) |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Died.Encounter.ENR',$header,$pid);

echo $hxpclient->getResponse();

```

Person.Registry.Date

Returns the date of person's registration in the database in YYYY-MM-DD format.

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | (string) |
| sample php query | \$hxpclient->query('Person.Registry.Date',\$header,\$PID) |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Registry.Date',$header,$pid);

echo $hxpclient->getResponse();

```

Person.Appointment.List

Returns an array of struct containing the person's appointments data which are not hidden, nor locked, nor deleted.

| | |
|------------------|--|
| parameters | ((struct) header, (int) PID) |
| return | (string) |
| sample php query | \$hxpclient->query('Person.Appointment.List',\$header,\$PID) |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$response = $hxpclient->query('Person.Registry.Date',$header,$pid);

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}
```

Person.Appointment

Returns a struct of person's appointments based on a supplied primary key.

| | |
|------------------|---|
| parameters | ((struct) header, (int) key) |
| return | (string) |
| sample php query | \$hxpclient->query('Person.Registry.Date',\$header,\$key) |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$response = $hxpclient->query('Person.Registry.Date',$header,$pid);

while(list($key,$data)=each($response)){
    echo " $key >>> $data ";
}
```

Person.Address

Person.Address(header,PID) = call

Returns a struct containing the person's address.

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | (string) |
| sample php query | \$hxpclient->query('Person.Address',\$header,\$PID) |

The returned struct contains the following keys:

| <i>Key</i> | <i>Data</i> |
|-------------|---------------|
| addr_str | Street name |
| addr_str_nr | Street number |
| addr_zip | Zip code |

| <i>Key</i> | <i>Data</i> |
|------------------|---------------------------------|
| addr_citytown_nr | Code number of the city or town |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Address',$header,$pid);

$response = $hxpclient->getResponse();

echo $response['addr_str'];
echo $response['addr_str_nr'];

```

Person.Address.FullText

Returns the person's full address in text format.

| | |
|------------------|--|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | \$hxpclient->query('Person.Address.FullText',\$header,\$PID) |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Address.FullText',$header,$pid);

echo $hxpclient->getResponse();

```

Person.Address.Street

Returns the address street name.

| | |
|------------------|--|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | \$hxpclient->query('Person.Address.Street',\$header,\$PID) |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Address.Street',$header,$pid);

```

```
echo $hxpclient->getResponse();
```

Person.Address.Street.Nr

Returns the address street number.

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpclient->query('Person.Address.Street.Nr',\$header,\$PID)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';  
$header['pw']='demopass';  
  
$pid = 100300455;  
  
$hxpclient->query('Person.Address.Street.Nr',$header,$pid);  
  
echo $hxpclient->getResponse();
```

Person.Address.CityTown

Returns the city or town's name.

| | |
|------------------|--|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpclient->query('Person.Address.CityTown',\$header,\$PID)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';  
$header['pw']='demopass';  
  
$pid = 100300455;  
  
$hxpclient->query('Person.Address.CityTown',$header,$pid);  
  
echo $hxpclient->getResponse();
```

Person.Address.Zip

Returns the zip code.

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpclient->query('Person.Address.Zip',\$header,\$PID)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Address.Zip',$header,$pid);

echo $hxpclient->getResponse();

```

Person.Address.Email

Returns the email address.

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpclient->query('Person.Address.Email',\$header,\$PID)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Address.Email',$header,$pid);

echo $hxpclient->getResponse();

```

Person.Phone.1

Returns the first telephone number.

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpclient->query('Person.Phone.1',\$header,\$PID)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Phone.1',$header,$pid);

echo $hxpclient->getResponse();

```

Person.Phone.1.Area.Code

Returns the first telephone area code.

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpclient->query('Person.Phone.1.Area.Code',\$header,\$PID)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Phone.1.Area.Code',$header,$pid);

echo $hxpclient->getResponse();

```

Person.Phone.2

Returns the second telephone number.

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpclient->query('Person.Phone.2',\$header,\$PID)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Phone.2',$header,$pid);

echo $hxpclient->getResponse();

```

Person.Phone.2.Area.Code

Returns the first telephone area code.

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpclient->query('Person.Phone.2.Area.Code',\$header,\$PID)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Phone.2.Area.Code',$header,$pid);

```

```
echo $hxpclient->getResponse();
```

Person.Phone.Cell.1

Returns the second cellphone number.

| | |
|------------------|--|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpclient->query('Person.Phone.Cell.1',\$header,\$PID)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Phone.Cell.1',$header,$pid);

echo $hxpclient->getResponse();
```

Person.Phone.Cell.2

Returns the second cellphone number.

| | |
|------------------|--|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpclient->query('Person.Phone.Cell.2',\$header,\$PID)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Phone.Cell.2',$header,$pid);

echo $hxpclient->getResponse();
```

Person.Fax.Nr

Returns the fax number.

| | |
|------------------|--|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpclient->query('Person.Fax.Nr',\$header,\$PID)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Fax.Nr',$header,$pid);

echo $hxpclient->getResponse();

```

Person.Blood.Group

Returns the fax number.

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | String (A, B, AB, O) |
| sample php query | <code>\$hxpclient->query('Person.Blood.Group',\$header,\$PID)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Blood.Group',$header,$pid);

echo $hxpclient->getResponse();

```

Person.Mother.PID

Returns the mother's PID.

| | |
|------------------|--|
| parameters | ((struct) header, (int) PID) |
| return | integer |
| sample php query | <code>\$hxpclient->query('Person.Mother.PID',\$header,\$PID)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Mother.PID',$header,$pid);

echo $hxpclient->getResponse();

```

Person.Father.PID

Returns the father's PID.

| | |
|------------------|--|
| parameters | ((struct) header, (int) PID) |
| return | Integer |
| sample php query | <code>\$hxpclient->query('Person.Father.PID',\$header,\$PID)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Mother.PID',$header,$pid);

echo $hxpclient->getResponse();

```

Person.Insurance.1.Nr

Returns the first insurance number.

| | |
|------------------|--|
| parameters | ((struct) header, (int) PID) |
| return | integer |
| sample php query | <code>\$hxpclient->query('Person.Insurance.1.Nr',\$header,\$PID)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Insurance.1.Nr',$header,$pid);

echo $hxpclient->getResponse();

```

Person.Insurance.1.Co.ID

Returns the first insurance company ID.

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpclient->query('Person.Insurance.1.Co.ID',\$header,\$PID)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Insurance.1.Co.ID',$header,$pid);

```

```
echo $hxpclient->getResponse();
```

Person.Insurance.2.Nr

Returns the second insurance number.

| | |
|------------------|--|
| parameters | ((struct) header, (int) PID) |
| return | integer |
| sample php query | <code>\$hxpclient->query('Person.Insurance.2.Nr',\$header,\$PID)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Insurance.2.Nr',$header,$pid);

echo $hxpclient->getResponse();
```

Person.Insurance.2.Co.ID

Returns the second insurance company ID.

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpclient->query('Person.Insurance.2.Co.ID',\$header,\$PID)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Insurance.2.Co.ID',$header,$pid);

echo $hxpclient->getResponse();
```

Person.Contact

Returns the contact person's name.

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpclient->query('Person.Contact',\$header,\$PID)</code> |

Sample use (php, without error handling)


```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Contact',$header,$pid);

echo $hxpclient->getResponse();

```

Person.Contact.PID

Returns the contact person's PID.

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | integer |
| sample php query | <code>\$hxpclient->query('Person.Contact.PID',\$header,\$PID)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Contact.PID',$header,$pid);

echo $hxpclient->getResponse();

```

Person.Contact.Relation

Returns the relation to the contact person.

| | |
|------------------|--|
| parameters | ((struct) header, (int) PID) |
| return | string |
| sample php query | <code>\$hxpclient->query('Person.Contact.Relation',\$header,\$PID)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Contact.Relation',$header,$pid);

echo $hxpclient->getResponse();

```

Person.Add

Add new person registration data.

| | |
|------------------|---|
| parameters | ((struct) header, (struct) regdata) |
| return | (integer) PID |
| sample php query | <code>\$hxpcient->query('Person.Contact.Relation',\$header,\$PID)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$regdata['name_first']= 'Smith';
$regdata['name_last']= 'Peter';
$regdata['date_birth']= '2001-11-03';

$hxpcient->query('Person.Add',$header,$regdata);

```

Person.Update

Update person registration data.

| | |
|------------------|---|
| parameters | ((struct) header, (struct) regdata) |
| return | character |
| sample php query | <code>\$hxpcient->query('Person.Contact.Relation',\$header,\$PID)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$regdata['name_first']= 'Smith';
$regdata['name_last']= 'Peter';
$regdata['date_birth']= '2001-11-03';

$hxpcient->query('Person.Update',$header,$regdata);

```

Person.Record.Hide

Flags the record as "hidden".

| | |
|------------------|--|
| parameters | ((struct) header, (int) PID) |
| return | character |
| sample php query | <code>\$hxpcient->query('Person.Record.Hide',\$header,\$PID)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpcient->query('Person.Record.Hide',$header,$pid);

```

Person.Record.Lock

Flags the record as "locked".

