Friday, January 18, 2013

This document contains the description of the changes to OpenEMR to allow Mobility Access to OpenEMR's functions. All items in the Mobility API are completed and ready.

We have added two tables to the database to be used by our Mobility API when it is presented for certification. Neither table is used by standard or rebranded OpenEMR and the SQL statements for creating the files are below. The files are used to tokenize sessions and calls made by the Mobility API for increased security of the Mobility API. The SQL statements for creation of the tables is include in subsequent pages.

Attached to the email this letter is included in is a copy of all of MedMaster Mobility API files. These files are basically modified versions of the OpenEMR API files that include additional security, encryption, and some changes needed for the iOS application.

All of the API calls return XML files with the appropriate data and success calls. We currently place these files in the medmasterproapi folder. We welcome the community's suggestions on this as well as the contents of the API.

Thanks!

Sincerely,

Karl R. Englund
CTO

tokens

```
SET NAMES utf8;
SET FOREIGN_KEY_CHECKS = 0;


-- ----------------------------
--  Table structure for `tokens`
-- ----------------------------
DROP TABLE IF EXISTS `tokens`;
CREATE TABLE `tokens` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `user_id` bigint(20) DEFAULT NULL,
  `token` varchar(150) DEFAULT NULL,
  `device_token` varchar(200) NOT NULL,
  `create_datetime` datetime DEFAULT NULL,
  `expire_datetime` datetime DEFAULT NULL,
  `message_badge` int(5) NOT NULL,
  `appointment_badge` int(5) NOT NULL,
  `labreports_badge` int(5) NOT NULL,
  `prescription_badge` int(5) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=200 DEFAULT CHARSET=utf8;


SET FOREIGN_KEY_CHECKS = 1;
```

The table contains one record per user that has successfully logged in via MedMaster Mobility. It is part of session tokenization to insure the highest levels of security during a mobility connection.

medmasterusers

```
SET NAMES utf8;
SET FOREIGN_KEY_CHECKS = 0;

-- ----------------------------
--  Table structure for `medmasterusers`
-- ----------------------------
DROP TABLE IF EXISTS `medmasterusers`;
CREATE TABLE `medmasterusers` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `firstname` varchar(300) DEFAULT NULL,
  `lastname` varchar(300) DEFAULT NULL,
  `phone` varchar(150) DEFAULT NULL,
  `email` varchar(300) DEFAULT NULL,
  `username` varchar(150) DEFAULT NULL,
  `password` varchar(150) DEFAULT NULL,
  `pin` varchar(150) DEFAULT NULL,
  `create_date` date DEFAULT NULL,
  `secret_key` varchar(96) DEFAULT NULL,
  `greeting` varchar(30) DEFAULT NULL,
  `title` varchar(64) DEFAULT 'Doctor',
  `uid` int(11) NOT NULL,
  `ip_address` varchar(20) NOT NULL,
  `country_code` varchar(10) NOT NULL,
  `country_name` varchar(50) NOT NULL,
  `region_name` varchar(50) NOT NULL,
  `city_name` varchar(50) NOT NULL,
  `zip_code` varchar(20) NOT NULL,
  `latidute` varchar(20) NOT NULL,
  `longitude` varchar(20) NOT NULL,
  `time_zone` varchar(10) NOT NULL,
  `emr` enum('openemr','greenway') NOT NULL DEFAULT 'openemr',
  `password_with_token` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=193 DEFAULT CHARSET=utf8;


SET FOREIGN_KEY_CHECKS = 1;
```

The table contains one record per user that has successfully registered in MedMaster Mobility for connection to OpenEMR. It is part of session tokenization to insure the highest levels of security during a mobility connection. Secret keys and encryption of passwords, etc. are used.

MedMaster Pro API structure:

- API Input:

  With every API call we need user to send authentication token ,Which will be verified with the token generated at the time of login.
  When user performs the login API call his credentials are matched with the stored information and a token for that user is generated and updated in "tokens" table.
  Other parameters required by a certain API call are supposed to be sent as POST parameters.

- API Output:

  Every API call response consists of data in XML format.
  In response data there will always be status tag that indicates if API call has been processed successfully or not.
  There will always be a message tag that will contain the description of the status code being returned in the status tag of the response.

MedMaster Pro API Deployment
After installation of OpenEMR following steps should be taken in order to make the MedMaster Pro API work correctly with the newly deployed OpenEMR:

- Make sure that the OpenEMR is installed correctly and is working perfectly before you begin to deploy the MedMaster Pro API.

- Copy/upload medmasterproapi.rar after unzipping inside the openemr directory (which results as a new installlation of OpenEMR). (medmasterproapi.rar is available at AWS - 184.72.52.108 inside /var/www/deploymentdata).

- Run http://your-openemr-url/medmasterproapi/initsetup.php, which will make necessary updates in OpenEMR database to work with MedMaster Pro API.

- Modify Setting2.xml file (Setting2.xml is available at AWS - 184.72.52.108 inside /var/www/deploymentdata) with proper path/URL to Medmaster Pro API deployment and upload it to your server.

- Distribute URL for Settings2.xml or whatever you have named it and uploaded to the users so that they can enter the provided URL in iOS app settings.

NOTE : You can use http://your-openemr-url/medmasterproapi/resetdb/ reset_database.php with proper credentials to refresh the OpenEMR deployed database to start with new dataset whenever you want.

List of Medmaster Pro API Files

- addappointment.php

  This API is used to schedule an appointment (Add an Appointment)
- addcontactgeneral.php

  This API add contact for user with contact image.
- addfacility.php

  This API used to add facilities.
- addinsurancecompany.php

  This API add new Insurance Companies.
- addlist.php

  This API add medication, Allergy, Medical problem, Dental, Surgery of a patient associated with/without visit.
- addlocation.php

  This API Add a location for already added facility.
- addpatient.php

  This API add new patient we can also save an image of patient when adding new patient. All insurance information (Primary, Secondary and Other Insurance) is also saved in this API.
- addpatientdocument.php

  We can add all types of document related to patient in this API.
- addpatientdocumentwithlink.php

  This API is also used to add patient documents but here a document is added via link.
- addpatientimage.php

  This API is used to save patient image on server.
- addprescription.php

  This API is used to add prescription associated with patient.
- addresource.php

  This API is used to add resources(Link, Image, Pdf, Video) for user.
- addresourcewithlink.php

  This API is also used to save resources but with link provided by user.
- addreviewofsystems.php

  This API is used to add ROS related to a patient visit.

- addroschecks.php

  This service is used to add ROS checks to a patient visit.
- addsoap.php

  This service is used to add SOAPs to a patient visit.
- addvisit.php

  This API is used to add Visits for patient.
- addvisitvitals.php

  This API add vitals against patient visit.

- deleteappointment

  This API removes a scheduled appointment.
- deletecontactgeneral.php

  This API removes contacts from users contact list.
- deletepatientdocument.php

  This appointment removes patient documents.
- deleteprescription.php

  This API removes patient existing prescriptions.
- deleteresource.php

  This API removes patient resources.
- deletesoap.php

  This API removes patient SOAP.
- deletevisit.php

  This API removes patient visit.
- forgetpassword.php

  This API rest user password, pin and send them through email.
- login.php

  This API checks user credentials also returns dashboard information.
- loginwithpin.php

  This API login user with his pin.
- register.php

  This service is used for signup user for both iOS application and OPENEMR.
- report_appointments.php

  This API generates all user (provider) appointments reports, html(sha1 converted) and there pdf file (sha1 converted).
- report_appt_visits.php

  This API generates all appointments plus all visits for user(provider) report there html(sha1 converted) and there pdf file (sha1 converted).
- report_visits.php

  This API generates all visits reports for user(provider), html(sha1 converted) and there pdf file (sha1 converted).
- resetpasswordpin.php

This API used to reset password /pin.