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | character |
| sample php query | <code>\$hxpclient->query('Person.Record.Lock',\$header,\$PID)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Record.Lock',$header,$pid);
```

Person.Record.Delete

Flags the record as "deleted".

| | |
|------------------|---|
| parameters | ((struct) header, (int) PID) |
| return | character |
| sample php query | <code>\$hxpclient->query('Person.Record.Delete',\$header,\$PID)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$pid = 100300455;

$hxpclient->query('Person.Record.Delete',$header,$pid);
```

5.0 Encounter Calls

Encounter

Returns a struct of the patient's encounter (admission) data:

*ENR = encounter number

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | struct |
| sample php query | <code>\$hxpclient->query('Encounter',\$header,\$ENR)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$pid = 20031009900;

$hxpclient->query('Encounter',$header,$pid);

$response=$hxpclient->getResponse();

echo $response['name_last'];

```

Returned struct contains the following keys and data:

| <i>key</i> | <i>data</i> |
|------------------------|--|
| encounter_nr | (integer) Encounter number |
| pid | (integer) PID (person identifier) |
| name_last | (string) Last or family name |
| name_first | (string) First or given name |
| date_birth | (string) Date of birth (YYYY-MM-DD) |
| sex | (char) Sex (m = male, f = female) |
| addr_str | (string) Street name |
| addr_str_nr | (string) Street number |
| addr_zip | (string) Zip code |
| citytown_name | (string) City or town's name |
| blood_group | (string) Blood group (A, B, AB, O) |
| encounter_date | Date of encounter (YYYY-MM-DD) |
| encounter_class_nr | (integer) 1 = inpatient, 2 = outpatient |
| encounter_type | (string) Encounter type e.g. emergency, normal |
| encounter_status | (string) Encounter status |
| Referrer_diagnosis | (string) Referrer's prelim diagnosis |
| Referrer_recom_therapy | (string) Referrer's recommended therapy |
| Referrer_dr | (string) Referrer's named |
| Referrer_institution | (string) Referring institution |
| Referrer_notes | (string) Referrer's notes |
| financial_class | (integer) Financial class |
| insurance_nr | (string) Insurance number |
| insurance_class_nr | (integer) Insurance class number |
| insurance_firm_id | (string) Insurance firm ID |
| insurance_2_nr | (string) Second insurance number |
| insurance_2_firm_id | (string) Second insurance firm ID |
| current_ward_nr | (integer) Current ward number (foreign key) |

| <i>key</i> | <i>data</i> |
|-------------------------|---|
| current_room_nr | (integer) Current room nr (foreign key) |
| in_ward | (integer) Flag if patient is admitted in ward |
| current_dept_nr | (integer) Current department number (foreign key) |
| current_firm_nr | (integer) Current firm number (foreign key) |
| current_att_dr | (integer) Current attending doctor (foreign key) |
| consulting_dr | (string) Consulting doctor |
| extra_service | (integer) Extra service (foreign key) |
| followup_date | (string) Follow up date (YYYY-MM-DD) |
| followup_responsibility | (string) Resonsible for follow up |
| post_encounter_notes | (string) Post encounter notes |
| photo_filename | (string) Filename of uploaded photo ID |
| death_date | (string) Date of death (YYYY-MM-DD) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYMMDDhhmmss) |

Encounter.List

Returns an array of structs containing the encounter data of patients who are not discharged, hidden,deleted, nor locked.

| | |
|------------------|--|
| parameters | ((struct) header) |
| return | array of struct |
| sample php query | <code>\$hxpclient->query('Encounter',\$header)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$hxpclient->query('Encounter.List',$header);

$response=$hxpclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}

```

The struct contains the following keys:

| <i>key</i> | <i>data</i> |
|--------------|-------------------------------------|
| encounter_nr | (integer) Encounter number |
| pid | (integer) PID (person identifier) |
| name_last | (string) Last or family name |
| name_first | (string) First or given name |
| date_birth | (string) Date of birth (YYYY-MM-DD) |
| sex | (char) Sex (m = male, f = female) |
| addr_zip | (string) Zip code |
| blood_group | (string) Blood group (A, B, AB, O) |

Encounter.Outpatient.List

Returns an array of structs containing the encounter data of **outpatients** who are not discharged, hidden, deleted, nor locked.

| | |
|------------------|--|
| parameters | ((struct) header) |
| return | array of struct |
| sample php query | <code>\$hxpclient->query('Encounter',\$header)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$hxpclient->query('Encounter.Outpatient.List',$header);

$response=$hxpclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}

```

The struct contains the following keys:

| <i>key</i> | <i>data</i> |
|--------------|-------------------------------------|
| encounter_nr | (integer) Encounter number |
| pid | (integer) PID (person identifier) |
| name_last | (string) Last or family name |
| name_first | (string) First or given name |
| date_birth | (string) Date of birth (YYYY-MM-DD) |
| sex | (char) Sex (m = male, f = female) |
| addr_zip | (string) Zip code |

| <i>key</i> | <i>data</i> |
|-------------|------------------------------------|
| blood_group | (string) Blood group (A, B, AB, O) |

Encounter.Inpatient.List

Returns an array of structs containing the encounter data of **inpatients** who are not discharged, hidden, deleted, nor locked.

| | |
|------------------|--|
| parameters | ((struct) header) |
| return | array of struct |
| sample php query | <code>\$hxpclient->query('Encounter',\$header)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$hxpclient->query('Encounter.Inpatient.List',$header);

$response=$hxpclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}

```

The struct contains the following keys:

| <i>key</i> | <i>data</i> |
|--------------|-------------------------------------|
| encounter_nr | (integer) Encounter number |
| pid | (integer) PID (person identifier) |
| name_last | (string) Last or family name |
| name_first | (string) First or given name |
| date_birth | (string) Date of birth (YYYY-MM-DD) |
| sex | (char) Sex (m = male, f = female) |
| addr_zip | (string) Zip code |
| blood_group | (string) Blood group (A, B, AB, O) |

Encounter.Exists

Returns the PID number if the encounter exists, otherwise returns zero or empty string.

*ENR = encounter number

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | (integer) PID (or zero or empty string) |
| sample php query | <code>\$hxpclient->query('Encounter',\$header,\$ENR)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;
```

Encounter.Search

Searches encounters basing on a supplied keyword and returns an array of structs containing the encounter data of patients who are not discharged, hidden,deleted, nor locked.

| | |
|------------------|---|
| parameters | ((struct) header , (string) keyword) |
| return | array of struct |
| sample php query | <code>\$hxpclient->query('Encounter.Search',\$header)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$hxpclient->query('Encounter.Search',$header, $keyword);

$response=$hxpclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}
```

The struct contains the following keys:

| <i>key</i> | <i>data</i> |
|--------------|-------------------------------------|
| encounter_nr | (integer) Encounter number |
| pid | (integer) PID (person identifier) |
| name_last | (string) Last or family name |
| name_first | (string) First or given name |
| date_birth | (string) Date of birth (YYYY-MM-DD) |
| sex | (char) Sex (m = male, f = female) |
| addr_zip | (string) Zip code |

| <i>key</i> | <i>data</i> |
|-------------|------------------------------------|
| blood_group | (string) Blood group (A, B, AB, O) |

Encounter.Status

Returns the status of the encounter (not the record).

*ENR = encounter number

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | (integer) PID (or zero or empty string) |
| sample php query | <code>\$hxpclient->query('Encounter.Status',\$header,\$ENR)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Status',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;

```

Encounter.PID

Returns the PID number of the encounter.

*ENR = encounter number

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | (integer) PID |
| sample php query | <code>\$hxpclient->query('Encounter.PID',\$header,\$ENR)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.PID',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;

```

Encounter.Admission.Date

Returns the admission date of encounter.

*ENR = encounter number

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | (string) Date of admission (YYYY-MM-DD) |
| sample php query | <code>\$hxpclient->query('Encounter.Admission.Date',\$header,\$ENR)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Admission.Date',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;
```

Encounter.Admission.Type

Returns the admission type of encounter.

*ENR = encounter number

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | (string) Date of admission (YYYY-MM-DD) |
| sample php query | <code>\$hxpclient->query('Encounter.Admission.Type',\$header,\$ENR)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Admission.Date',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;
```

Encounter.Discharge.Time

Returns the time of discharge.

*ENR = encounter number

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | (string) Date of admission (HH:mm:ss) |
| sample php query | <code>\$hxpclient->query('Encounter.Discharge.Time',\$header,\$ENR)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Discharge.Time',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;
```

Encounter.Discharge.Is

Returns the time of discharge.

*ENR = encounter number

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | (string) Date of admission (HH:mm:SS) |
| sample php query | <code>\$hxpclient->query('Encounter.Discharge.Time',\$header,\$ENR)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Discharge.Time',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;
```

Encounter.Discharge.Type

Returns the type of discharge.

*ENR = encounter number

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | (integer) Type of discharge (foreign key) |
| sample php query | <code>\$hxpclient->query('Encounter.Discharge.Type',\$header,\$ENR)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;
```

```

$hxpcclient->query('Encounter.Discharge.Type',$header,$ENR);

$response=$hxpcclient->getResponse();

echo $response;

```

Encounter.Payment.Type

Returns the type of payment (financial class).

*ENR = encounter number

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | (integer) Type of payment (foreign key) |
| sample php query | \$hxpcclient->query('Encounter.Payment.Type',\$header,\$ENR) |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpcclient->query('Encounter.Payment.Type ', $header, $ENR);

$response=$hxpcclient->getResponse();

echo $response;

```

Encounter.Dept.Current.Nr

Returns a numeric key to the current department of the encounter.

This key can be used to retrieve more detailed information about the department using appropriate procedure calls.

*ENR = encounter number

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | (integer) Current department number (foreign key) |
| sample php query | \$hxpcclient->query('Encounter.Dept.Current.Nr ', \$header, \$ENR) |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpcclient->query('Encounter.Dept.Current.Nr', $header, $ENR);

$response=$hxpcclient->getResponse();

echo $response;

```

Encounter.Firm.Current.Nr

Returns a numeric key to the current firm related with the encounter.

This key can be used to retrieve more detailed information about the firm using appropriate procedure calls.

*ENR = encounter number

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | (integer) Current firm number (foreign key) |
| sample php query | <code>\$hxpclient->query('Encounter.Dept.Current.Nr ', \$header, \$ENR)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Dept.Current.Nr', $header, $ENR);

$response=$hxpclient->getResponse();

echo $response;
```

Encounter.Doctor.Attending

Returns the attending doctor.

*ENR = encounter number

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | (string) Attending doctor |
| sample php query | <code>\$hxpclient->query('Encounter.Doctor.Attending', \$header, \$ENR)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Doctor.Attending', $header, $ENR);

$response=$hxpclient->getResponse();

echo $response;
```

Encounter.Doctor.Consulting

Returns the consulting doctor.

*ENR = encounter number

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | (string) Attending doctor |
| sample php query | <code>\$hxpclient->query('Encounter.Doctor.Consulting',\$header,\$ENR)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Doctor.Consulting',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;

```

Encounter.Referrer

Returns the Referrer.

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | (string) Referrer |
| sample php query | <code>\$hxpclient->query('Encounter.Referrer',\$header,\$ENR)</code> |

*ENR = encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Referrer',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;

```

Encounter.Referrer.Diagnosis

Returns the Referrer's preliminary diagnosis.

*ENR = encounter number

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | (string) Attending doctor |
| sample php query | <code>\$hxpclient->query('Encounter.Referrer.Diagnosis',\$header,\$ENR)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Referrer.Diagnosis ', $header, $ENR);

$response=$hxpclient->getResponse();

echo $response;

```

Encounter.Referrer.Therapy.Recommend

Returns the recommended therapy from the Referrer.