- searchappointments.new.php

  This API is used to search appointments with different parameters.
- searchdiagnosiscode.php

  This Service is used to find diagnosis code.
- searchdrug.php

  This API used to search drugs.
- searchpatient.php

  This service is used to search patient with his firstname and lastname.
- searchrx.php

  This API used to search prescription and can applying different filters on it.
- sendmessage.php

  This API is used to send message about patient to different user.
- updateappointment.php

  This API updates information in existing scheduled appointments.
- updatecontactgeneral.php

  This API updates information of user contacts it can also updates contact image.
- updatefacility.php

  This API update facility information.
- updateinsurancecompany.php

  With this API we update insurance company information.
- updatelist.php

  This API updates user medication, Allergy, Medical problem, Dental, Surgery information.
- updatelocation.php

  This API Updates location information for already added facility.
- updatenotificationbadge.php

  This API updates count for push notification against token.
- updatepatient.php

  This API updates patient information along with patient image.
- updatepatientdocument.php

  This API updates patient documents.

- updatepatientnotes.php

  This API updates notes for patient.
- updateprescription.php

  This API update patient prescription data.
- updateprofileimage

  This API used to update patient profile image.
- updatereviewofsystems.php

  This API updates patient ROS information.
- updateroschecks.php

  This API updates patient ROS checks information.
- updatesoap.php

  This API updates patient SOAP information.
- updatevisit.php

  This API updates patient visit information.
- updatevisitvitals.php

  This API updates patient visit vital information.
- visitsummery.php

  This API is used to get Summery(number of Soaps + latestSoap, number of ROS + latest ROS, number of ROS checks + latest ROS checks, number of vitals + latest vital) of any visit.
- addfeesheet.php

  This API is use for add fee sheet items and preparing bills of patient.
- deletefeesheet.php

  This API is use to delete fee sheet item.
- addCheckout.php

  This API is use for add patient checkout information for billing (billed amount, Discount, payment methods and payment information).
- addpayment.php

  This API is use for add batch payments of patient.
- updatefeesheet.php

  This API is use to update fee sheet items.
- getallpatients.php

This API is use to get list and details of All patients.

- getappointments.php

  This API is use to get list of all scheduled appointments.

- getcontact.php

  This API is use to specific contact information.

- getcontactgeneral.php

  This API is use to get List and details of contacts.

- getdailyschedule.php

  This API is use to get daily schedule.

- getfacility.php

  This API is use get the Information of facility.

- getfeesheet.php

  This API is use to retrieve the patient stored fee items.

- getfeesheetoptions.php

  This API is use for get Patient information (New, Established) and feesheet Items for add fee sheet.

- getinsurancecompanies.php

  This API is use for get details of available insurance companies.

- getinsurancecompany.php

  This API is use for get details of specific insurance company.

- getlist.php

  This API is use for get problem list like (allergies, surgeries, and dentals etc).

- getlistbyvisitid.php

  This API is use for get problem list like (allergies, surgeries, and dentals etc) for specific visit.

- getlocations.php

  This API is use for get list and details of locations for the facility.

- getlocation.php

  This API is use for get details of the location.

- getmessages.php

  This API is use to get list and details of messages.

- getonotes.php

- getpatientdocuments.php

  This API is use to get list and details of documents for patient.
- getpatientrecord.php

  This API is use to get details (demographics, medical problems, dentals, medication, allergies, surgeries, and vitals) of a patient.
- getprescription.php

  This API is use to get prescription of a patient.
- getproviders.php

  This API is use to get list and details of providers.
- getresources.php

  This API is use to get list of resources for a user.
- getreviewofsystems.php

  This API is use to get list and details of review of systems of a patient for a visit.
- getreviewofsystemsbyid.php

  This API is use to get list and details of review of systems of a patient.
- getreviewofsystemssummary.php

  This API is use to get summary of review of systems of a patient.
- getroschecks.php

  This API is use to get list and details of ROS checks of a patient for a visit.
- getroschecksbyid.php

  This API is use to get list and details of ROS checks of a patient.
- getroscheckssummary.php

  This API is use to get summary of ROS checks of a patient.
- getsendmessages.php

  This API is use to get list and details of send messages.
- getsoap.php

  This API is use to get SOAP list of a patient for a visit.
- getsoaplist.php

  This API is use to get list and details of SOAP of a patient.
- getstatsoptions.php

  This API is use to get list and details of stats (language, race and ethnicity).

- getuserlist.php

  This API is use to get list and details of system users.
- getvisits.php

  This API is use to get list and details of visits of a patients.
- getvitals.php

  This API is use to get vitals of patient for a visit.