*ENR = encounter number

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | (string) Referrer's recommended therapy |
| sample php query | <code>\$hxpclient->query('Encounter.Referrer.Therapy.Recommend', \$header, \$ENR)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Referrer.Therapy.Recommend', $header, $ENR);

$response=$hxpclient->getResponse();

echo $response;

```

Encounter.Referrer.Notes

Returns the Referrer's notes or remarks.

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | (string) Referrer |
| sample php query | <code>\$hxpclient->query('Encounter.Referrer.Notes', \$header, \$ENR)</code> |

*ENR = encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Referrer.Notes', $header, $ENR);

$response=$hxpclient->getResponse();

```

```
echo $response;
```

Encounter.Referrer.Dept

Returns the referring department.

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | (string) Referrer department |
| sample php query | <code>\$hxpclient->query('Encounter.Referrer.Dept',\$header,\$ENR)</code> |

*ENR = encounter number

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Referrer.Dept',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;
```

Encounter.Referrer.Institution

Returns the referring department.

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | (string) Referrer department |
| sample php query | <code>\$hxpclient->query('Encounter.Referrer.Dept',\$header,\$ENR)</code> |

*ENR = encounter number

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Referrer.Dept',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;
```

Encounter.Dept.Is.In

Returns 1 if the patient is admitted in the given department.

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR, (int) department) |
| return | (integer) 1 = True, 0 = False |
| sample php query | <code>\$hxpclient->query('Encounter.Dept.Is.In',\$header,\$ENR)</code> |

*ENR = encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Referrer.Dept',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;

```

Encounter.Ward.Current.Nr

Returns a numeric key to the current ward of the encounter. This key can be used to extract more detailed information about the ward using appropriate calls.

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR, (int) department) |
| return | (integer) Current ward number (foreign key) |
| sample php query | <code>\$hxpclient->query('Encounter.Ward.Current.Nr',\$header,\$ENR)</code> |

*ENR = encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Ward.Current.Nr',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;

```

Encounter.Ward.Is.In

Returns a numeric key to ward information if the patient is admitted in the given ward. If false, the call returns either a zero or an empty string.

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR, (int) ward) |
| return | (integer) Current ward number (foreign key) |
| sample php query | <code>\$hxpclient->query('Encounter.Ward.Is.In',\$header,\$ENR,\$ward)</code> |

*ENR = encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Ward.Current.Nr',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;

```

Encounter.Followup.Date

Returns the followup date.

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | (integer) Followup date (YYYY-MM-DD) |
| sample php query | <code>\$hxpclient->query('Encounter.Followup.Date',\$header,\$ENR)</code> |

*ENR = encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Followup.Date',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;

```

Encounter.Followup.Responsible

Returns the name of person (or institution) responsible for follow up.

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | (string) Responsible for follow up |
| sample php query | <code>\$hxpclient->query('Encounter.Followup.Responsible',\$header,\$ENR)</code> |

*ENR = encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Followup.Responsible',$header,$ENR);

```

```

$response=$hxpclient->getResponse();

echo $response;

```

Encounter.Post.Notes

Returns the post-encounter notes.

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | (string) Responsible for follow up |
| sample php query | \$hxpclient->query('Encounter.Post.Notes',\$header,\$ENR) |

*ENR = encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Post.Notes',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;

```

Encounter.Service.Extra

Returns extra services information.

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR, (int) ENR) |
| return | (string) Responsible for follow up |
| sample php query | \$hxpclient->query('Encounter.Service.Extra',\$header,\$ENR) |

*ENR = encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Service.Extra',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;

```

Encounter.Record.Status

Returns the record's status.

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | (string) Record's status (e.g. Normal, locked, hidden, deleted) |
| sample php query | <code>\$hxpclient->query('Encounter.Record.Status',\$header,\$ENR)</code> |

*ENR = encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Record.Status',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;

```

Encounter.Record.History

Returns the record's history.

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | (string) Record's status (e.g. Normal, locked, hidden, deleted) |
| sample php query | <code>\$hxpclient->query('Encounter.Record.History',\$header,\$ENR)</code> |

*ENR = encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Record.History',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;

```

Encounter.Record.Hide

Returns the record's status.

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | (string) Record's status (e.g. Normal, locked, hidden, deleted) |
| sample php query | <code>\$hxpclient->query('Encounter.Record.Status',\$header,\$ENR)</code> |

*ENR = encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';

```

```

$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Record.Status',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;

```

Encounter.Record.Lock

Locks the encounter record. Sets the record status to “locked”.

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | (integer) 1 = Locked successfully, 0 = lock failed |
| sample php query | <code>\$hxpclient->query('Encounter.Record.Lock',\$header,\$ENR)</code> |

*ENR = encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Record.Lock',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;

```

Encounter.Record.Delete

“Deletes” the encounter record. Sets the record status to “deleted”. The record is not physically deleted.

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | (integer) 1 = “Deleted” successfully, 0 = “delete” failed |
| sample php query | <code>\$hxpclient->query('Encounter.Record.Delete',\$header,\$ENR)</code> |

*ENR = encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Record.Delete',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;

```

Encounter.Record.Normal

“Normalizes” the encounter record. Sets the record status to “normal”.

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | (integer) 1 = “Normalized” successfully, 0 = “normalize” failed |
| sample php query | <code>\$hxpclient->query('Encounter.Record.Normal',\$header,\$ENR)</code> |

*ENR = encounter number

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('Encounter.Record.Normal',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response;
```

Encounter.Pregnancy

Reserved

Electronic Medical Record (EMR) Calls

EMR.History.List

Returns an array of struct containing pointers to admission history (physical) records of the encounter.

| | |
|------------------|---|
| parameters | ((struct) header, (integer) ENR) |
| return | Array of struct |
| sample php query | <code>\$hxpclient->query('EMR.History.List',\$header,\$ENR)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('EMR.History.List',$header,$ENR);

$response=$hxpclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}
```

The struct contains the following keys:

| | |
|----------------|---|
| nr | (integer) Key to the record |
| short_notes | (string) Short notes |
| aux_notes | (string) Auxillary notes |
| ref_notes_nr | (integer) Key to a related record |
| personell_nr | (integer) Author's personnel number (key) |
| personell_name | (string) Author's name |
| date | (string) Date of creation (YYYY-MM-DD) |
| time | (string) Time of creation (HH:mm:ss) |

EMR.History

Returns a struct of the encounter's admission history (physical) based on a key:

| | |
|------------------|--|
| parameters | ((struct) header, (int) key) |
| return | struct |
| sample php query | <code>\$hxpclient->query('EMR.History',\$header,\$key)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$key = 1032;

$hxpclient->query('EMR.History',$header,$key);

$response=$hxpclient->getResponse();

echo $response['notes'];
```

Returned struct has the following keys:

| | |
|----------------|---|
| nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |
| type_nr | (integer) Record type number (foreign key) |
| notes | (string) Admission history (physical) |
| short_notes | (string) Short notes or summary |
| aux_notes | (string) Auxillary notes or remarks |
| ref_notes_nr | (integer) Key to a related record |
| personell_nr | (integer) Author's personnel number |
| personell_name | (string) Author's name |
| send_to_pid | (integer) PID number of record's receiving person |
| send_to_name | (string) Name of record's receiver |
| date | (string) Date of creation (YYYY-MM-DD) |
| time | (string) Time of creation (HH:mm:ss) |
| location_type | (string) Location type (e.g. Department,ward.etc) |
| location_nr | (integer) Key to location |
| location_id | (string) Location ID |
| ack_short_id | (string) Acknowledgement ID |
| date_ack | (string) Date of acknowledgement (YYYY-MM-DD) |
| date_checked | (string) Date the record checked (YYYY-MM-DD) |
| date_printed | (string) Date the record printed (YYYY-MM-DD) |
| send_by_mail | (integer) Record send by mail flag (1 = true, 0 = false) |
| send_by_email | (integer) Record send by email flag (1 = true, 0 = false) |
| send_by_fax | (integer) Record send by faxl flag (1 = true, 0 = false) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmdhhmmss) |

EMR.History.Add

Adds an admission history record for an encounter.

| | |
|------------------|---|
| parameters | ((struct) header, (int) data) |
| return | (integer) Record's primary key if successful |
| sample php query | <code>\$hxpclient->query('EMR.History',\$header,\$data)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$data['encounter_nr'] = 100308900;
$data['notes'] = ' The patient shows signs of insomnia.....';
$data['short_notes'] = 'Sleeplessness, distress';
.....
.....

$hxpclient->query('EMR.History.Add',$header,$data);

```

The data passed as parameter must be contained in a struct having the following keys:

| | |
|----------------|---|
| encounter_nr | (integer) Encounter number |
| notes | (string) Admission history (physical) (main record) |
| short_notes | (string) Short notes or summary (recommended) |
| aux_notes | (string) Auxillary notes or remarks (optional) |
| ref_notes_nr | (integer) Key to a related record (optional) |
| personell_nr | (integer) Author's personnel number (optional) |
| personell_name | (string) Author's name |
| send_to_pid | (integer) PID number of record's receiving person |
| send_to_name | (string) Name of record's receiver |
| date | (string) Date of creation (YYYY-MM-DD) (optional) |
| time | (string) Time of creation (HH:mm:ss) (optional) |
| location_type | (string) Location type (e.g. Department,ward.etc) |
| location_nr | (integer) Key to location |
| location_id | (string) Location ID |
| ack_short_id | (string) Acknowledgement ID |
| date_ack | (string) Date of acknowledgement (YYYY-MM-DD) |
| date_checked | (string) Date the record checked (YYYY-MM-DD) |
| date_printed | (string) Date the record printed (YYYY-MM-DD) |
| send_by_mail | (integer) Record send by mail flag (1 = true, 0 = false) |
| send_by_email | (integer) Record send by email flag (1 = true, 0 = false) |

| | |
|--------------|--|
| encounter_nr | (integer) Encounter number |
| send_by_fax | (integer) Record send by faxl flag (1 = true, 0 = false) |
| create_id | Username of record creator |

EMR.Prescription.List

Returns an array of struct of the encounter's prescriptions:

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | struct |
| sample php query | <code>\$hxpclient->query('EMR.Prescription.List',\$header,\$ENR)</code> |

*ENR = Encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('EMR.Prescription.List',$header,$ENR);

$response=$hxpclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}

```

Returned struct has the following keys:

| | |
|----------------------|--|
| nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |
| prescription_type_nr | (integer) Prescription type number (foreign key) |
| article | (string) Prescription article |
| drug_class | (string) Drug class |
| order_nr | (string) Related order number |
| dosage | (string) Dosage |
| application_type_nr | (integer) Applicate type number (foreign key) |
| notes | (string) Notes |
| prescribe_date | (string) Date of prescription (YYYY-MM-DD) |
| prescriber | (string) Name of prescribing doctor |
| color_marker | (string) Color marker |

| | |
|---------------|--|
| nr | (integer) Record's primary key |
| is_stopped | (integer) Prescription is stopped flag (1 = true, 0 = false) |
| location_type | (string) Location type (e.g. Department,ward.etc) |
| location_nr | (integer) Key to location |
| location_id | (string) Location ID |
| ack_short_id | (string) Acknowledgement ID |
| date_ack | (string) Date of acknowledgement (YYYY-MM-DD) |
| date_checked | (string) Date the record checked (YYYY-MM-DD) |
| date_printed | (string) Date the record printed (YYYY-MM-DD) |
| send_by_mail | (integer) Record send by mail flag (1 = true, 0 = false) |
| send_by_email | (integer) Record send by email flag (1 = true, 0 = false) |
| send_by_fax | (integer) Record send by faxl flag (1 = true, 0 = false) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmdhhmmss) |

EMR.Prescription

Returns a struct containing the encounter's prescription data based on a supplied primary key:

| | |
|------------------|---|
| parameters | ((struct) header, (int) key) |
| return | struct |
| sample php query | <code>\$hxpclient->query('EMR.Prescription',\$header,\$key)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$key = 340;

$hxpclient->query('EMR.Prescription',$header,$key);

$response=$hxpclient->getResponse();

while(list($key,$data)=each($response)){
    echo " $key >>> $data ";
}

```

Returned struct has the following keys:

| | |
|----------------------|--|
| nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |
| prescription_type_nr | (integer) Prescription type number (foreign key) |
| article | (string) Prescription article |
| drug_class | (string) Drug class |
| order_nr | (string) Related order number |
| dosage | (string) Dosage |
| application_type_nr | (integer) Applicate type number (foreign key) |
| notes | (string) Notes |
| prescribe_date | (string) Date of prescription (YYYY-MM-DD) |
| prescriber | (string) Name of prescribing doctor |
| color_marker | (string) Color marker |
| is_stopped | (integer) Prescription is stopped flag (1 = true, 0 = false) |
| location_type | (string) Location type (e.g. Department,ward.etc) |
| location_nr | (integer) Key to location |
| location_id | (string) Location ID |
| ack_short_id | (string) Acknowledgement ID |
| date_ack | (string) Date of acknowledgement (YYYY-MM-DD) |
| date_checked | (string) Date the record checked (YYYY-MM-DD) |
| date_printed | (string) Date the record printed (YYYY-MM-DD) |
| send_by_mail | (integer) Record send by mail flag (1 = true, 0 = false) |
| send_by_email | (integer) Record send by email flag (1 = true, 0 = false) |
| send_by_fax | (integer) Record send by faxl flag (1 = true, 0 = false) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmdhmmss) |

EMR.Prescription.Add

Reserved

EMR.Report.Care.List

Returns an array of structs containing nursing care reports available for an encounter:

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | Array of struct |
| sample php query | <code>\$hxpclient->query('EMR.Report.Care.List',\$header,\$ENR)</code> |

*ENR = Encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('EMR.Report.Care.List',$header,$ENR);

$response=$hxpclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}

```

EMR.Report.Care

Returns a struct of the encounter's nursing care report entry based on a key:

| | |
|------------------|--|
| parameters | ((struct) header, (int) key) |
| return | struct |
| sample php query | <code>\$hxpclient->query('EMR.Report.Care',\$header,\$key)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$key = 122;

$hxpclient->query('EMR.Report.Care',$header,$key);

$response=$hxpclient->getResponse();

echo $response['notes'];

```

Returned struct has the following keys:

| | |
|--------------|--|
| nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |
| type_nr | (integer) Record type number (foreign key) |
| notes | (string) Nursing Care Report |
| short_notes | (string) Short notes or summary |
| aux_notes | (string) Auxillary notes or remarks |

| | |
|----------------|---|
| nr | (integer) Record's primary key |
| ref_notes_nr | (integer) Key to a related record |
| personell_nr | (integer) Author's personnel number |
| personell_name | (string) Author's name |
| send_to_pid | (integer) PID number of record's receiving person |
| send_to_name | (string) Name of record's receiver |
| date | (string) Date of creation (YYYY-MM-DD) |
| time | (string) Time of creation (HH:mm:ss) |
| location_type | (string) Location type (e.g. Department,ward.etc) |
| location_nr | (integer) Key to location |
| location_id | (string) Location ID |
| ack_short_id | (string) Acknowledgement ID |
| date_ack | (string) Date of acknowledgement (YYYY-MM-DD) |
| date_checked | (string) Date the record checked (YYYY-MM-DD) |
| date_printed | (string) Date the record printed (YYYY-MM-DD) |
| send_by_mail | (integer) Record send by mail flag (1 = true, 0 = false) |
| send_by_email | (integer) Record send by email flag (1 = true, 0 = false) |
| send_by_fax | (integer) Record send by faxl flag (1 = true, 0 = false) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmdhhmmss) |

EMR.Report.Care.Add

Reserved

EMR.Report.Nursing

Alias of EMR.Report.Care

EMR.Report.Development

Returns a struct of the encounter's development report based on a key:

| | |
|------------------|---|
| parameters | ((struct) header, (int) key) |
| return | struct |
| sample php query | <code>\$hxpclient->query('EMR.Report.Development',\$header,\$key)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$key = 122;

$hxpclient->query('EMR.Report.Development',$header,$key);

$response=$hxpclient->getResponse();

echo $response['notes'];

```

Returned struct has the following keys:

| | |
|----------------|---|
| nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |
| type_nr | (integer) Record type number (foreign key) |
| notes | (string) Development report |
| short_notes | (string) Short notes or summary |
| aux_notes | (string) Auxillary notes or remarks |
| ref_notes_nr | (integer) Key to a related record |
| personell_nr | (integer) Author's personnel number |
| personell_name | (string) Author's name |
| send_to_pid | (integer) PID number of record's receiving person |
| send_to_name | (string) Name of record's receiver |
| date | (string) Date of creation (YYYY-MM-DD) |
| time | (string) Time of creation (HH:mm:ss) |
| location_type | (string) Location type (e.g. Department,ward.etc) |
| location_nr | (integer) Key to location |
| location_id | (string) Location ID |
| ack_short_id | (string) Acknowledgement ID |
| date_ack | (string) Date of acknowledgement (YYYY-MM-DD) |
| date_checked | (string) Date the record checked (YYYY-MM-DD) |
| date_printed | (string) Date the record printed (YYYY-MM-DD) |
| send_by_mail | (integer) Record send by mail flag (1 = true, 0 = false) |
| send_by_email | (integer) Record send by email flag (1 = true, 0 = false) |
| send_by_fax | (integer) Record send by faxl flag (1 = true, 0 = false) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |

| | |
|-------------|-------------------------------------|
| nr | (integer) Record's primary key |
| create_time | Creation timestamp (YYYYmmddhhmmss) |

EMR.Report.Development.Add

Reserved

EMR.Report.Care.Effectivity.List

Returns an array of structs containing the nursing care effectivity reports of an encounter.

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | struct |
| sample php query | <code>\$hxpclient->query('EMR.Report.Care.Effectivity.List',\$header,\$ENR)</code> |

*ENR = Encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('EMR.Report.Care.Effectivity.List',$header,$ENR);

$response=$hxpclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}

```

Returned struct has the following keys:

| | |
|----------------|---|
| nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |
| type_nr | (integer) Record type number (foreign key) |
| notes | (string) Nursing care effectivity report |
| short_notes | (string) Short notes or summary |
| aux_notes | (string) Auxillary notes or remarks |
| ref_notes_nr | (integer) Key to a related record |
| personell_nr | (integer) Author's personnel number |
| personell_name | (string) Author's name |
| send_to_pid | (integer) PID number of record's receiving person |
| send_to_name | (string) Name of record's receiver |
| date | (string) Date of creation (YYYY-MM-DD) |

| | |
|---------------|---|
| nr | (integer) Record's primary key |
| time | (string) Time of creation (HH:mm:ss) |
| location_type | (string) Location type (e.g. Department,ward.etc) |
| location_nr | (integer) Key to location |
| location_id | (string) Location ID |
| ack_short_id | (string) Acknowledgement ID |
| date_ack | (string) Date of acknowledgement (YYYY-MM-DD) |
| date_checked | (string) Date the record checked (YYYY-MM-DD) |
| date_printed | (string) Date the record printed (YYYY-MM-DD) |
| send_by_mail | (integer) Record send by mail flag (1 = true, 0 = false) |
| send_by_email | (integer) Record send by email flag (1 = true, 0 = false) |
| send_by_fax | (integer) Record send by faxl flag (1 = true, 0 = false) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmddhhmmss) |

EMR.Report.Care.Effectivity

Returns a struct of the nursing care effectivity report entry based on a key:

| | |
|------------------|--|
| parameters | ((struct) header, (int) key) |
| return | struct |
| sample php query | <code>\$hxpclient->query('EMR.Report.Care.Effectivity',\$header,\$key)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$key = 122;

$hxpclient->query('EMR.Report.Care.Effectivity',$header,$key);

$response=$hxpclient->getResponse();

echo $response['notes'];

```

Returned struct has the following keys:

| | |
|--------------|--------------------------------|
| nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |

| | |
|----------------|---|
| nr | (integer) Record's primary key |
| type_nr | (integer) Record type number (foreign key) |
| notes | (string) Nursing care effectivity report |
| short_notes | (string) Short notes or summary |
| aux_notes | (string) Auxillary notes or remarks |
| ref_notes_nr | (integer) Key to a related record |
| personell_nr | (integer) Author's personnel number |
| personell_name | (string) Author's name |
| send_to_pid | (integer) PID number of record's receiving person |
| send_to_name | (string) Name of record's receiver |
| date | (string) Date of creation (YYYY-MM-DD) |
| time | (string) Time of creation (HH:mm:ss) |
| location_type | (string) Location type (e.g. Department,ward.etc) |
| location_nr | (integer) Key to location |
| location_id | (string) Location ID |
| ack_short_id | (string) Acknowledgement ID |
| date_ack | (string) Date of acknowledgement (YYYY-MM-DD) |
| date_checked | (string) Date the record checked (YYYY-MM-DD) |
| date_printed | (string) Date the record printed (YYYY-MM-DD) |
| send_by_mail | (integer) Record send by mail flag (1 = true, 0 = false) |
| send_by_email | (integer) Record send by email flag (1 = true, 0 = false) |
| send_by_fax | (integer) Record send by faxl flag (1 = true, 0 = false) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmddhhmmss) |

EMR.Effectivity.Care.Add

Reserved

EMR.Incident.List

Reserved

EMR.Incident

Reserved

EMR.Incident.Add

Reserved

EMR.Immunization.List

Returns an array of structs containing the immunization records of an encounter.

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | struct |
| sample php query | <code>\$hxpclient->query('EMR.Immunization.List',\$header,\$ENR)</code> |

*ENR = Encounter number

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('EMR.Immunization.List',$header,$ENR);

$response=$hxpclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}
```

Returned struct has the following keys:

| | |
|---------------------|---|
| nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |
| date | (string) Date of prescription (YYYY-MM-DD) |
| type | (string) Immunization type |
| medicine | (string) Medicine name |
| dosage | (string) Dosage |
| application_type_nr | (integer) Application type number (key) |
| application_by | (string) Name of applying person |
| titer | (integer) Titer |
| refresh_date | (string) Date of refresh application (YYYY-MM-DD) |
| notes | (string) Notes or remarks |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |

| | |
|-------------|-------------------------------------|
| nr | (integer) Record's primary key |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmddhhmmss) |

EMR.Immunization

Returns a struct of the immunization record entry based on a record's primary key:

| | |
|------------------|---|
| parameters | ((struct) header, (int) key) |
| return | struct |
| sample php query | <code>\$hxpclient->query('EMR.Immunization',\$header,\$key)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$key = 122;

$hxpclient->query('EMR.Immunization',$header,$key);

$response=$hxpclient->getResponse();

echo $response['medicine'];

```

Returned struct has the following keys:

| | |
|---------------------|---|
| nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |
| date | (string) Date of prescription (YYYY-MM-DD) |
| type | (string) Immunization type |
| medicine | (string) Medicine name |
| dosage | (string) Dosage |
| application_type_nr | (integer) Application type number (key) |
| application_by | (string) Name of applying person |
| titer | (integer) Titer |
| refresh_date | (string) Schedule of refresher application (YYYY-MM-DD) |
| notes | (string) Notes or remarks |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmddhhmmss) |

EMR.Immunization.Add

Reserved

EMR.Diagnosis.Text

Returns a struct of the text based diagnosis based on the record's primary key.

| | |
|------------------|---|
| parameters | ((struct) header, (int) key) |
| return | struct |
| sample php query | <code>\$hxpclient->query('EMR.Diagnosis.Text',\$header,\$key)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$key = 122;

$hxpclient->query('EMR.Report.Care.Effectivity',$header,$key);

$response=$hxpclient->getResponse();

echo $response['notes'];
```

Returned struct has the following keys:

| | |
|----------------|---|
| nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |
| type_nr | (integer) Record type number (foreign key) |
| notes | (string) Diagnosis text |
| short_notes | (string) Short notes or summary |
| aux_notes | (string) Auxillary notes or remarks |
| ref_notes_nr | (integer) Key to a related record |
| personell_nr | (integer) Author's personnel number |
| personell_name | (string) Author's name |
| send_to_pid | (integer) PID number of record's receiving person |
| send_to_name | (string) Name of record's receiver |
| date | (string) Date of creation (YYYY-MM-DD) |
| time | (string) Time of creation (HH:mm:ss) |
| location_type | (string) Location type (e.g. Department,ward.etc) |
| location_nr | (integer) Key to location |
| location_id | (string) Location ID |
| ack_short_id | (string) Acknowledgement ID |
| date_ack | (string) Date of acknowledgement (YYYY-MM-DD) |

| | |
|---------------|---|
| nr | (integer) Record's primary key |
| date_checked | (string) Date the record checked (YYYY-MM-DD) |
| date_printed | (string) Date the record printed (YYYY-MM-DD) |
| send_by_mail | (integer) Record send by mail flag (1 = true, 0 = false) |
| send_by_email | (integer) Record send by email flag (1 = true, 0 = false) |
| send_by_fax | (integer) Record send by faxl flag (1 = true, 0 = false) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmdhhmmss) |

EMR.Diagnosis.Text.List

Returns an array of structs containing text based diagnoses of an encounter.

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | struct |
| sample php query | <code>\$hxpclient->query('EMR.Diagnosis.Text.List',\$header,\$ENR)</code> |

*ENR = Encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('EMR.Immunization.List',$header,$ENR);

$response=$hxpclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}

```

Returned struct has the following keys:

| | |
|--------------|--|
| nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |
| type_nr | (integer) Record type number (foreign key) |
| notes | (string) Diagnosis text |
| short_notes | (string) Short notes or summary |
| aux_notes | (string) Auxillary notes or remarks |

| | |
|----------------|---|
| nr | (integer) Record's primary key |
| ref_notes_nr | (integer) Key to a related record |
| personell_nr | (integer) Author's personnel number |
| personell_name | (string) Author's name |
| send_to_pid | (integer) PID number of record's receiving person |
| send_to_name | (string) Name of record's receiver |
| date | (string) Date of creation (YYYY-MM-DD) |
| time | (string) Time of creation (HH:mm:ss) |
| location_type | (string) Location type (e.g. Department,ward.etc) |
| location_nr | (integer) Key to location |
| location_id | (string) Location ID |
| ack_short_id | (string) Acknowledgement ID |
| date_ack | (string) Date of acknowledgement (YYYY-MM-DD) |
| date_checked | (string) Date the record checked (YYYY-MM-DD) |
| date_printed | (string) Date the record printed (YYYY-MM-DD) |
| send_by_mail | (integer) Record send by mail flag (1 = true, 0 = false) |
| send_by_email | (integer) Record send by email flag (1 = true, 0 = false) |
| send_by_fax | (integer) Record send by faxl flag (1 = true, 0 = false) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmdhhmmss) |

EMR.Diagnosis.Text.Add

Reserved

EMR.Diagnosis.Code

Returns a struct containing ICD coded diagnosis of an encounter based on the record's primary key.

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | struct |
| sample php query | <code>\$hxpcient->query('EMR.Diagnosis.Code.List',\$header,\$ENR)</code> |

*ENR = Encounter number

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';
```

```

$ENR = 20031009900;

$hxpclient->query('EMR.Diagnosis.Code.List',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response['code'];

```

Returned struct has the following keys:

| | |
|----------------------|--|
| diagnosis_nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |
| op_nr | (integer) Record type number (foreign key) |
| date | (string) Date of entry (YYYY-MM-DD) |
| code | (string) ICD code |
| code_parent | (string) ICD code of parent diagnosis |
| group_nr | (integer) DRG group number (key) |
| code_version | (integer) ICD code version number |
| local_code | (string) Code of the local code system mapped to ICD |
| category_nr | (integer) Category number (key) |
| type | (string) Diagnosis type |
| localization | (string) Localization of diagnosis |
| diagnosing_clinician | (string) Name of diagnosing clinician/physician |
| diagnosing_dept_nr | (integer) Diagnosing department number (key) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmddhhmmss) |

EMR.Diagnosis.Code.List

Returns an array of structs containing ICD code based diagnoses of an encounter.

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | struct |
| sample php query | <code>\$hxpclient->query('EMR.Diagnosis.Code.List',\$header,\$ENR)</code> |

*ENR = Encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

```



```

$ENR = 20031009900;

$hxpclient->query('EMR.Diagnosis.Code.List',$header,$ENR);

$response=$hxpclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}

```

Returned struct has the following keys:

| | |
|----------------------|--|
| diagnosis_nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |
| op_nr | (integer) Record type number (foreign key) |
| date | (string) Date of entry (YYYY-MM-DD) |
| code | (string) ICD code |
| code_parent | (string) ICD code of parent diagnosis |
| group_nr | (integer) DRG group number (key) |
| code_version | (integer) ICD code version number |
| local_code | (string) Code of the local code system mapped to ICD |
| category_nr | (integer) Category number (key) |
| type | (string) Diagnosis type |
| localization | (string) Localization of diagnosis |
| diagnosing_clinician | (string) Name of diagnosing clinician/physician |
| diagnosing_dept_nr | (integer) Diagnosing department number (key) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmdhhmmss) |

EMR.Diagnosis.Code.Add

Reserved

EMR.Discharge.Summary.List

Returns an array of structs containing discharge summaries of an encounter.

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | struct |
| sample php query | \$hxpclient->query('EMR.Discharge.Summary.List',\$header,\$ENR) |

*ENR = Encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('EMR.Discharge.Summary.List',$header,$ENR);

$response=$hxpclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}

```

Returned struct has the following keys:

| | |
|----------------|--|
| nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |
| type_nr | (integer) Record type number (key) |
| notes | (string) Discharge summary |
| short_notes | (string) Short notes or summary |
| aux_notes | (string) Auxillary notes or remarks |
| ref_notes_nr | (integer) Key to a related record |
| personell_nr | (integer) Author's personnel number |
| personell_name | (string) Author's name |
| send_to_pid | (integer) PID number of record's receiving person |
| send_to_name | (string) Name of record's receiver |
| date | (string) Date of creation (YYYY-MM-DD) |
| time | (string) Time of creation (HH:mm:ss) |
| location_type | (string) Location type (e.g. Department,ward.etc) |
| location_nr | (integer) Key to location |
| location_id | (string) Location ID |
| ack_short_id | (string) Acknowledgement ID |
| date_ack | (string) Date of acknowledgement (YYYY-MM-DD) |
| date_checked | (string) Date the record checked (YYYY-MM-DD) |
| date_printed | (string) Date the record printed (YYYY-MM-DD) |
| send_by_mail | (integer) Record send by mail flag (1 = true, 0 = false) |

| | |
|---------------|---|
| nr | (integer) Record's primary key |
| send_by_email | (integer) Record send by email flag (1 = true, 0 = false) |
| send_by_fax | (integer) Record send by faxl flag (1 = true, 0 = false) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmdhmmss) |

EMR.Discharge.Summary

Returns a struct containing discharge summary record of an encounter based on the record's primary key.

| | |
|------------------|--|
| parameters | ((struct) header, (int) key) |
| return | struct |
| sample php query | <code>\$hxpclient->query('EMR.Discharge.Summary',\$header,\$key)</code> |

*ENR = Encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('EMR.Discharge.Summary',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response['notes'];

```

Returned struct has the following keys:

| | |
|----------------|-------------------------------------|
| nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |
| type_nr | (integer) Record type number (key) |
| notes | (string) Discharge summary |
| short_notes | (string) Short notes or summary |
| aux_notes | (string) Auxillary notes or remarks |
| ref_notes_nr | (integer) Key to a related record |
| personell_nr | (integer) Author's personnel number |
| personell_name | (string) Author's name |

| | |
|---------------|---|
| nr | (integer) Record's primary key |
| send_to_pid | (integer) PID number of record's receiving person |
| send_to_name | (string) Name of record's receiver |
| date | (string) Date of creation (YYYY-MM-DD) |
| time | (string) Time of creation (HH:mm:ss) |
| location_type | (string) Location type (e.g. Department,ward.etc) |
| location_nr | (integer) Key to location |
| location_id | (string) Location ID |
| ack_short_id | (string) Acknowledgement ID |
| date_ack | (string) Date of acknowledgement (YYYY-MM-DD) |
| date_checked | (string) Date the record checked (YYYY-MM-DD) |
| date_printed | (string) Date the record printed (YYYY-MM-DD) |
| send_by_mail | (integer) Record send by mail flag (1 = true, 0 = false) |
| send_by_email | (integer) Record send by email flag (1 = true, 0 = false) |
| send_by_fax | (integer) Record send by faxl flag (1 = true, 0 = false) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmdhhmmss) |

EMR.Discharge.Summary.Add

Reserved

EMR.Vitals.Weight.List

Returns an array of structs containing weight measurement data of an encounter.

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | Array of struct |
| sample php query | <code>\$hxpclient->query('EMR.Vitals.Weight.List',\$header,\$ENR)</code> |

*ENR = Encounter number

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';
```

```
$ENR = 20031009900;
```

```
$hxpclient->query('EMR.Vitals.Weight.List',$header,$ENR);
```

```

$response=$hxpclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}

```

Returned struct has the following keys:

| | |
|--------------|--|
| nr | (integer) Record's primary key |
| msr_date | (string) Date of measurement (YYYY-MM-DD) |
| msr_time | (string) Time of measurement (HH:mm:ss) |
| encounter_nr | (integer) Encounter number |
| msr_type_nr | (integer) Measurement type number (key) |
| value | (integer) Measured weight value |
| unit_nr | (integer) Key number to unit of measurement |
| notes | (string) Notes or remarks |
| measured_by | (string) Name of person taking measurement |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmddhhmmss) |

EMR.Vitals.Weight

Returns a struct containing weight measurement data of an encounter based on the record's primary key.

| | |
|------------------|--|
| parameters | ((struct) header, (int) key) |
| return | struct |
| sample php query | <code>\$hxpclient->query('EMR.Vitals.Weight',\$header,\$key)</code> |

*ENR = Encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('EMR.Vitals.Weight',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response['value'];

```

Returned struct has the following keys:

| | |
|--------------|--|
| nr | (integer) Record's primary key |
| msr_date | (string) Date of measurement (YYYY-MM-DD) |
| msr_time | (string) Time of measurement (HH:mm:ss) |
| encounter_nr | (integer) Encounter number |
| msr_type_nr | (integer) Measurement type number (key) |
| value | (integer) Measured weight value |
| unit_nr | (integer) Key number to unit of measurement |
| notes | (string) Notes or remarks |
| measured_by | (string) Name of person taking measurement |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmddhhmmss) |

EMR.Vitals.Weight.Add

Reserved

EMR.Vitals.Head.Circumference.List

Returns an array of structs containing head circumference measurement data of an encounter.

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | Array of struct |
| sample php query | <code>\$hxpclient->query('EMR.Vitals.Head.Circumference.List',\$header,\$ENR)</code> |

*ENR = Encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

```

```

$ENR = 20031009900;

$hxpclient->query('EMR.Vitals.Head.Circumference.List',$header,$ENR);

$response=$hxpclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}

```

Returned struct has the following keys:

| | |
|--------------|--|
| nr | (integer) Record's primary key |
| msr_date | (string) Date of measurement (YYYY-MM-DD) |
| msr_time | (string) Time of measurement (HH:mm:ss) |
| encounter_nr | (integer) Encounter number |
| msr_type_nr | (integer) Measurement type number (key) |
| value | (integer) Measured head circumference value |
| unit_nr | (integer) Key number to unit of measurement |
| notes | (string) Notes or remarks |
| measured_by | (string) Name of person taking measurement |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmddhhmmss) |

EMR.Vitals.Head.Circumference

Returns a struct containing head circumference measurement data of an encounter based on the record's primary key.

| | |
|------------------|--|
| parameters | ((struct) header, (int) key) |
| return | struct |
| sample php query | <code>\$hxpclient->query('EMR.Vitals.Head.Circumference',\$header,\$key)</code> |

*ENR = Encounter number

Sample use (php, without error handling)

```
$header['usr']='demouser';  
$header['pw']='demopass';  
  
$ENR = 20031009900;  
  
$hxpclient->query('EMR.Vitals.Head.Circumference',$header,$ENR);  
  
$response=$hxpclient->getResponse();  
  
echo $response['value'];
```

Returned struct has the following keys:

| | |
|--------------|--|
| nr | (integer) Record's primary key |
| msr_date | (string) Date of measurement (YYYY-MM-DD) |
| msr_time | (string) Time of measurement (HH:mm:ss) |
| encounter_nr | (integer) Encounter number |
| msr_type_nr | (integer) Measurement type number (key) |
| value | (integer) Measured head circumference value |
| unit_nr | (integer) Key number to unit of measurement |
| notes | (string) Notes or remarks |
| measured_by | (string) Name of person taking measurement |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmddhhmmss) |

EMR.Vitals.Head.Circumference.Add

Reserved

EMR.Vitals.Head.Circumference.Update

Reserved

EMR.Vitals.Height.List

Returns an array of structs containing height measurement data of an encounter.

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | Array of struct |
| sample php query | \$hxpclient->query('EMR.Vitals.Height.List',\$header,\$ENR) |

*ENR = Encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('EMR.Vitals.Height.List',$header,$ENR);

$response=$hxpclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}

```

Returned struct has the following keys:

| | |
|--------------|--|
| nr | (integer) Record's primary key |
| msr_date | (string) Date of measurement (YYYY-MM-DD) |
| msr_time | (string) Time of measurement (HH:mm:ss) |
| encounter_nr | (integer) Encounter number |
| msr_type_nr | (integer) Measurement type number (key) |
| value | (integer) Measured height value |
| unit_nr | (integer) Key number to unit of measurement |
| notes | (string) Notes or remarks |
| measured_by | (string) Name of person taking measurement |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmddhhmmss) |

EMR.Vitals.Height

Returns a struct containing height measurement data of an encounter based on the record's primary key.

| | |
|------------------|--|
| parameters | ((struct) header, (int) key) |
| return | struct |
| sample php query | \$hxpclient->query('EMR.Vitals.Height',\$header,\$key) |

*ENR = Encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('EMR.Vitals.Height',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response['value'];

```

Returned struct has the following keys:

| | |
|--------------|--|
| nr | (integer) Record's primary key |
| mnr_date | (string) Date of measurement (YYYY-MM-DD) |
| mnr_time | (string) Time of measurement (HH:mm:ss) |
| encounter_nr | (integer) Encounter number |
| mnr_type_nr | (integer) Measurement type number (key) |
| value | (integer) Measured height value |
| unit_nr | (integer) Key number to unit of measurement |
| notes | (string) Notes or remarks |
| measured_by | (string) Name of person taking measurement |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmddhhmmss) |

EMR.Vitals.Height.Add

Reserved

EMR.Vitals.Height.Update

Reserved

EMR.Vitals.Temperature.List

Returns an array of structs containing temperature measurement data of an encounter.

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | Array of struct |
| sample php query | \$hxpclient->query('EMR.Vitals.Temperature.List',\$header,\$ENR) |

*ENR = Encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('EMR.Vitals.Temperature.List',$header,$ENR);

$response=$hxpclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}

```

Returned struct has the following keys:

| | |
|--------------|--|
| nr | (integer) Record's primary key |
| msr_date | (string) Date of measurement (YYYY-MM-DD) |
| msr_time | (string) Time of measurement (HH:mm:ss) |
| encounter_nr | (integer) Encounter number |
| msr_type_nr | (integer) Measurement type number (key) |
| value | (integer) Measured temperature value |
| unit_nr | (integer) Key number to unit of measurement |
| notes | (string) Notes or remarks |
| measured_by | (string) Name of person taking measurement |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmddhhmmss) |

EMR.Vitals.Temperature

Returns a struct containing temperature measurement data of an encounter based on the record's primary key.

| | |
|------------------|---|
| parameters | ((struct) header, (int) key) |
| return | struct |
| sample php query | <code>\$hxpclient->query('EMR.Vitals.Temperature',\$header,\$key)</code> |

*ENR = Encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('EMR.Vitals.Temperature',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response['value'];

```

Returned struct has the following keys:

| | |
|--------------|--|
| nr | (integer) Record's primary key |
| msr_date | (string) Date of measurement (YYYY-MM-DD) |
| msr_time | (string) Time of measurement (HH:mm:ss) |
| encounter_nr | (integer) Encounter number |
| msr_type_nr | (integer) Measurement type number (key) |
| value | (integer) Measured temperature value |
| unit_nr | (integer) Key number to unit of measurement |
| notes | (string) Notes or remarks |
| measured_by | (string) Name of person taking measurement |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmddhhmmss) |

EMR.Vitals.Temperature.Add

Reserved

EMR.Vitals.Temperature.Update

Reserved

EMR.Vitals.Bloodpressure.List

Returns an array of structs containing blood pressure measurement data of an encounter.

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | Array of struct |
| sample php query | \$hxpclient->query('EMR.Vitals.Temperature.List',\$header,\$ENR) |

*ENR = Encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('EMR.Vitals.Temperature.List',$header,$ENR);

$response=$hxpclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}

```

Returned struct has the following keys:

| | |
|--------------|--|
| nr | (integer) Record's primary key |
| msr_date | (string) Date of measurement (YYYY-MM-DD) |
| msr_time | (string) Time of measurement (HH:mm:ss) |
| encounter_nr | (integer) Encounter number |
| msr_type_nr | (integer) Measurement type number (key) |
| value | (integer) Measured blood pressure value |
| unit_nr | (integer) Key number to unit of measurement |
| notes | (string) Notes or remarks |
| measured_by | (string) Name of person taking measurement |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmddhhmmss) |

EMR.Vitals.Bloodpressure

Returns a struct containing blood pressure measurement data of an encounter based on the record's primary key.

| | |
|------------------|---|
| parameters | ((struct) header, (int) key) |
| return | struct |
| sample php query | \$hxpclient->query('EMR.Vitals.Bloodpressure',\$header,\$key) |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('EMR.Vitals.Bloodpressure',$header,$key);

$response=$hxpclient->getResponse();

echo $response['value'];

```

Returned struct has the following keys:

| | |
|--------------|--|
| nr | (integer) Record's primary key |
| msr_date | (string) Date of measurement (YYYY-MM-DD) |
| msr_time | (string) Time of measurement (HH:mm:ss) |
| encounter_nr | (integer) Encounter number |
| msr_type_nr | (integer) Measurement type number (key) |
| value | (integer) Measured blood pressure value |
| unit_nr | (integer) Key number to unit of measurement |
| notes | (string) Notes or remarks |
| measured_by | (string) Name of person taking measurement |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmddhhmmss) |

EMR.Vitals.Bloodpressure.Add

Reserved

EMR.Vitals.Bloodpressure.Update

Reserved

EMR.Vitals.Input.List

Reserved

EMR.Vitals.Input

Reserved

EMR.Vitals.Input.Add

Reserved

EMR.Vitals.Output.List

Reserved

EMR.Vitals.Output

Reserved

EMR.Vitals.Output.Add

Reserved

EMR.Report.Daily.Ward.List

Returns an array of structs containing daily reports about the patient in the ward.

| | |
|------------------|---|
| parameters | ((struct) header, (int) ENR) |
| return | Array of struct |
| sample php query | <code>\$hxpclient->query('EMR.Report.Daily.Ward.List',\$header,\$ENR)</code> |

*ENR = Encounter number

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('EMR.Report.Daily.Ward.List',$header,$ENR);

$response=$hxpclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}
```

Returned struct has the following keys:

| | |
|--------------|------------------------------------|
| nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |
| type_nr | (integer) Record type number (key) |
| notes | (string) Ward day report |

| | |
|----------------|---|
| nr | (integer) Record's primary key |
| short_notes | (string) Short notes or summary |
| aux_notes | (string) Auxillary notes or remarks |
| ref_notes_nr | (integer) Key to a related record |
| personell_nr | (integer) Author's personnel number |
| personell_name | (string) Author's name |
| send_to_pid | (integer) PID number of record's receiving person |
| send_to_name | (string) Name of record's receiver |
| date | (string) Date of creation (YYYY-MM-DD) |
| time | (string) Time of creation (HH:mm:ss) |
| location_type | (string) Location type (e.g. Department,ward.etc) |
| location_nr | (integer) Key to location |
| location_id | (string) Location ID |
| ack_short_id | (string) Acknowledgement ID |
| date_ack | (string) Date of acknowledgement (YYYY-MM-DD) |
| date_checked | (string) Date the record checked (YYYY-MM-DD) |
| date_printed | (string) Date the record printed (YYYY-MM-DD) |
| send_by_mail | (integer) Record send by mail flag (1 = true, 0 = false) |
| send_by_email | (integer) Record send by email flag (1 = true, 0 = false) |
| send_by_fax | (integer) Record send by faxl flag (1 = true, 0 = false) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmdhmmss) |

EMR.Report.Daily.Ward

Returns a struct containing daily report about the patient in the ward based on the record's primary key.

| | |
|------------------|--|
| parameters | ((struct) header, (int) key) |
| return | struct |
| sample php query | <code>\$hxpclient->query('EMR.Report.Daily.Ward',\$header,\$key)</code> |

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';
```



```

$ENR = 20031009900;

$hxpclient->query('EMR.Report.Daily.Ward',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response['notes'];

```

Returned struct has the following keys:

| | |
|----------------|---|
| nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |
| type_nr | (integer) Record type number (key) |
| notes | (string) Ward's day report |
| short_notes | (string) Short notes or summary |
| aux_notes | (string) Auxillary notes or remarks |
| ref_notes_nr | (integer) Key to a related record |
| personell_nr | (integer) Author's personnel number |
| personell_name | (string) Author's name |
| send_to_pid | (integer) PID number of record's receiving person |
| send_to_name | (string) Name of record's receiver |
| date | (string) Date of creation (YYYY-MM-DD) |
| time | (string) Time of creation (HH:mm:ss) |
| location_type | (string) Location type (e.g. Department,ward.etc) |
| location_nr | (integer) Key to location |
| location_id | (string) Location ID |
| ack_short_id | (string) Acknowledgement ID |
| date_ack | (string) Date of acknowledgement (YYYY-MM-DD) |
| date_checked | (string) Date the record checked (YYYY-MM-DD) |
| date_printed | (string) Date the record printed (YYYY-MM-DD) |
| send_by_mail | (integer) Record send by mail flag (1 = true, 0 = false) |
| send_by_email | (integer) Record send by email flag (1 = true, 0 = false) |
| send_by_fax | (integer) Record send by faxl flag (1 = true, 0 = false) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmdhhmmss) |

EMR.Report.Daily.Ward.Add

Reserved

EMR.Report.Daily.Chart.List

Returns an array of structs containing daily chart notes/reports about the encounter.

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | Array of struct |
| sample php query | <code>\$hxpclient->query('EMR.Report.Daily.Chart.List',\$header,\$ENR)</code> |

*ENR = Encounter number

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('EMR.Report.Daily.Chart.List',$header,$ENR);

$response=$hxpclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}
```

Returned struct has the following keys:

| | |
|----------------|---|
| nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |
| type_nr | (integer) Record type number (key) |
| notes | (string) Chart's day notes or report |
| short_notes | (string) Short notes or summary |
| aux_notes | (string) Auxillary notes or remarks |
| ref_notes_nr | (integer) Key to a related record |
| personell_nr | (integer) Author's personnel number |
| personell_name | (string) Author's name |
| send_to_pid | (integer) PID number of record's receiving person |
| send_to_name | (string) Name of record's receiver |
| date | (string) Date of creation (YYYY-MM-DD) |
| time | (string) Time of creation (HH:mm:ss) |
| location_type | (string) Location type (e.g. Department,ward.etc) |
| location_nr | (integer) Key to location |
| location_id | (string) Location ID |
| ack_short_id | (string) Acknowledgement ID |

| | |
|---------------|---|
| nr | (integer) Record's primary key |
| date_ack | (string) Date of acknowledgement (YYYY-MM-DD) |
| date_checked | (string) Date the record checked (YYYY-MM-DD) |
| date_printed | (string) Date the record printed (YYYY-MM-DD) |
| send_by_mail | (integer) Record send by mail flag (1 = true, 0 = false) |
| send_by_email | (integer) Record send by email flag (1 = true, 0 = false) |
| send_by_fax | (integer) Record send by faxl flag (1 = true, 0 = false) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmddhhmmss) |

EMR.Report.Daily.Chart

Returns a struct containing daily chart notes/report about the encounter based on the record's primary key.

| | |
|------------------|---|
| parameters | ((struct) header, (int) key) |
| return | struct |
| sample php query | <code>\$hxpclient->query('EMR.Report.Daily.Chart',\$header,\$key)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('EMR.Report.Daily.Chart',$header,$ENR);

$response=$hxpclient->getResponse();

echo $response['notes'];

```

Returned struct has the following keys:

| | |
|--------------|-------------------------------------|
| nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |
| type_nr | (integer) Record type number (key) |
| notes | (string) Chart's day report |
| short_notes | (string) Short notes or summary |
| aux_notes | (string) Auxillary notes or remarks |

| | |
|----------------|---|
| nr | (integer) Record's primary key |
| ref_notes_nr | (integer) Key to a related record |
| personell_nr | (integer) Author's personnel number |
| personell_name | (string) Author's name |
| send_to_pid | (integer) PID number of record's receiving person |
| send_to_name | (string) Name of record's receiver |
| date | (string) Date of creation (YYYY-MM-DD) |
| time | (string) Time of creation (HH:mm:ss) |
| location_type | (string) Location type (e.g. Department, ward.etc) |
| location_nr | (integer) Key to location |
| location_id | (string) Location ID |
| ack_short_id | (string) Acknowledgement ID |
| date_ack | (string) Date of acknowledgement (YYYY-MM-DD) |
| date_checked | (string) Date the record checked (YYYY-MM-DD) |
| date_printed | (string) Date the record printed (YYYY-MM-DD) |
| send_by_mail | (integer) Record send by mail flag (1 = true, 0 = false) |
| send_by_email | (integer) Record send by email flag (1 = true, 0 = false) |
| send_by_fax | (integer) Record send by faxl flag (1 = true, 0 = false) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmdhhmmss) |

EMR.Report.Daily.Chart.Add

Reserved

EMR.Report.Daily.IV.List

Reserved

EMR.Report.Daily.IV.Add

Reserved

EMR.Report.Daily.Anticoagulant.List

Reserved

EMR.Report.Daily.Anticoagulant.Add

Reserved

EMR.Report.Development.List

Returns an array of structs containing development reports about the encounter.

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | Array of struct |
| sample php query | <code>\$hxpclient->query('EMR.Report.Development.List',\$header,\$ENR)</code> |

*ENR = Encounter number

Sample use (php, without error handling)

```
$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('EMR.Report.Development.List',$header,$ENR);

$response=$hxpclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}
```

Returned struct has the following keys:

| | |
|----------------|---|
| nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |
| type_nr | (integer) Record type number (key) |
| notes | (string) Development report |
| short_notes | (string) Short notes or summary |
| aux_notes | (string) Auxillary notes or remarks |
| ref_notes_nr | (integer) Key to a related record |
| personell_nr | (integer) Author's personnel number |
| personell_name | (string) Author's name |
| send_to_pid | (integer) PID number of record's receiving person |
| send_to_name | (string) Name of record's receiver |
| date | (string) Date of creation (YYYY-MM-DD) |
| time | (string) Time of creation (HH:mm:ss) |
| location_type | (string) Location type (e.g. Department,ward.etc) |
| location_nr | (integer) Key to location |
| location_id | (string) Location ID |
| ack_short_id | (string) Acknowledgement ID |
| date_ack | (string) Date of acknowledgement (YYYY-MM-DD) |

| | |
|---------------|---|
| nr | (integer) Record's primary key |
| date_checked | (string) Date the record checked (YYYY-MM-DD) |
| date_printed | (string) Date the record printed (YYYY-MM-DD) |
| send_by_mail | (integer) Record send by mail flag (1 = true, 0 = false) |
| send_by_email | (integer) Record send by email flag (1 = true, 0 = false) |
| send_by_fax | (integer) Record send by faxl flag (1 = true, 0 = false) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmdhhmmss) |

EMR.Report.Development

Returns a struct containing development report for an encounter based on the record's primary key.

| | |
|------------------|---|
| parameters | ((struct) header, (int) key) |
| return | struct |
| sample php query | <code>\$hxpclient->query('EMR.Report.Development',\$header,\$key)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('EMR.Report.Development',$header,$key);

$response=$hxpclient->getResponse();

echo $response['notes'];

```

Returned struct has the following keys:

| | |
|--------------|-------------------------------------|
| nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |
| type_nr | (integer) Record type number (key) |
| notes | (string) Development report |
| short_notes | (string) Short notes or summary |
| aux_notes | (string) Auxillary notes or remarks |
| ref_notes_nr | (integer) Key to a related record |
| personell_nr | (integer) Author's personnel number |

| | |
|----------------|---|
| nr | (integer) Record's primary key |
| personell_name | (string) Author's name |
| send_to_pid | (integer) PID number of record's receiving person |
| send_to_name | (string) Name of record's receiver |
| date | (string) Date of creation (YYYY-MM-DD) |
| time | (string) Time of creation (HH:mm:ss) |
| location_type | (string) Location type (e.g. Department,ward.etc) |
| location_nr | (integer) Key to location |
| location_id | (string) Location ID |
| ack_short_id | (string) Acknowledgement ID |
| date_ack | (string) Date of acknowledgement (YYYY-MM-DD) |
| date_checked | (string) Date the record checked (YYYY-MM-DD) |
| date_printed | (string) Date the record printed (YYYY-MM-DD) |
| send_by_mail | (integer) Record send by mail flag (1 = true, 0 = false) |
| send_by_email | (integer) Record send by email flag (1 = true, 0 = false) |
| send_by_fax | (integer) Record send by fax flag (1 = true, 0 = false) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmdhhmmss) |

EMR.Report.Development.Add
EMR.Report.Daily.PT.List
EMR.Report.Daily.PT.Add
EMR.Test.Result.List
EMR.Test.Result
EMR.Test.Request.List
EMR.Test.Request.Add

EMR.Therapy.Text.List

Returns an array of structs containing text based therapy records about the encounter.

| | |
|------------------|--|
| parameters | ((struct) header, (int) ENR) |
| return | Array of struct |
| sample php query | <code>\$hxpclient->query('EMR.Therapy.Text.List',\$header,\$ENR)</code> |

*ENR = Encounter number

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('EMR.Therapy.Text.List',$header,$ENR);

$response=$hxpclient->getResponse();

while(list($ar,$st)=each($response)){
    while(list($key,$data)=each($st)){
        echo " $key >>> $data ";
    }
}

```

Returned struct has the following keys:

| | |
|----------------|---|
| nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |
| type_nr | (integer) Record type number (key) |
| notes | (string) Therapy text |
| short_notes | (string) Short notes or summary |
| aux_notes | (string) Auxillary notes or remarks |
| ref_notes_nr | (integer) Key to a related record |
| personell_nr | (integer) Author's personnel number |
| personell_name | (string) Author's name |
| send_to_pid | (integer) PID number of record's receiving person |
| send_to_name | (string) Name of record's receiver |
| date | (string) Date of creation (YYYY-MM-DD) |
| time | (string) Time of creation (HH:mm:ss) |
| location_type | (string) Location type (e.g. Department,ward.etc) |
| location_nr | (integer) Key to location |
| location_id | (string) Location ID |
| ack_short_id | (string) Acknowledgement ID |
| date_ack | (string) Date of acknowledgement (YYYY-MM-DD) |
| date_checked | (string) Date the record checked (YYYY-MM-DD) |
| date_printed | (string) Date the record printed (YYYY-MM-DD) |
| send_by_mail | (integer) Record send by mail flag (1 = true, 0 = false) |
| send_by_email | (integer) Record send by email flag (1 = true, 0 = false) |
| send_by_fax | (integer) Record send by faxl flag (1 = true, 0 = false) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |

| | |
|-------------|-------------------------------------|
| nr | (integer) Record's primary key |
| create_time | Creation timestamp (YYYYmmddhhmmss) |

EMR.Therapy.Text

Returns a struct containing text based therapy data for an encounter based on the record's primary key.

| | |
|------------------|---|
| parameters | ((struct) header, (int) key) |
| return | struct |
| sample php query | <code>\$hxpclient->query('EMR.Therapy.Text',\$header,\$key)</code> |

Sample use (php, without error handling)

```

$header['usr']='demouser';
$header['pw']='demopass';

$ENR = 20031009900;

$hxpclient->query('EMR.Therapy.Text',$header,$key);

$response=$hxpclient->getResponse();

echo $response['notes'];

```

Returned struct has the following keys:

| | |
|----------------|---|
| nr | (integer) Record's primary key |
| encounter_nr | (integer) Encounter number |
| type_nr | (integer) Record type number (key) |
| notes | (string) Therapy text |
| short_notes | (string) Short notes or summary |
| aux_notes | (string) Auxillary notes or remarks |
| ref_notes_nr | (integer) Key to a related record |
| personell_nr | (integer) Author's personnel number |
| personell_name | (string) Author's name |
| send_to_pid | (integer) PID number of record's receiving person |
| send_to_name | (string) Name of record's receiver |
| date | (string) Date of creation (YYYY-MM-DD) |
| time | (string) Time of creation (HH:mm:ss) |
| location_type | (string) Location type (e.g. Department,ward.etc) |
| location_nr | (integer) Key to location |
| location_id | (string) Location ID |
| ack_short_id | (string) Acknowledgement ID |
| date_ack | (string) Date of acknowledgement (YYYY-MM-DD) |

| | |
|---------------|---|
| nr | (integer) Record's primary key |
| date_checked | (string) Date the record checked (YYYY-MM-DD) |
| date_printed | (string) Date the record printed (YYYY-MM-DD) |
| send_by_mail | (integer) Record send by mail flag (1 = true, 0 = false) |
| send_by_email | (integer) Record send by email flag (1 = true, 0 = false) |
| send_by_fax | (integer) Record send by faxl flag (1 = true, 0 = false) |
| status | Record's status (normal,hidden,deleted,locked) |
| history | Record's history |
| modify_id | Username of modifier |
| modify_time | Modification timestamp (YYYYMMDDhhmmss) |
| create_id | Username of record creator |
| create_time | Creation timestamp (YYYYmmddhhmmss) |

EMR.Therapy.Text.Add

Reserved

EMR.Birth.Details
 EMR.Orders.Doctor.List
 EMR.Orders.Doctor.Add
 EMR.Operation.Notes
 EMR.Operation.Notes.Add
 EMR.Problem.Care
 EMR.Problem.Care.Add
 EMR.Allergy.List
 EMR.Allergy.Add
 EMR.Inquiry.ToDoctor
 EMR.Inquiry.ToDoctor.Add
 EMR.Inquiry.ToNurse
 EMR.Inquiry.ToNurse.Add
 EMR.Diet.Daily
 EMR.Diet.Daily.Add
 EMR.Image.Dicom.List
 EMR.Image.Dicom
 EMR.Image.List
 EMR.Image.URL
 EMR.Image.Add

Department

Department
 Department.Name
 Department.ID
 Department.Description
 Department.Logo
 Department.Logo.Filename
 Department.List

Ward

Ward
Ward.Name
Ward.ID
Ward.Description
Ward.List
Ward.Room.List

Room

Room.Nr
Room.Bed.List
Room.Description

System

System.Error.Keys.error_msg
System.Error.Keys.error_nr
System.Error.Keys.List
System.Header.Keys.lang
System.Header.Keys.sid
System.Header.Keys.version
System.Header.Keys.List

System.HXP.PCD.Version
System.HXP.PCD.Version.List
System.HXP.PCD.Anonymized
System.HXP.Type.List
System.HXP.Type.Set

System.Server.Advanced.Exists

Anonymized data calls

Anonymized data calls receive anonymized data. Anonymized data are medical data which are true and accurate but cannot be directly linked to a patient nor his personal data. Most of the anonymized data are extracted for purposes of research and statistics. The data are strictly selected to ensure the anonymity of the patient.

Anonymized data calls are strictly "READ ONLY" calls.

All anonymized data calls begin with "Anon".

Anon.PID.List
Anon.PID.Search
Anon.PID.Search.ByDOB
Anon.PID.Search.ByAddress
Anon.PID.Dead.List
Anon.PID.Search.ByAge
Anon.PID.Male.List
Anon.PID.Female.List
Anon.Demographic
Anon.Encounter.ENR.List
Anon.Encounter.ENR.Search.ByDiagnosis
Anon.Encounter.ENR.Search.ByTherapy
Anon.Encounter.ENR.Search.ByPrescription
Anon.Encounter.ENR.Search.ByIncident
Anon.Outpatient.ENR.List
Anon.Inpatient.ENR.List
Anon.Encounter
Anon.EMR.History.List
Anon.EMR.History
Anon.EMR.Prescription.List
Anon.EMR.Prescription
Anon.EMR.Report.Care
Anon.EMR.Report.Development
Anon.EMR.Effectivity.Care
Anon.EMR.Incident.List

Anon.EMR.Incident
Anon.EMR.Immunization.List
Anon.EMR.Immunization
Anon.EMR.Diagnosis.Text
Anon.EMR.Diagnosis.Text.List
Anon.EMR.Diagnosis.Code
Anon.EMR.Diagnosis.Code.List
Anon.EMR.Discharge.Summary
Anon.EMR.Vitals.Weight.List
Anon.EMR.Vitals.Head.Circumference.List

Anon.EMR.Vitals.Height.List

Anon.EMR.Vitals.Temperature.List
Anon.EMR.Vitals.Bloodpressure.List
Anon.EMR.Vitals.Input.List
Anon.EMR.Vitals.Output.List
Anon.EMR.Report.Daily.List
Anon.EMR.Report.Daily
Anon.EMR.Report.Daily.IV.List
Anon.EMR.Report.Daily.Anticoagulant.List
Anon.EMR.Report.Development
Anon.EMR.Report.Daily.PT.List
Anon.EMR.Test.Result.List
Anon.EMR.Test.Result
Anon.EMR.Test.Request.List
Anon.EMR.Therapy.Text.List
Anon.EMR.Therapy.Text
Anon.EMR.Birth.Details
Anon.EMR.Orders.Doctor.List
Anon.EMR.Operation.Notes
Anon.EMR.Problem.Care
Anon.EMR.Allergy.List
Anon.EMR.Inquiry.ToDoctor
Anon.EMR.Inquiry.ToNurse
Anon.EMR.Diet.Daily